An Energy Efficient ECG Ventricular Ectopic Beat Classifier Using Binarized CNN for Edge AI Devices

David Liang Tai Wong, *Senior Member, IEEE*, Yongfu Li, *Senior Member, IEEE*, Deepu John, *Senior Member, IEEE*, Weng Khuen Ho, *Senior Member, IEEE* Chun Huat Heng, *Senior Member, IEEE*

***Abstract*— Wearable Artificial Intelligence-of-Things (AIoT) requires edge devices to be resource and energy-efficient. In this paper, we design and implement an efficient binary convolutional neural network (bCNN) algorithm utilizing function-merging and block-reuse techniques to classify between Ventricular and non-Ventricular Ectopic Beat images. We deploy our model into a low-resource low-power field programmable gate array (FPGA) fabric. Our model achieves a classification accuracy of 97.3%, sensitivity of 91.3%, specificity of 98.1%, precision of 86.7%, and F1-score of 88.9%, along with dynamic power dissipation of only 10.5-µW.**

***Index Terms*— Artificial Intelligence-of-Things, low-power design, wearable, field programmable gate array, ECG, co-design, convolutional neural network, inference, fusion, state machine**

I. Introduction

CARDIOVASCULAR diseases (CVD) remain as the leading cause of death according to the World Health Organization [1]–[3]. Early detection of CVD via continuous vital signs monitoring and preventive healthcare management could potentially reduce complication and cost. In this regard, a wearable electrocardiogram (ECG) device can aid in arrhythmia detection, such as the ventricular ectopic (V) beat that originates from the ventricles, disrupting the regular heart rhythm [4], [5]. However, commonly deployed Holter devices for ECG recording require manual classification and is labour intensive.

Recently, there has been growing interest in Artificial Intelligence-of-Things (AIoT) for both wearable and healthcare applications [6]–[13]. Nevertheless, to realize intelligence at the edge entails trade-off between model complexity, inference accuracy, computing resources and power consumption [14]–[18]. In this paper, we attempt to incorporate such ECG anomalies detection at the edge device.

D. L. T. Wong, W. K. Ho, and C.-H. Heng are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583 (e-mail: elehch@nus.edu.sg).

Y. Li is with the Department of Micro-Nano Electronics and MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University, Shanghai 200240, China.).

D. John is with the School of Electrical and Electronic Engineering, University College Dublin, Dublin 4, Ireland.

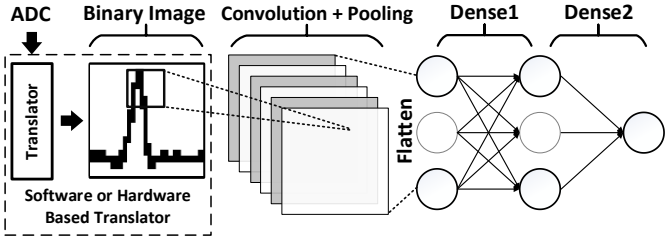

Fig. 1: Overall design of the bCNN model

An ECG anomaly detection algorithm have been reported in [19], [20]. However, the algorithm requires statistical knowledge of the signal and domain expertise to determine the key parameters for the model. Furthermore, the implementation is demonstrated on PC or high-end FPGA instead of low-power wearables. Machine learning has also been deployed in these regards. However, domain expertise is often needed for feature engineering, and to improve the detection accuracy [21]–[24].

The 1D and 2D CNN deep learning model for ECG classification eliminates the need for feature selection and has shown promising results [25]–[27]. However, 2D CNN is gaining traction over 1D CNN in terms of robustness during ambulatory along with better spatial awareness due to an additional dimension and the ability to perform feature enhancement [28]–[32]. Nevertheless, deep learning models are computationally expensive. A model with 2 to 3 convolution layers for ECG classification requires 1M to 50M MAC operations [31], [33]–[36].

In this work, we propose a binarized 2D CNN to reduce the computational complexity. This allows the multiplication to be simplified with simple XNOR logic. However, bCNN requires us to first convert the ECG samples from ADC into binary images. In this work, we employ a software method to convert ADC samples into binary images, as our focus is on the simplified anomaly detection problem. Likewise, it should be noted that we can also implement the ADC to image conversion using hardware. We also propose a novel function-merging and block-reuse techniques for complexity reduction of the bCNN, such that it can be implemented in a compact field programmable gate array (FPGA). This would also aid in reducing the total power consumption.

This paper is organized as follows. Section II outlines the methodology and algorithm of the experimental setup. The overall hardware architecture is discussed in Section III. The results are then compared against other works in Section IV. Finally, we conclude our work in Section V.

II. Methodology And Algorithm

A. Model Design and Test

The Massachusetts Institute of Technology – Boston's Beth Israel Hospital (MIT-BIH) Arrhythmia Database is used to assess the performance of our model. It contains 48 records, digitized at 360 Hz per channel with an 11-bit resolution with a maximum signal amplitude of 10 mV [37].

As shown in Fig. 1, our bCNN model contains a merged convolution+pooling layer, and 2 fully-connected dense layers (Dense1 and Dense2). Binary hyperbolic tangent activation function (bTanH) is employed for both bCNN and Dense1 layer while sigmoid activation function is used in Dense2 layer. The bTanH activation function seamlessly provides binary output for the subsequent input layer, which simplifies the multiplication operation (XNOR) for subsequent layer.

We evaluate our bCNN using Python with Keras library on a 2.50 GHz Intel Xeon with 128 GB memory along with Nvidia GTX 2080 GPU. During training, the batch and epoch size are set to 2048 and 10000, respectively. Adam optimizer is chosen with starting and ending learning rate of 10^{-3} and 10^{-4}, respectively. The same training parameters are used on different models. The weights containing the lowest loss rate is stored and then ported to FPGA for inference on our proposed edge device.

The ANSI/AAMI EC-57 testing standard is used in our methodology to classify ventricular ectopic (V) beat from other classes such as non-ectopic (N), supraventricular ectopic (S), fusion (F) and unknown (Q). The model utilizes the first 5 minutes of each record for training. The test datasets are taken from 11 records (200, 202, 210, 213, 214, 219, 221, 228 231, 233 and 234). The total number of V and non-V beats are 8554 and 77916, respectively. Given the imbalance of V beat class, it is resampled from the training set to avoid bias during training. In this paper, each ECG beat is converted into binary image tagged with its respective label, where V beat is assigned as class 1 and the rest as 0. The binary image is generated from ADC samples via a translator as shown in Fig. 1. In this paper, the translation is done using software. The software based approach allows the generation of image with various sizes to optimize the bCNN model.

B. Samples To Binary Image Conversion

The main purpose of converting ECG samples into binary images is to leverage on bCNN algorithm and eliminate the need for feature engineering. Given that the MIT-BIH has R-peak annotated for each ECG beat, we took 55 ECG samples around the R-peak (22 and 32 samples on the left and right of R-peak) [38]–[45]. The Python 3 programming language and the open-source OpenCV2 library are then employed to convert the samples into an image. A 256×256 8-bit RGB image is first obtained by up-sampling the columns and re-scaling the rows. We enhanced the resolution from 55 to 256 by interpolating the missing samples to avoid stair-case connection and connect the upscale ECG signal. Then, we re-scale the samples vertically,

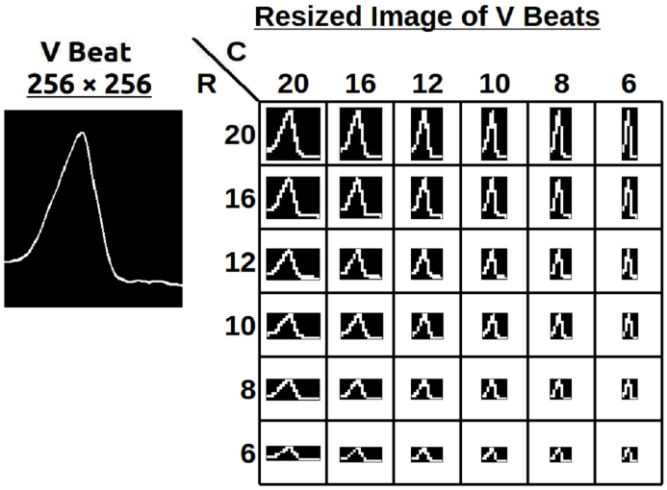

Fig. 2: A template resolution (256×256) of a V beat resized into smaller resolution of rows and columns

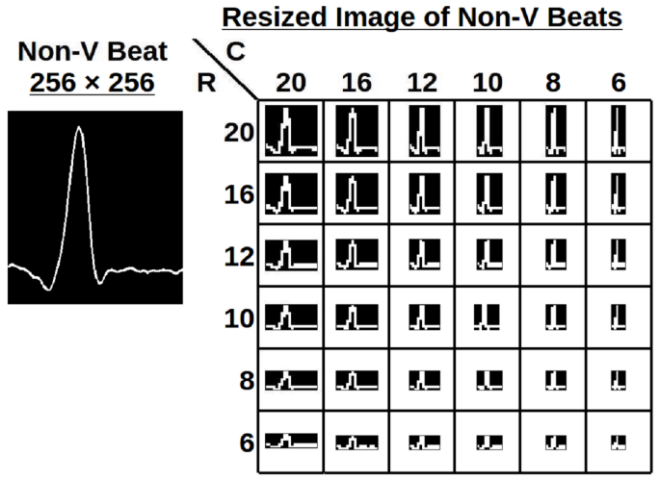

Fig. 3: A template resolution (256×256) of a non-V beat resized into smaller resolution of rows and columns

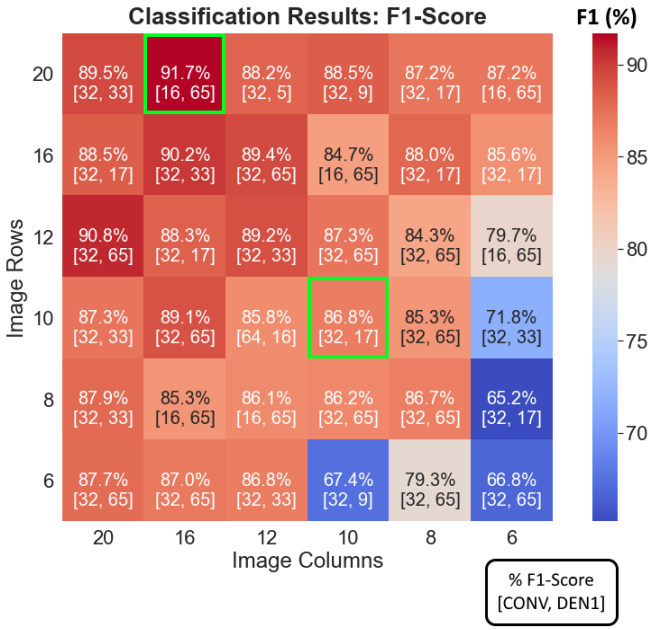

Fig. 4: F1-Score using the multi-size image as input into the bCNN network

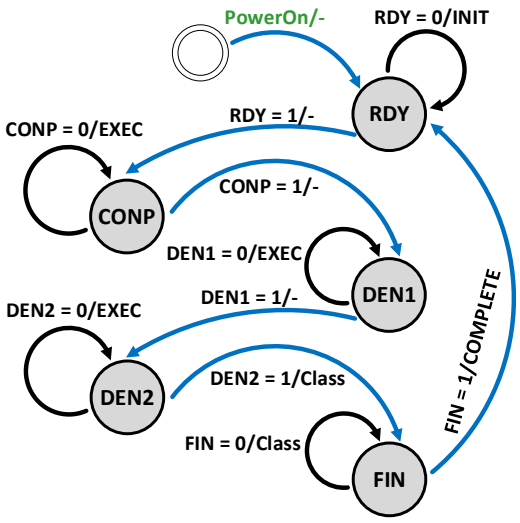

Fig. 5: Overall mealy state machine with its transitions [17]

where all morphologies are scaled accordingly. These operations are done by amplitude normalization where all samples are subtracted against the lowest value and then divided by the largest value. Following that, the samples are scaled by 256. Through scaling, the prominent R-peak is also enlarged which helps to improve the classification accuracy. To reduce the image size, we rasterize the image into 8-bit greyscale. The resulting image will form the basis template for the study of our bCNN model with different image sizing (20×20, 16×16, 16×20, 20×16, 10×8, etc). To generate binary images of various pixels, each grayscale pixel is then compared against a threshold. Fig. 2 shows the 256×256 template image with V beat, and various resized images on the right. Fig. 3 shows the images corresponding to non-V beat. The various resized images allow us to study their impact on classification accuracy and model complexity.

C. Design Space Exploration And Target Hardware

Using the 2D images generated, we explore the model complexity by exploring the required number of convolution filters {32, 16, 8, 4, and 2} and the number of nodes needed for Dense1 layer {65, 33, 17, 9, 5, 3}. Binary networks are known to be notoriously hard to train [46], [47]. Thus, we perform a Monte Carlo simulation with various settings (Image sizing, number of convolution filters, number of nodes for Dense1) and pick the model with the best F1-score. Due to the highly imbalanced dataset of ECG, F1-score is used as the evaluation metric [48].

As shown in Fig. 4, x and y-axis indicate the number of column and row pixels while the number of convolution filters and nodes in Dense1 layer are indicated within the square bracket [M, N]. We noted that higher F1-Score is observed at the top-left corner with larger number of column and row pixels. The F1-score degrades towards the bottom-right corner with reduced number of column and row pixels. This is expected as larger image size contains more input information to attain better F1-score. However, it is not straightforward to deduce F1-score relationship to the number of convolution filters and number of nodes in Dense1 layer from Fig. 4. Nevertheless, it can serve as a rough guide for choosing the model. Two models are chosen based on performance and power, which is indicated in green.

As shown, our first chosen better performance (BP) model is based on image size of 16×20, which gives rise to the best F1-score of 91.7% along with accuracy, train and validation loss of 98.0%, 0.03 and 0.09. Our second chosen low power (LP) model is based on image size of 10×10, which only requires 32 convolution filters and 17 nodes for Dense1 layer while attaining decent F1-score of 86.8% along with accuracy, train and validation loss of 97.0%, 0.20 and 0.22.

By having BP and LP models, we can reconfigure the edge device to trade-off F1-score with power consumption. In this work, the Lattice Semiconductor's ultra-low power iCE40 UltraPlus iCE40UP5k is chosen as the target FPGA for our wearable edge IoT device [49]. It has a small footprint, which contains only 5k 4-input lookup tables (LUT) and registers (REG). The pre-trained weights are stored in the 120 kbits of EBR SRAM embedded within the FPGA.

III. Overall Hardware Architecture

Our bCNN model can be classified into three main parts. They are finite state machine (FSM) for orchestrating and controlling modules, a function-merging block that combines binary convolution with pooling (bCONP), and a fully connected binarized dense network for output classification.

A. Finite State Machine (FSM)

The FSM is responsible for triggering various modules within the bCNN in sequence and ensuring signal transfer between modules. It contains 5 states, i.e. RDY, CONP, DEN1, DEN2, and FIN as illustrated in Fig. 5. The transition begins with PowerOn in green and the main transition pathways between states are indicated by the blue arrows. During RDY state, all registers within the system are initialized. At this point, the state machine monitors the RDY flag and transitions into CONP when RDY flag is set to 1. When CONP is active, the function-merging block (bCONP) will continue to execute (EXEC) the convolution and pooling operations. With CONP flag set to 1, CONP state will transition into DEN1 state to perform the subsequent multiply-accumulate (MAC) operation. Similar routine is repeated for DEN2 state. Finally, classification is done at FIN state. Once FIN flag is set and 'Complete' interrupt signal is asserted, the whole procedure is repeated for the next inference cycle.

B. Function-Merging Block (bCONP)

In a CNN architecture, the input samples are convolved with a 3×3 filter to form a single output sample of the feature map. This is then followed by a 4-input 2×2 max-pooling for down-sampling, which results in a single bit output. By examining the convolution and max-pooling operation in details, it is found that the required data by max-pooling functions depends on the 4-neighboring 3×3 convolutional filters output. With bCNN, the 4-neighboring 3×3 convolutions and subsequent max-pooling can be merged into a single combinational logic operation (bCONP). There are 3 main advantages. Firstly, it eliminates the intermediate registers and saves the resources. Secondly, with the reduced registers, register-only datapath can be implemented, eliminating the constant exchanges between the datapath and SRAM for intermediate result storage. This also reduces the number of operation clock cycles, and eliminates logic needed for synchronization. Finally, the

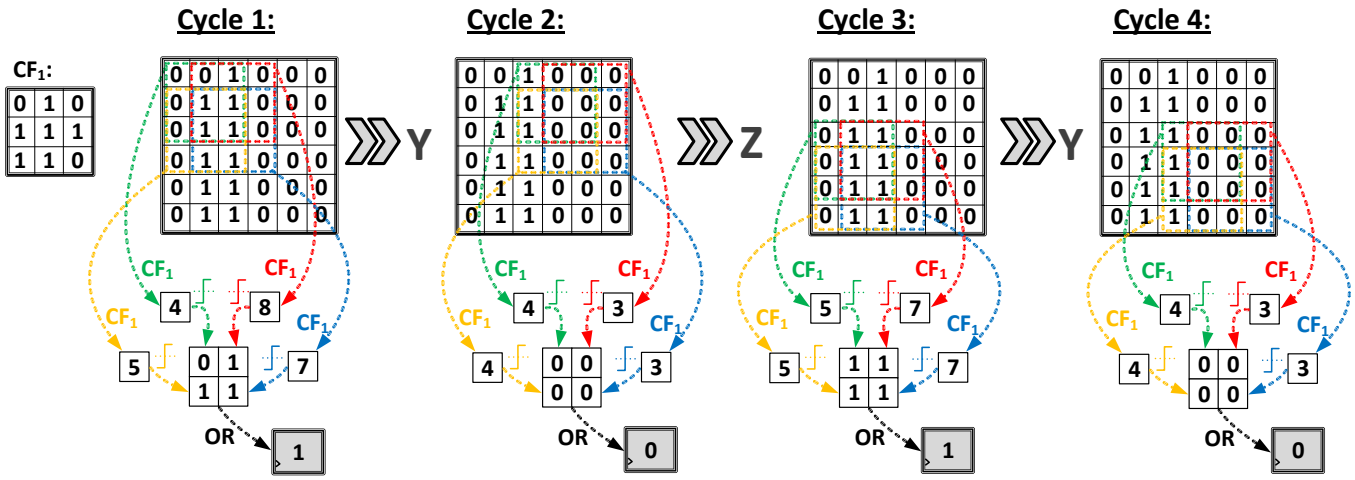

Fig. 6: Fusion of convolution to pooling layer on a 6x6 binary image [17]

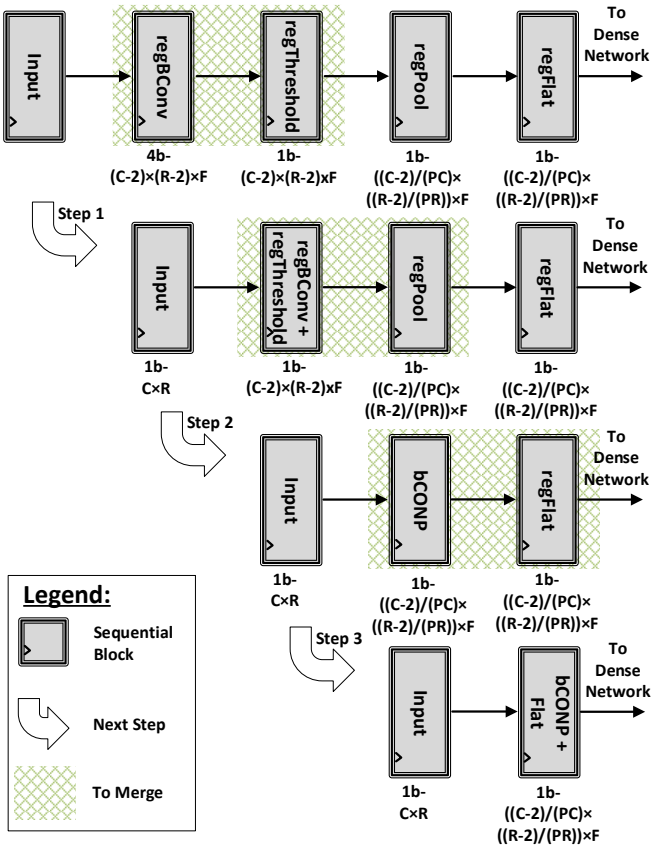

Fig. 7: Register optimization by merging regBConv, regThreshold and regPool into CONP + Flat

bCONP processing element (PE) can be reused as it strides through the entire binary image. This leads to better energy and resource efficiency at the expense of latency. With the low sampling rate of ECG signals, faster system clock can be employed to overcome the latency issue.

The cycle-by-cycle walkthrough of the proposed function-merging bCONP is illustrated in Fig. 6. During the first cycle, a combinatorial binary multiply and accumulate (cbMAC) is used to perform 3×3 pixels convolution with the first convolution filter (CF1). A two-input 9-bit XNOR performs the multiplication while the summation is done through a population count (popcount). The resulting output from the popcount is then fed into bTanH activation function. Both the cbMAC and bTanH operations are performed in parallel on the 4 neighboring 3×3 pixels as indicated in green, red, yellow, and blue giving rise to a 4-bit 2×2 output. The output is then sent into the binary max-pooling operation, which employs a simple 4-input OR gate to generate the 1-bit output. All the described 80 operations needed for bCONP are performed within one clock cycle, eliminating intermediate registers. For subsequent cycles, the bCONP operation is applied to different 4 neighboring 3×3 pixels along the row and column directions depending on the stride. In the example shown, stride of 2 is illustrated. Overall, the number of clock cycles is reduced by 4× with bCONP compared to conventional approach.

To determine the number of registers saved, we examine the detailed RTL implementation as shown in Fig. 7. In conventional model, the 'MAC' block requires regBConv registers, regThreshold registers to store the thresholds, and regPool and regFlat registers for pooling and flattening, respectively. The amounts of registers needed can be estimated using (1) to (3), where C, R, PC, and PR represents the size of image columns, image rows, pool column, and pool row, respectively.

$$regBConv = 4 \times (C-2) \times (R-2) \times \mathrm{F} \quad (1)$$

$$regThreshold = 1 \times (C-2) \times (R-2) \times \mathrm{F} \quad (2)$$

$$regPool = regFlat = 1 \times \left(\frac{C-2}{PC}\right) \times \left(\frac{R-2}{PR}\right) \times \mathrm{F} \quad (3)$$

With bCONP, these registers can be eliminated. The detailed estimation of the number of registers required, clock cycles, and others are given in Section IV.

C. Binarized Dense Network

The direct implementation of a fully connected dense network would require an XNOR engine per input neuron along with its corresponding weights of the target output neuron. This will lead to a large number of XNORs following the convolution. To reduce the resources required, we propose bDense processing element containing a single accumulator located at each output neuron along with two circular buffers containing the inputs and their corresponding weights. The concept is illustrated in Fig. 8 with only 4 flatten neurons, 2 Dense1 neurons, and 1 Dense2 neuron. In the example, for Dense1 layer, an array of registers (red) is used to store 1-bit

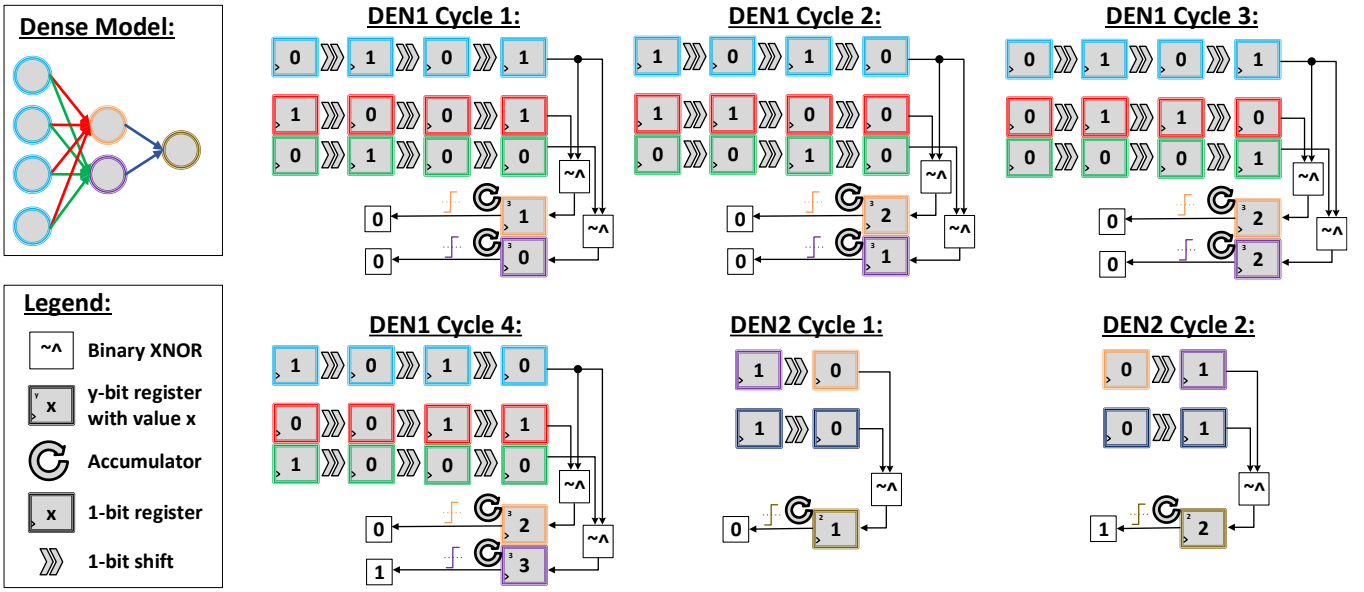

Fig. 8.: Overall architecture of the dense architecture with two sbMAC-bTanH engines at DEN1 and a single sbMAC-bTanH at DEN2 [17]

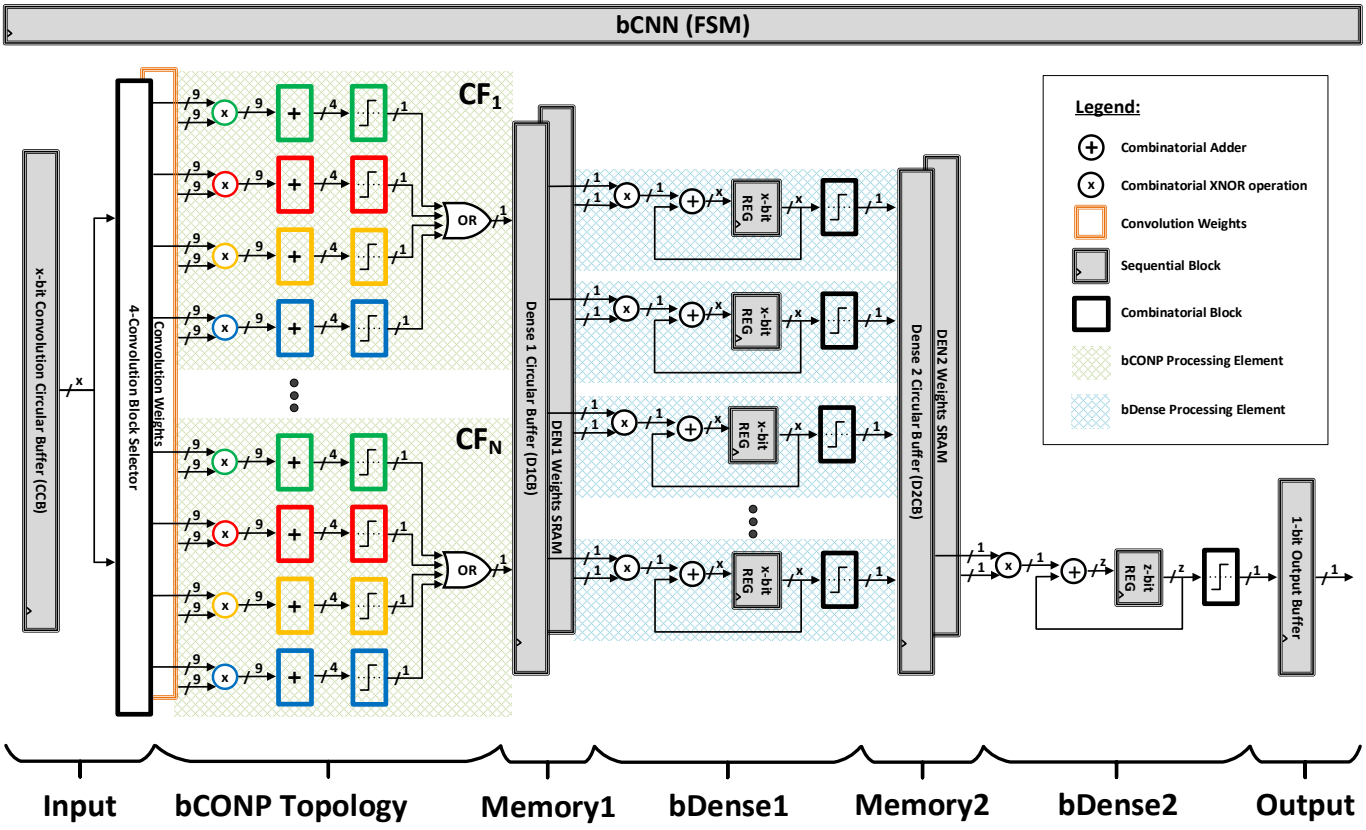

Fig. 9: Overall bCNN architecture containing the combinatorial and sequential block for binary image classification

binary weights between each input and the first output neuron. Similarly, another array of registers (green) is used for the 2nd output neuron. The size of the array is the same as input neurons (cyan) shown on top. Instead of performing MAC for all input neuron at one go, operation on only one input neuron is performed with the result stored at the output accumulator. After each operation, both the input (cyan) and weight (red and green) registers are right shifted to perform operation on next input neuron. By cycling through all input neurons, the output accumulators will contain the equivalent result from single MAC operation on all input nodes. The bTanH activation function can then be applied to determine the two 1-bit Dense1 neuron output (Orange and purple). Similar concept is applied for Dense2, with the difference that sigmoid activation function is used instead. With these approaches, only single XNOR per output neuron is needed, which significantly reduces the resources and routing parasitic and leads to lower power dissipation.

ALGORITHM 1: 2D CONP on 1D CCB

Input: load, flattenBinEcgImage
Output: 1D circular buffer

BR ← Obtain the 1D index of the top-left 4×4 bCONP filter while the filter is at the bottom right of a 2D image
RM ← Obtain the 1D indexes of the top-left 4×4 bCONP filter while the filter is on the right most of a 2D image

if load **then** // Begin synchronous block with priority
  Initialise 1D circular buffer with flattenBinEcgImage
  Initialise count to 0
else if (count **equals** BR) **then**
  Rotate registers back to its origin
  Initialise count to 0
else if (count **does not exists in** RR) **then** // Yet to reach end of column (Y)
  Rotate 1D circular buffer by 2
  Increment count by 1
else if (count **exists in** RM) **then** // End of column (Z), shift by COLUMNS + FILTERWIDTH
  Rotate 1D circular buffer to point to the start of the next 2 rows (imaginary 2D)
  Initialise count to 0
end if // End synchronous block with priority

Fig. 10.: Algorithm of 2D CONP on 1D CCB

D. Block Integration

The overall RTL block-level bCNN architecture is shown in Fig. 9. It consists of an input layer for buffering the input image, bCONP Topology for convolution and pooling, Memory1 for storing bCONP output and pre-trained weights of the fully connected binary Dense1 (bDense1) network, Memory2 for storing the bDense1 output and pre-trained weights of the fully connected binary Dense2 (bDense2) network, and a bCNN finite state machine (FSM) to coordinate various blocks within bCNN.

The input block consists of an x-bit Convolution Circular Buffer (CCB) where the x-bit refers to the number of input bits. It can be configured for BP model (320-bits) and LP model (100-bits). The main role of the CCB is to store and rotate the buffer based on the input needs of bCONP as described in Section III.B. The algorithm is shown in Fig. 10. The 4-Convolution Block Selector selects the desired 1D-registers and rearranges it into a 2D-register. As it only involves indexing, no additional registers or clock cycles are incurred. With this arrangement, it is the input image that strides through the convolution filter. However, the result is identical to conventional implementation where the convolution filter strides through an image.

Each bCONP PE is shaded with a green background containing four 9-bit XNOR, four 9-bit parallel adders, four comparators, and one 4-input OR gate. Given the low complexity of the bCONP block, it can be easily duplicated depending on the desired number of convolution filters. For 16×20 and 10×10 binary images, it takes 63 and 16 cycles, respectively, to complete. The cycles are estimated as follows:

$$Cycles = \frac{\text{ROWS}-2}{2} \times \frac{\text{COLUMNS}-2}{2} \quad (4)$$

The results generated from the bCONP PE are stored in the Dense1 Circular Buffer 1 (D1CB). The D1CB is constructed to fit the output from each bCONP to simplify the subsequent flattening operation.

ALGORITHM 2: Dense Circular Buffer

Input: enable, load, binaryData
Output: First data index of 1D circular buffer

if enable **then** // Begin synchronous block with priority
  if load **then**
    Initialise 1D circular buffer with binaryData
  else
    Rotate 1D circular buffer by one
  end if
end if // End synchronous block with priority

Fig. 11.: Algorithm of DCB for Dense1 and Dense2

TABLE I
THEORETICAL PERFORMANCE OF OUR MODELS

Size	Type/ Layer OP	NODES	Weight (Bits)	bMAC OP	REGs	Clock Cycles
16 x 20	Conven. bCNN	402	65729	101873	21168[1]	252[1]
	Our BP					
	Input	320	-	-	320	-
	CONP	16	144	36288	1008	63
	DEN1	65	65520	65520	715	1008
	DEN2	1	65	65	8	65
	IMPR	-	-	-	21.0×[1]	4.0×[1]
10 x 10	Conven. bCNN	150	9009	27153	10752[1]	64
	Our LP					
	Input	100	-	-	100	-
	CONP	32	288	18432	512	16
	DEN1	17	8704	8704	170	512
	DEN2	1	17	17	6	17
	IMPR	-	-	-	21.0×[1]	4.0×[1]

[1] Improvement based on CONP layer only
OP = Operations
IMPR = Improvements

BP		Predicted	
		0	**1**
Actual	**0**	46636	888
	1	553	5795

LP		Predicted	
		0	**1**
Actual	**0**	47012	512
	1	1232	5116

Fig. 12.: Confusion Matrix of BP and LP

In bDense1, the number of PE is determined by the number of output nodes, indicated as the shaded blue region in Fig. 9. The D1CB will rotate based on the number of input nodes in the flatten layer according to algorithm shown in Fig. 11. Hence, it will take the same number of clock cycles as the number of flatten nodes to complete the whole MAC operation. As the output node needs to store the summed result, the number of bits (K) required for output register can be estimated as follows:

$$K = \text{Ceil}(\log_2(x)) \quad (5)$$

where x is the number of flattened input nodes. At the final cycle, the accumulator output will be sent to bTanH activation

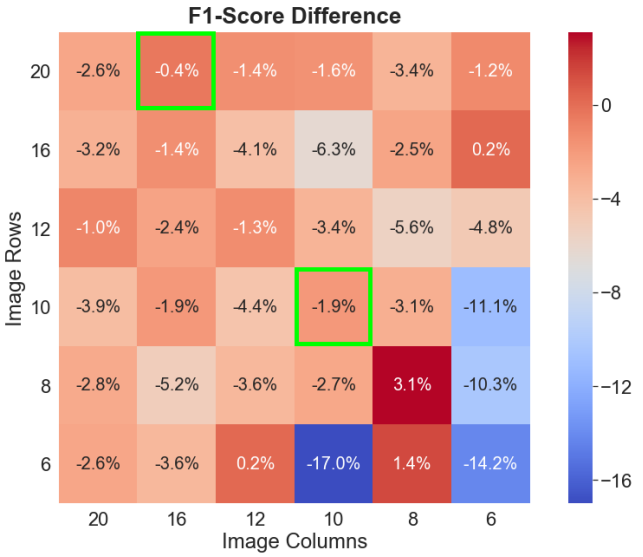

Fig. 13.: Difference of F1-Score between greyscale-floating engine vs bCNN

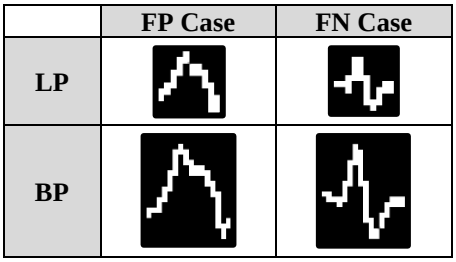

Fig. 14.: FP and FN cases for both bCNN and float-greyscale engines

function to determine the binary output needed for Dense2. Similar operation is performed for bDense2.

IV. Implementation And Results

A. Algorithm Accuracy and Model Complexity

The bCNN model is evaluated against other works using standard metrics, such as accuracy (ACC), sensitivity (SEN), specificity (SPE), positive predictivity (PPR), and are defined as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

$$Sensitivity = \frac{TP}{TP+FN} \quad (7)$$

$$Specificity = \frac{TN}{TN+FP} \quad (8)$$

$$Precision = \frac{TP}{TP+FP} \quad (9)$$

$$F1-score = \frac{2\times TP}{2\times TP+FP+FN} \quad (10)$$

where TP, TN, FP, FN are true positive, true negative, false positive, and false negative, respectively.

Based on the methodology outlined in Section II.B for model selection, the corresponding model parameters for BP and LP models are tabulated in Table I. Compared to conventional bCNN implementation, our BP and LP models achieve 21× improvement in register saving while reducing the number of clock cycles by 4×. The performance metrics for our proposed models is derived from the confusion matrix of Fig. 12 and are compared against others in Table II. As shown, our BP results are comparable to other reported hardware implementation. Although our LP results are slightly poorer, the model complexity is much reduced compared to BP model as shown in Table III. Both models are implemented with iCE40UP5k FPGA. As shown, our LP model in general achieve 2× improvement in both power and hardware resources compared to our BP model.

TABLE II
Comparison Of Evaluation Metrics

V Beat	ETCI 2010 [51]	TBME 2016 [33]	TBCAS 2019 [35]	Our BP	Our LP
ACC	97.7	99.0	98.8	97.3	96.8
SEN	85.8	93.9	85.6	91.3	80.6
SPE	98.7	98.9	99.8	98.1	98.9
PPR	86.6	90.6	96.8	86.7	90.9
F1	86.2	92.2	90.9	88.9	85.4

BP = Better Performance
LP = Low Power

TABLE III
Performance Of Our Models For Target FPGA ICE40UP5K

Model	REG (bits)	LUTs	Clock Cycles	Dynamic Power @ 100kHz (µW)	Total Power (µW)
BP	2104	2873	1141	26.6	404.4
LP	855	1210	550	10.5	227.3
IMPR	2.5×	2.4×	2.1×	2.5×	1.8×

We also conducted a study to observe the variation of classification accuracy between our bCNN implementation and non-resource aware implementation containing greyscale input with floating-point engines. In Fig. 13, it is found that there is a slight performance degradation as highlighted in green in F1-Score, about -0.4% and -1.9% for BP and LP, respectively.

Furthermore, we also conducted another study to observe failure cases, FP and FN, by our BP and LP classifier. On the left column of Fig. 14 (Record 202, Beat 688), FP happens because the image appears to have wide QRS complex. On the right column (Record 200, Beat 1741), the FN case is due to the classifier observing narrow QRS complex. To validate that this is not a binary input or engine issue, we also observe similar classification output for greyscale input with floating engines.

B. Algorithm Complexity

We also compare our algorithm complexity with other reported ECG anomaly detection works employing CNN in Table IV. Most are employing 1D-CNN with 8 to 11-bit input resolution and larger input size. In addition, floating point multiplication is employed.

For CNN, the key operation is MAC. As shown, our BP and LP models require the least number of MAC operations, achieving 7.3× and 27.6× reduction for BP and LP, respectively. This is despite the fact that 2D-CNN is employed in our model, which usually is much more computationally intensive. This is clearly evident by comparing [31] which employs 2D-CNN with other 1D-CNN. The required MAC is about 60× more. The achieved reduction in MAC operation for

TABLE IV
COMPARISON OF COMPUTATIONAL COMPLEXITY OF CNN ALGORITHM FOR ECG CLASSIFICATION

	TBME 2016 [33]	ACCESS 2019 [34]	TBCAS 2019 [35]	IRBM 2021 [31]	Our BP	Our LP
Convolution Type	1-D	1-D	1-D	2-D	2-D	2-D
No. of Convolution Layers	3	2	2	3	1 (↓2×)	1 (↓2×)
Input Sample Resolution	11-BITS	11-bits	11-bits	8-bits	1-bit (↓8×)	1-bit (↓8×)
No. of Input Samples	512	200	400	64×64	16×10	10×10 (↓2×)
No. of Kernels	64	96	48	170	16 (↓3×)	32 (↓1.5×)
Largest Kernel Size	1×15	1×5	1×15	2×2	3×3	3×3
No. of Fully Connected Layers	2	2	2	2	2	2
Activation	N. A	ReLU	N. A.	ReLU	bTanH	bTanH
Output Classes	5	4	5	5	2	2
Multiplication Precision	float-32	float-32	float-32	float-32	1-bit (↓32×)	1-bit (↓32×)
No. of MACs	929,650	1,289,312[1]	749,620	58.1 M [1]	101,873 (↓7.3×)	27,153 (↓27.6×)
ACC (%)	98.9	99.8	98.4	99.7	97.3	96.8

[1] The value is estimated based on the model, ↓ = decreased by, ↑ = increased by

TABLE V
FPGA PERFORMANCE EVALUATION FOR NEURAL NETWORK BASED BEAT CLASSIFICATION

Type	TBIOCAS 2020 [14]	NCA 2020 [52]	ISCAS 2017 [50]	Our BP	Our LP
Technology (ASIC/FPGA)	Zynq XC7Z020	Artix7	Zynq XC7Z020	iCE40UP5k	
Multiplication Precision	24-bit Fixed Point			1-bit (↓24×)	1-bit (↓24×)
Network Type	Multi-layer Perceptron			2D-CNN	2D-CNN
Network Size	Input: 96 Hidden: 32 & 16 Output: 5	-	Input:10 Hidden:6 Output:1	Input:320 Filter:16 Dense1:65 Output / Dense2:1	Input:100 Filter:32 Dense1:17 Output / Dense2:1
Inherent Feature Extractor	×	×	×	√	√
DSP Blocks	0	214	32	0	0
LUTs	5,269[1]	7,298[1]	1,932[1]	2,873[2]	1,210[2] (↓1.6×)
REGs (bits)	1,331	2,474	1,926	2,104	855 (↓1.6×)
SRAM (bits)	5,8624[4]	-	2112	65,585	8721
Max. Frequency (MHz)	-	98.2	100	29.9	32.2
Max. Giga OPS	-	-	0.067[4]	38.3 (↑571.6×)	82.4 (↑1,229.8×)
Clock Cycles Per Classification	-	-	99	1141	550
Operating Clock (Hz)	-	98.2 M	100 M	100 k	100 k
Power (µW)	-	-	124,000[3]	404.4 (↓254.7× FPGA)	227.3 (↓453.1× FPGA)
Energy Per Classification (nJ)	-	-	122.8	320.6	62.5 (↓2.0×)
ACC (%)	99.7	95.0	99.8	97.3	96.8
Output Classes	5	2	2	2	2
Size (mm^2)	13×13	15×15	13×13	2.15×2.55 (↓30.8×)	

[1] 6-input LUT, [2] 4-input LUT, [3] Taken from thesis by the same author on the same work, [4] Estimated based on model parameters, OPS = Operations Per Second, ↓ = decreased by, ↑ = increased by

our proposed BP and LP is the key that allows us to implement the whole CNN algorithm on a compact FPGA. However, lowering computational complexity and resources comes at a cost of reduced accuracy.

C. Hardware Resource Usage

We benchmark our work with other hardware-based ECG anomaly detection implementations in Table V. It should be noted that our adopted FPGA is the most compact one with size of only 2.15×2.55 mm^2. In addition, we are also the only one that leverage on 2D CNN, which eliminates the need for feature engineering for FPGA implementation. Other hardware-based works adopt multi-layer perceptron (MLP). Hence, significant feature engineering might still be needed to achieve good classification metrics, which is not included in their implementations.

In terms of LUT and register usage, our LP achieves at least 1.6× saving compared to others. It should be pointed out that our BP and LP models do not need DSP for fixed-point arithmetic, which is usually computationally expensive.

For power consumption, our BP and LP models only consume dynamic power of 26.6 and 10.5-µW at 100 kHz clock in theory. Unfortunately, there is large static power due to transistor leakage. Hence, in practice, our BP and LP models consume 404.4 µW and 227.3 µW, respectively. Nevertheless, these number is still about 250× to 450× lower than others. In contrast, the larger FPGA employed by other works due to their higher resource requirement (DSP, LUT, registers) suffers from

much higher static power. In terms of energy per classification, our BP and LP model consume 320.6 nJ and 62.5 nJ per classification, respectively. The LP model is about 2× more energy-efficient than [50].

Although our BP and LP model are binary classifiers, we also extended the number of classification classes to 5 while operating at 100 kHz to understand the power behavior. It is found that the incremental power is minimal, at 1.397 µW and 892 nW for BP and LP, respectively.

Although only bCNN is presented in this paper, a complete end-to-end pipeline requires 3 additional stages prior to the bCNN classifier, which is a) the ECG AFE for DC blocking, anti-aliasing filtering, dynamic range alignment and signal buffering, b) QRS detector, c) sample to image converter. With reference to the ECG-On-Chip with QRS detector [1], the AFE consumes 0.75 µW and the digital QRS detector consumes 1.5 µW, whereby both engines require 5.74 mm^2. As mentioned, we have proposed and implemented a hardware-based sample to image converter to facilitate the usability of the proposed bCNN architecture in real-life scenario. The result shows that the accuracy of classification does not vary much and the estimated number of additional MACs required for image conversion is about 14~28 k for LP and BP, respectively, which corresponds to 1.9~3.6× lesser MAC than its bCNN counterpart. However, after the addition of the hardware image converter, the total number of coefficients in the SRAM has increased by 4.4~13.9× for BP and LP, respectively. Both hardware image converter and classifier can fit within the 5k LUT of the FPGA fabric for both BP and LP configurations. In terms of dynamic power, there is about an additional 25~30 µW of power consumption for BP and LP, respectively. The static power remains similar for BP but increases significantly for LP, probably due to more activated LUTs and additional memory nodes for coefficients. As mentioned, the static power can be reduced significantly with ASIC implementation.

V. Conclusion

In this paper, we proposed a compact bCNN model for ECG anomaly detection at the edge device. Within the bCNN, we propose function-merging and block-reuse techniques, which lead to significant resource and power savings. In our implementation, our BP model achieves comparable classification metrics compared to other works while LP model is more resource and power efficient. The BP and LP model can fit into a compact FPGA containing only 5k LUTs and REGs while achieving dynamic power consumption of 26.6 and 10.5-µW, respectively.

Acknowledgments

The authors would like to thank Prof. Thean Aaron Voon-Yew and his team, for their supervision and discussion in this work.

References

[1] R. Suzman and J. Beard, “Global Health and Aging,” *World Health Organization*, pp. 1–32, 2011.

[2] World Health Organization, “Cardiovascular diseases (CVDs),” *WHO Newsroom*, 2021. Accessed: Aug. 14, 2021. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds).

[3] S. S. Virani *et al.*, “Heart Disease and Stroke Statistics—2021 Update,” *Circulation*, vol. 143, no. 8, pp. 1–6, Feb. 2021.

[4] E. P. Gerstenfeld and T. De Marco, “Premature Ventricular Contractions,” *Circulation*, vol. 140, no. 8, pp. 624–626, 2019.

[5] T. Yang, L. Yu, Q. Jin, L. Wu, and B. He, “Localization of Origins of Premature Ventricular Contraction by Means of Convolutional Neural Network from 12-Lead ECG,” *IEEE Trans. Biomed. Eng.*, vol. 65, no. 7, pp. 1662–1671, 2018.

[6] D. L. T. Wong *et al.*, “An Integrated Wearable Wireless Vital Signs Biosensor for Continuous Inpatient Monitoring,” *IEEE Sens. J.*, vol. 20, no. 1, pp. 448–462, Jan. 2020.

[7] C. J. Deepu, X. Zhang, W. S. Liew, D. L. T. Wong, and Y. Lian, “An ECG-on-Chip With 535 nW/Channel Integrated Lossless Data Compressor For Wearable Sensors,” *IEEE J. Solid-State Circuits*, vol. 49, no. 11, pp. 2435–2448, 2014.

[8] C. J. Deepu, X. Y. Xu, D. L. T. Wong, C. H. Heng, and Y. Lian, “A 2.3 uW ECG-On-Chip for Wireless Wearable Sensors,” *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 65, no. 10, pp. 1385–1389, 2018.

[9] C. J. Deepu, X. Zhang, C. H. Heng, and Y. Lian, “A 3-Lead ECG-on-Chip with QRS Detection and Lossless Compression for Wireless Sensors,” *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 63, no. 12, pp. 1151–1155, Dec. 2016.

[10] K. A. Ng *et al.*, “A 3-Mbps, 802.11g-Based EMG Recording System With Fully Implantable 5-Electrode EMG Acquisition Device,” *IEEE Trans. Biomed. Circuits Syst.*, vol. 14, no. 6, pp. 1441–1441, Dec. 2020.

[11] C. B. Gungor, P. P. Mercier, and H. Toreyin, “A 1.2nW Analog Electrocardiogram Processor Achieving a 99.63% QRS Complex Detection Sensitivity,” *IEEE Trans. Biomed. Circuits Syst.*, vol. 15, no. 3, pp. 617–628, Jun. 2021.

[12] Y. Luo, K.-H. Teng, Y. Li, W. Mao, Y. Lian, and C.-H. Heng, “A 74-µW 11-Mbps Wireless Vital Signs Monitoring SoC for 3-Lead ECG, Respiration Rate, and Body Temperature,” *IEEE Trans. Biomed. Circuits Syst.*, vol. 4545, pp. 1–12, 2019.

[13] X. Zhang, Z. Zhang, Y. Li, C. Liu, Y. X. Guo, and Y. Lian, “A 2.89 µw Dry-Electrode Enabled Clockless Wireless ECG SoC for Wearable Applications,” *IEEE J. Solid-State Circuits*, vol. 51, no. 10, pp. 2287–2298, 2016.

[14] Y. Zhao, Z. Shang, and Y. Lian, “A 13.34 µ W Event-Driven Patient-Specific ANN Cardiac Arrhythmia Classifier for Wearable ECG Sensors,” *IEEE Trans. Biomed. Circuits Syst.*, vol. 14, no. 2, pp. 186–197, Apr. 2020.

[15] Y. Jiang, P. Huang, Z. Zhou, and J. Kang, “Circuit design of RRAM-based neuromorphic hardware systems for classification and modified Hebbian learning,” *Sci. China Inf. Sci.*, vol. 62, no. 6, p. 62408, Jun. 2019.

[16] X. Yang, Z. Zhang, W. Zhu, S. Yu, L. Liu, and N. Wu, “Deterministic conversion rule for CNNs to efficient spiking convolutional neural networks,” *Sci. China Inf. Sci.*, vol. 63, no. 2, p. 122402, Feb. 2020.

[17] D. L. T. Wong, Y. Li, D. John, W. K. Ho, and C. H. Heng, “Resource and Energy Efficient Implementation of ECG Classifier Using Binarized CNN for Edge AI Devices,” in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2021, pp. 1–5.

[18] X. Zhang and Y. Lian, “A 300-mV 220-nW Event-Driven ADC With Real-Time QRS Detection for Wearable ECG Sensors,” *IEEE Trans. Biomed. Circuits Syst.*, vol. 8, no. 6, pp. 834–843, Dec. 2014.

[19] D. Ngo and B. Veeravalli, “Design of a Real-time Morphology-based Anomaly Detection Method from ECG Streams,” in *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Nov. 2015, pp. 829–836.

[20] B. Veeravalli, C. J. Deepu, and D. Ngo, “Real-Time, Personalized Anomaly Detection in Streaming Data for Wearable Healthcare Devices,” in *Handbook of Large-Scale Distributed Computing in Smart Healthcare*, Cham, Switzerland: Springer, 2017, pp. 403–426.

[21] M. Alfaras, M. C. Soriano, and S. Ortín, “A Fast Machine Learning Model for ECG-Based Heartbeat Classification and Arrhythmia Detection,” *Front. Phys.*, vol. 7, pp. 1–11, 2019.

[22] M. A. Sohail, Z. Taufique, S. M. Abubakar, W. Saadeh, and M. A. Bin Altaf, “An ecg processor for the detection of eight cardiac

arrhythmias with minimum false alarms," *BioCAS 2019 - Biomed. Circuits Syst. Conf. Proc.*, pp. 1–4, 2019.
[23] S. Y. Lee, J. H. Hong, C. H. Hsieh, M. C. Liang, S. Y. C. Chien, and K. H. Lin, "Low-power wireless ECG acquisition and classification system for body sensor networks," *IEEE J. Biomed. Heal. Informatics*, vol. 19, no. 1, pp. 236–246, 2015.
[24] Y. Zhang, Y. Liu, and C.-H. Chen, "Review on Deep Learning in Feature Selection," 2021, pp. 439–447.
[25] T. J. Jun, H. M. Nguyen, D. Kang, D. Kim, D. Kim, and Y.-H. Kim, "ECG arrhythmia classification using a 2-D convolutional neural network," Apr. 2018.
[26] J. T. Ruiz, J. D. B. Pérez, and J. R. B. Blázquez, "Arrhythmia Detection Using Convolutional Neural Models," in *Advances in Intelligent Systems and Computing*, vol. 800, no. June, 2019, pp. 120–127.
[27] S. Kiranyaz, T. Ince, R. Hamila, and M. Gabbouj, "Convolutional Neural Networks for patient-specific ECG classification," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, vol. 2015-Novem, pp. 2608–2611, 2015.
[28] Q. Xie, S. Tu, G. Wang, Y. Lian, and L. Xu, "Feature enrichment based convolutional neural network for heartbeat classification from electrocardiogram," *IEEE Access*, vol. 7, pp. 153751–153760, 2019.
[29] X. Zhai and C. Tin, "Automated ECG Classification Using Dual Heartbeat Coupling Based on Convolutional Neural Network," *IEEE Access*, vol. 6, pp. 27465–27472, 2018.
[30] U. R. Acharya *et al.*, "A deep convolutional neural network model to classify heartbeats," *Comput. Biol. Med.*, vol. 89, no. August, pp. 389–396, 2017.
[31] M. Degirmenci, M. A. Ozdemir, E. Izci, and A. Akan, "Arrhythmic Heartbeat Classification Using 2D Convolutional Neural Networks," *IRBM*, vol. 1, pp. 1–12, Apr. 2021.
[32] Y. Wu, F. Yang, Y. Liu, X. Zha, and S. Yuan, "A Comparison of 1-D and 2-D Deep Convolutional Neural Networks in ECG Classification," *Conf. Proc. ... Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE Eng. Med. Biol. Soc. Annu. Conf.*, vol. 2018, pp. 324–327, 2018.
[33] S. Kiranyaz, T. Ince, and M. Gabbouj, "Real-Time Patient-Specific ECG Classification by 1-D Convolutional Neural Networks," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 3, pp. 664–675, Mar. 2016.
[34] Y. Xia and Y. Xie, "A novel wearable electrocardiogram classification system using convolutional neural networks and active learning," *IEEE Access*, vol. 7, pp. 7989–8001, 2019.
[35] N. Wang, J. Zhou, G. Dai, J. Huang, and Y. Xie, "Energy-Efficient Intelligent ECG Monitoring for Wearable Devices," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 5, pp. 1112–1121, Oct. 2019.
[36] L. G. Rocha *et al.*, "Binary CorNET: Accelerator for HR Estimation from Wrist-PPG," *IEEE Trans. Biomed. Circuits Syst.*, vol. 14, no. 4, pp. 715–726, 2020.
[37] M. Porumb, E. Iadanza, S. Massaro, and L. Pecchia, "A convolutional neural network approach to detect congestive heart failure," *Biomed. Signal Process. Control*, vol. 55, pp. 1–9, Jan. 2020.
[38] Fei Zhang and Yong Lian, "QRS Detection Based on Multiscale Mathematical Morphology for Wearable ECG Devices in Body Area Networks," *IEEE Trans. Biomed. Circuits Syst.*, vol. 3, no. 4, pp. 220–228, 2009.
[39] Chio-In Ieong *et al.*, "A 0.83-μW QRS Detection Processor Using Quadratic Spline Wavelet Transform for Wireless ECG Acquisition in 0.35-μm CMOS," *IEEE Trans. Biomed. Circuits Syst.*, vol. 6, no. 6, pp. 586–595, Dec. 2012.
[40] P. Li *et al.*, "A 410-nW Efficient QRS Processor for Mobile ECG Monitoring in 0.18-μm CMOS," *IEEE Trans. Biomed. Circuits Syst.*, vol. 11, no. 6, pp. 1356–1365, Dec. 2017.
[41] X. Tang, Q. Hu, and W. Tang, "A Real-Time QRS Detection System With PR/RT Interval and ST Segment Measurements for Wearable ECG Sensors Using Parallel Delta Modulators," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 4, pp. 751–761, Aug. 2018.
[42] C. Nayak, S. K. Saha, R. Kar, and D. Mandal, "An Efficient and Robust Digital Fractional Order Differentiator Based ECG Pre-Processor Design for QRS Detection," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 4, pp. 682–696, Aug. 2019.
[43] K. Zhao, Y. Li, G. Wang, Y. Pu, and Y. Lian, "A robust QRS detection and accurate R-peak identification algorithm for wearable ECG sensors," *Sci. China Inf. Sci.*, vol. 64, no. 8, 2021.
[44] W. Yan *et al.*, "A resource-efficient, robust QRS detector using data compression and time-sharing architecture," *Proc. - IEEE Int. Symp. Circuits Syst.*, vol. 2021-May, no. 1, pp. 7–11, 2021.
[45] and Y. L. W. Yan, Y. Ji, C. Ma, L. Hu, Y. Zhao, Y. Li, G. W, "A Computationally Efficient, Hardware Re-configurable Architecture for QRS Detection and ECG authentication," in *ASSCC 2021 - Integrated Circuits and Systems for the Connection of Intelligent Things*, 2021, pp. 1–2.
[46] T. Chen, Z. Zhang, X. Ouyang, Z. Liu, Z. Shen, and Z. Wang, "'BNN - BN = ?': Training Binary Neural Networks without Batch Normalization," pp. 1–11, Apr. 2021.
[47] B. Zhuang, L. Liu, M. Tan, C. Shen, and I. Reid, "Training quantized neural networks with a full-precision auxiliary module," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1485–1494, 2020.
[48] A. Y. Hannun *et al.*, "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network," *Nat. Med.*, vol. 25, no. 1, pp. 65–69, Jan. 2019.
[49] L. Semiconductor, "iCE40 UltraPlus ™ Family Data Sheet," 2017.
[50] M. Wess, P. D. S. Manoj, and A. Jantsch, "Neural network based ECG anomaly detection on FPGA and trade-off analysis," *IEEE Int. Symp. Circuits Syst.*, pp. 1–5, 2017.
[51] Y. Jewajinda and P. Chongstitvatana, "FPGA-based Online-learning Using Parallel Genetic Algorithm and Neural Network for ECG Signal Classification," in *ECTI International Confernce on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2010, pp. 1–5.
[52] H. Zairi, M. Kedir Talha, K. Meddah, and S. Ould Slimane, "FPGA-based System for Artificial Neural Network Arrhythmia Classification," *Neural Comput. Appl.*, vol. 32, no. 8, pp. 4105–4120, 2020.

David Liang Tai Wong (Senior Member, IEEE) received the B. Eng. degree with honours in Computer Systems Engineering from Curtin University of Technology in 2007 and the M. Eng. degree in Integrated Circuits and Embedded Systems from National University of Singapore in 2014.

He was with Panasonic Corporation as a Research and Development engineer cum technical lead from 2008 to 2011. He is currently working as a researcher at National University of Singapore. His research interests include flexible wearable and wireless biomedical devices, low-power embedded hardware and software co-design, data communications and networking. He is actively contributing to the Singapore Section Young Professionals Affinity Group, where he served as the Vice-Chair, in 2016 and 2019, respectively, and Chair from 2017 to 2018. He is currently serving as an IEEE R10 Young Professionals Awards & Recognition Coordinator from 2021 to 2022.

Yongfu Li (Senior Member, IEEE) received the B.Eng. and Ph.D. degrees from the Department of Electrical and Computing Engineering, National University of Singapore (NUS), Singapore.

He is currently an Associate Professor with the Department of Micro and Nano Electronics Engineering and MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University, China. He was a research engineer with NUS, from 2013 to 2014. He was a senior engineer (2014-2016), principal engineer (2016-2018) and member of technical staff (2018-2019) with GLOBALFOUNDRIES, as a Design-to-Manufacturing (DFM) Computer-Aided Design (CAD) research and development engineer. His research interests include analog/mixed signal circuits, data converters, power converters, biomedical signal processing with deep learning technique and DFM circuit automation.

Deepu John (Senior Member, IEEE) received the B.Tech. degree in Electronics & Communication Engineering from the University of Kerala, India, in 2002, and the M.Sc. and Ph.D. degrees in electrical engineering from National University Singapore, Singapore, in 2008 and 2014, respectively. He is currently an Assistant Professor with the School of Electrical and Electronics Engineering, University College Dublin, Ireland. He was a Postdoctoral Researcher with the Bio-Electronics Lab, National University Singapore from 2014 to 2017. Previously, he worked as a Senior Engineer with Sanyo Semiconductors, Gifu, Japan. He is a recipient of the Institution of Engineers Singapore Prestigious Engineering Achievement Award in 2011, the Best Design Award at the Asian SolidState Circuit Conference in 2013, and the IEEE Young Professionals, Region 10 Individual Award in 2013. He served as a member of the organizing committee/technical program committee for several IEEE conferences, including TENCON, ASICON, ISCAS, BioCAS, and ICTA. He served as a Guest Editor for IEEE Transactions on Circuits & Systems-I and IEEE Open Journal of Circuits & Systems. He currently serves as an Associate Editor for IEEE Transactions on Biomedical Circuits & Systems, IEEE Transactions on Circuits & Systems- II and Wiley International Journal of Circuit Theory & Applications. His research interests include low-power biomedical circuit design, energy-efficient signal processing, and edge computing.

Weng Khuen Ho (Senior Member, IEEE) received the B.Eng. and Ph.D. degrees in electrical engineering from the National University of Singapore in 1987 and 1992 respectively. He worked as an engineer at the Chartered Industry of Singapore from 1987 to 1988. He joined the Department of Electrical and Computer Engineering, National University of Singapore in 1992. His research interests include control and signal processing in semiconductor manufacturing and process control. He received the 2003 Best Paper Award, IEEE Transactions on Semiconductor Manufacturing. He has supervised more than 30 graduate students. He served full-time National Service in the Singapore Armed Forces as an infantry officer from 1979 to 1981 and in the army reserve from 1981 till 2012.

Chun-Huat Heng (Senior Member, IEEE) received the B.Eng. and M.Eng. degrees from the National University of Singapore, Singapore, in 1996 and 1999, respectively, and the Ph.D. degree from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2003.

He has been working on CMOS integrated circuits involving synthesizers, delay-locked loop, and transceiver circuits. From 2001 to 2004, he was with Wireless Interface Technologies, which was later acquired by Chrontel. Since 2004, he has been with the National University of Singapore, where he is currently an Associate Professor.

Dr. Heng received the NUS Annual Teaching Excellence Award in 2008, 2011, 2013, and 2021, and ATEA Honor Roll in 2014. He has also won the Faculty Innovative Teaching Award in 2009 and 2020. He has won the 2018 and 2020 IES Prestigious Engineering Award. He was an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II and the IEEE TRANSACTIONS ON VLSI. He has served as a Technical Program Committee Member for the International Solid-State Circuits Conference and the Asian Solid-State Circuits Conference. He currently serves as a Technical Program Committee Member for the VLSI Symposium on Circuits and the IEEE Radio Frequency Integrated Circuit Symposium.