Computers in Biology and Medicine 155 (2023) 106623

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

A lightweight convolutional neural network hardware implementation for wearable heart rate anomaly detection

Minghong Gu [a], Yuejun Zhang [a,*], Yongzhong Wen [a], Guangpeng Ai [a], Huihong Zhang [a,**], Pengjun Wang [b], Guoqing Wang [c,***]

[a] Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo, 315211, Zhejiang, China
[b] Electrical and Electronic Engineering, Wenzhou University, Wenzhou, 325035, Zhejiang, China
[c] Zhejiang Suosi Technology Co. Ltd, Wenzhou, 325000, Zhejiang, China

ARTICLE INFO

Keywords:
Convolutional neural networks
ECG detection
Data reuse
Hardware efficiency

ABSTRACT

In this article, we propose a lightweight and competitively accurate heart rhythm abnormality classification model based on classical convolutional neural networks in deep neural networks and hardware acceleration techniques to address the shortcomings of existing wearable devices for ECG detection. The proposed approach to build a high-performance ECG rhythm abnormality monitoring coprocessor achieves a high degree of data reuse in time and space, which reduces the number of data flows, provides a more efficient hardware implementation and reduces hardware resource consumption than most existing models. The designed hardware circuit relies on 16-bit floating-point numbers for data inference at the convolutional, pooling, and fully connected layers, and implements acceleration of the computational subsystem through a 21-group floating-point multiplicative-additive computational array and an adder tree. The front- and back-end design of the chip was completed on the TSMC 65 nm process. The device has an area of 0.191 mm^2, a core voltage of 1 V, an operating frequency of 20 MHz, a power consumption of 1.1419 mW, and requires 5.12 kByte of storage space. The architecture was evaluated using the MIT-BIH arrhythmia database dataset, which showed a classification accuracy of 97.69% and a classification time of 0.3 ms for a single heartbeat. The hardware architecture offers high accuracy with a simple structure, low resource footprint, and the ability to operate on edge devices with relatively low hardware configurations.

1. Introduction

According to the World Health Organization, tens of millions of people die each year from cardiovascular disease, making it an enemy of mankind that cannot be ignored [1]. Cardiovascular diseases (CVD) are insidious, have a long latency period, and are difficult to cure after onset. Despite significant improvements in research or knowledge of CVD and corresponding advances in diagnostic or therapeutic methods, it remains by far the leading cause of human death. Therefore, it is particularly important to take effective measures to prevent and treat cardiovascular disease. There is growing evidence that increased participation in cardiac rehabilitation significantly improves the prognosis of cardiovascular patients, with some reports suggesting that participation in cardiac rehabilitation programs reduces the risk of long-term hospital readmission by 25% and the risk of death by 42% [2]. The electrocardiogram (ECG) is the most visual means for doctors to help monitor a patient's heart condition during the clinical phase, and specialists can intervene in the early stages of heart disease with the help of real-time ECG monitoring. However, numerous clinical observations have shown that most arrhythmias are sudden or occasional, with no certain pattern in the timing and frequency of attacks, thus requiring long-term continuous monitoring of arrhythmias to capture the moment of occurrence and hence reduce the risk of underdiagnosis [3]. Due to the wide variety and aberrations of pathological ECGs, the same etiology may show differences on the ECG of different patients, even in the same individual. Therefore, cardiologists may require extensive knowledge of

* Corresponding author.
** Corresponding author.
*** Corresponding author.
E-mail addresses: 2111082277@nbu.edu.cn (M. Gu), zhangyuejun@nbu.edu.cn (Y. Zhang), wyz2011082029@163.com (Y. Wen), 494337617@qq.com (G. Ai), zhanghuihong@nbu.edu.cn (H. Zhang), wangpengjun@wzu.edu.cn (P. Wang), wangguoqing79@139.com (G. Wang).

https://doi.org/10.1016/j.compbiomed.2023.106623
Received 30 November 2022; Received in revised form 9 January 2023; Accepted 28 January 2023
Available online 8 February 2023

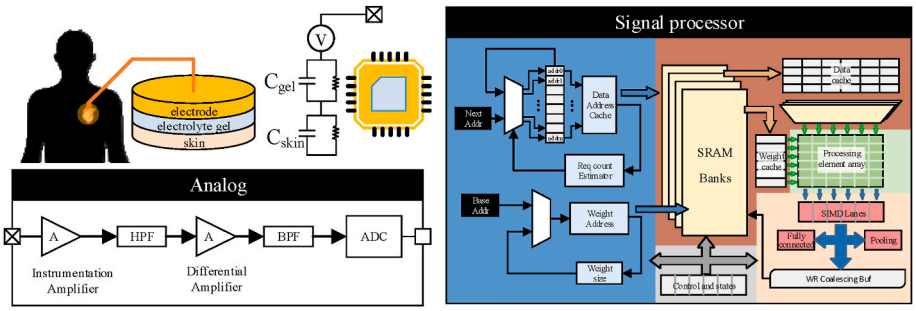


Fig. 1. Portable wearable heart rhythm health monitoring system.

the field and substantial clinical experience to make an accurate and valid diagnosis. It is easy to observe that the identification of abnormal heart rhythms is demanding with a huge workload for cardiologists, which can easily be missed or misdiagnosed when doctors are physically and mentally exhausted, resulting in patients missing out on prime time to be used for treatment. In order to accurately and efficiently determine the type of heart rhythm abnormality, it is widely expected that artificial intelligence (AI) technology will be applied to classify heart rhythm abnormalities, thereby alleviating some of the doctors' workload and saving patients' lives in a timely manner. Therefore, research work on wearable ECG smart monitoring devices has a profound contemporary and social value. Considering the recent boom in AI technology and machine learning algorithms, we would like to provide a reliable AI algorithm for ECG detection based on AI technology for hardware integration of wearable health monitors. We would like to contribute to the construction of medical equipment in the rehabilitation system for cardiovascular diseases.

This study focuses on the design of a signal analysis processor, while a complete portable heart rhythm monitoring system, consisting of a heart rhythm monitoring sensor, an analog circuit and a signal analysis processing circuit, is shown in Fig. 1. In the ECG sensor section, there are wet electrode sensors with a common electrode-electrolyte gel-skin structure, dry sensors that rely on sweat as an intermediate medium, and fabric sensors consisting of strain-relief fabric and metal yarns [4]. In this system, the analog circuit mainly plays the role of amplification and filtered signal regulation. The first stage amplifier has a high input impedance to achieve low noise, and the second stage uses a differential amplifier circuit and feeds into an ADC for acquisition. After this, data transmission and data analysis are the main focus of this study.

The technology of deep learning has been developing in recent years, and the research of diagnostic and therapeutic models based on medical data and images has gained wide attention [5]. Chen et al. [6], Yu et al. [7], and Guan et al. [8] reported that deep learning applied to low-quality iris images, thyroid ultrasound image, and pancreatic tumor images for recognition of pancreatic tumor images, respectively. A reported the application of deep learning for the recognition of low-quality iris images, thyroid ultrasound images, and pancreatic tumor images, respectively. number of these researchers have also used deep learning to achieve medical image enhancement, such as the ref [8] using a texture-constrained multichannel progressive generative adversarial network (TMP-GAN) to improve the fidelity of synthetic images. As well as a dynamic step shuffled frog leaping algorithm is used to optimize the performance design and feature selection to solve the problem of getting stuck in local optimization or poor search ability [9]. Several artificial intelligence algorithms for detecting and classifying ECG signals are already available. The most commonly used methods are Support Vector Machines (SVM), K-Nearest Neighbor algorithm (KNN), Principal Component Analysis (PCA), logistic regression, decision tree models and Deep Convolutional Neural Networks (DCNN). Classical algorithms such as SVM and KNN have been proposed as early as the last century, and these types of algorithms have the advantage of being simple and efficient, with certain advantages in terms of hardware implementation. These algorithms are often used to handle binary classification tasks, such as rating seizure detection [10], and (COVID-19) performed well for mask detection during the pandemic. However, such algorithms do not classify well when using a cascading binary tree approach [11] for multi-classification tasks. In Ref. [12], which shows that a novel regularization method to minimize the signal-to-noise ratio to improve the robustness of DNNs to counteract noise in ECG signal classification applications. Ref. [13] proposed a transformer-based deep learning neural network, which focuses more on the dependencies between heartbeats. This method is a more continuous arrhythmia detection algorithm compared to single beat classification methods. Exarchos et al. proposed a method to automatically create a fuzzy expert system for ischemic and arrhythmic heartbeat classification, but only three types of people were found to fit the model through experiments [14]. Carrault et al. used inductive logic programming and collaborative offline debugging to achieve heartbeat recognition [15], but this approach has difficulty in resolving the effects of noise and ambiguous data on the ECG signal. Shen et al. proposed an ECG arrhythmia detection system using adaptive feature selection and improved SVM [16], but the amount of data equivalence was achieved by replication during experiments, which increased the amount of classification computation. After registering ECG signals using the Pan Tompkins algorithm, the GMM-based classifier and the principal components extracted from ECG features by PCA can also classify ECG signals [17]. Liu used a dictionary learning algorithm to encode the ECG heartbeat, which effectively reduced the computational complexity of the ECG heartbeat classification system and maintained a high accuracy rate. In addition, he optimized the ECG classification system based on encoded features [18].

Convolutional neural networks (CNNs) are widely used in artificial intelligence applications and several efficient models have been proposed that offer unique advantages in detecting cardiac rhythm abnormalities. For example, direct recognition of ECG waveform images for classification using 2D-CNN achieved 99.05% classification accuracy and 97.85% average sensitivity on the MIT-BIH arrhythmia database. In contrast, 1D-CNN is more concise compared to 2D-CNN, fusing the feature extraction amount and classification of traditional ECG classification into one learning body [19]. Analyzing ECG detection with traditional CNN requires GPU accelerators, the memory requirements of standard frameworks such as TensorFlow or Pytorch, and the computational requirements of CNNs, so although many works have proposed CNN models with high accuracy and low computational complexity [20], these models are not easy to implement on wearable devices. Moreover, there are also articles with many convolutional layers and complex networks to achieve higher accuracy [21–23], and complex networks are not easy for hardware implementation because hardware implementation of neural networks needs to follow the rules of circuit design. This paper proposes a low-power CNN hardware architecture for ECG classification with good performance in terms of resources and hardware consumption. The system is highly portable, cost-effective and lightweight and can be used as a home heart monitoring device or a patient-provided heart rhythm screening device.

The design of neural network accelerators becomes necessary as the

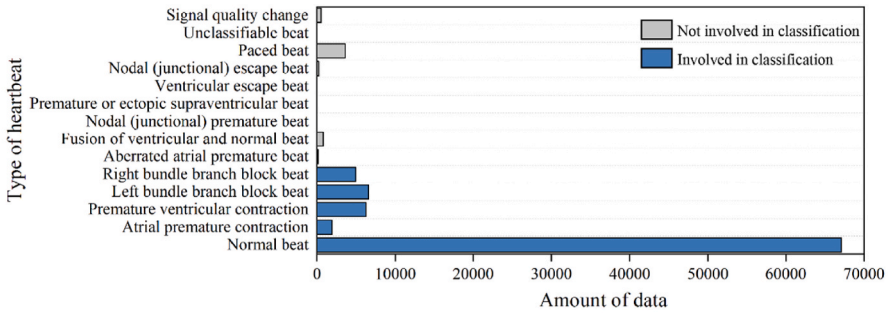


Fig. 2. Dataset heartbeat number statistics of various classes.

objects targeted by algorithmic models become diverse and the amount of data becomes complex. Hardware accelerators can improve the efficiency of data computation, and various hardware platforms are also widely used for automatic driving image recognition processors [24], and biological signal processors [25], including microcontrollers, FPGAs and ASICs. FPGAs are more suitable for portable wearable devices because of their configurable capabilities, while microcontrollers can perform more complex algorithms than FPGAs, and ASICs can be more efficient than FPGAs to optimize power consumption and computing speed.

This article is organized as follows. Section 2 presents the ECG database used in this study. Section 3 introduces our proposed method. Section 4 gives a description of the experiments and a discussion of the results, respectively. Finally, the conclusions of this paper are summarized in Section 5.

2. ECG databases

The proposed architecture was trained, validated, and tested using the MIT-BIH Arrhythmia Database dataset available on the PhysioNet portal. A Research Resource for Complex Physiological Signals was established under the auspices of the National Center for Research Resources (NCRR) of the National Institutes of Health (NIH) to facilitate current research and new investigations in the study of cardiovascular disease and other complex biomedical signaling research. It currently includes a multi-parametric database of cardiopulmonary, neurological, and other biomedical signals from healthy subjects and patients with multiple diseases that have significant public health implications, including life-threatening arrhythmias, atrial fibrillation, neurological brain signals, chest radiation data, and more. The data used in this paper were obtained from a total of 48 ECG recordings from 25 male volunteers between the ages of 32 and 89 and 22 female volunteers between the ages of 23 and 89 studied at the BIH Arrhythmia Laboratory between 1975 and 1979 [26]. Each recording contained ECG signals acquired at 360 Hz for a duration of about 30 min. The upper part is a modified limb lead with the electrode placed on the chest, in which the QRS wave group is more prominent. The lower signal comes from the thoracic lead, which is almost orthogonal to the electrical axis of the heart, and it is difficult to identify a normal heartbeat in this lead. Therefore, it is more appropriate to classify the heartbeat using the limb leads. The most important contribution of this database to this study is its labeling of ECG signals by heartbeat type.

The purpose of this study was to classify single heartbeats. The target heartbeats for classification were divided into the following five types: normal beat (N), atrial premature contraction (A), premature ventricular contraction (V), Right bundle branch block fluctuation (R), Left bundle branch block fluctuation (L). In fact, that dataset was more annotated by cardiologists. MIT divided all the collected heartbeats into 14

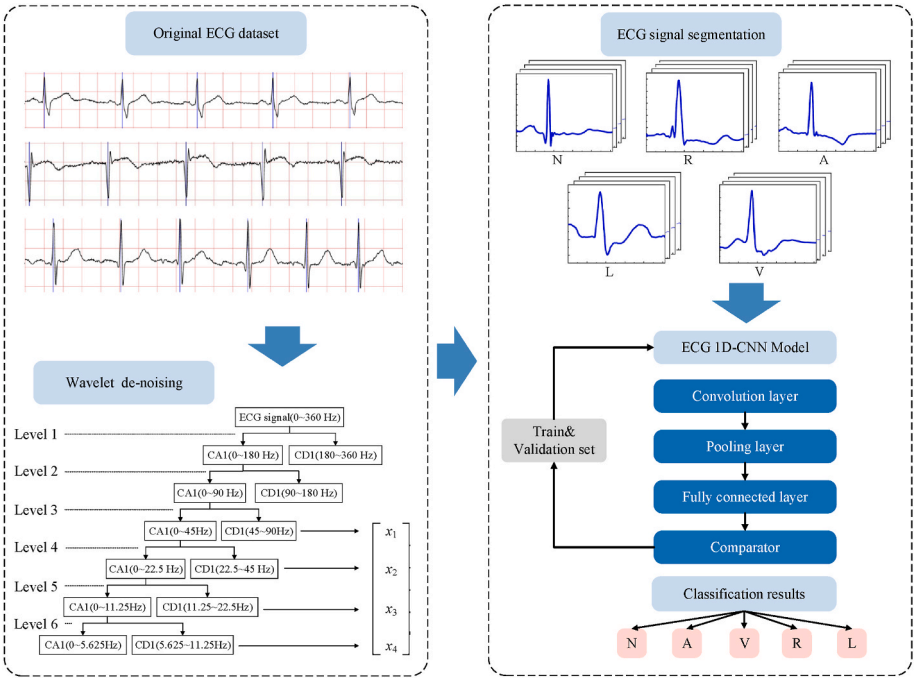


Fig. 3. Wavelet de-noising and heartbeat segmentation methods to make the dataset.

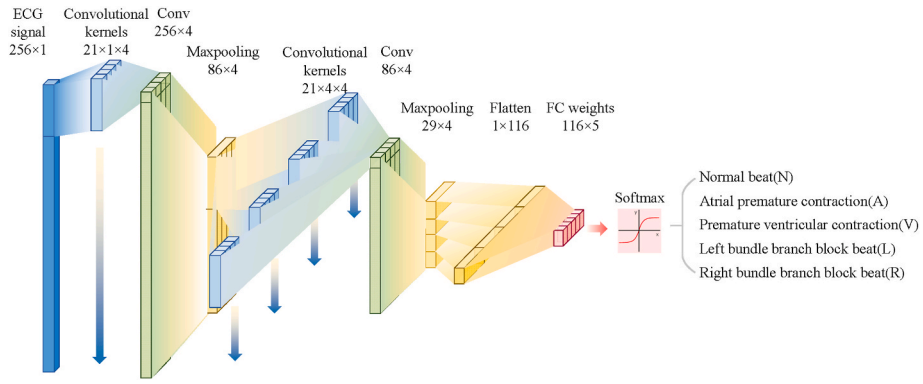


Fig. 4. Structure of the abnormal heart rhythm detection model.

sub-categories, and the statistics of the number of various heartbeat types in the dataset are shown in Fig. 2. The above five categories were chosen for model training because of the large amount of data available for that heartbeat type and because of the threat to human life and health posed by that abnormal heartbeat type. In this work, we augment certain arrhythmia types with low sample size by varying both noise level and noise density based on the MIT-BIH arrhythmia public dataset. Arrhythmia noise samples with different noise levels and different noise densities are generated based on the original samples in the dataset, while the points where the noise occurs are generated in a randomized manner.

Firstly, Db6 wavelet decomposition was performed on the data with a sampling frequency of 360 Hz. The key information and frequency components whose central beats are worth extracting are between 45 and 90 Hz, besides the presence of baseline drift below 5 Hz and possible high-frequency noise above 90 Hz caused by clothing friction, environment, etc. Therefore, the wavelet coefficients below 5.625 Hz and above 90 Hz were replaced by zero and only the coefficients between the 3rd and 6th detail sub-band coefficients were retained and used for reconstructing the signal [27]. Due to the small amount of data for some categories, we used sample replication to supplement the training set. Then, based on the ECG label attached to the data file, 86 data points before the R-wave and 170 data points after the R-wave were intercepted, with a total length of 256 data points. The ECG signals were Z-score normalized before being fed into the AI algorithm. The formulas used for Z-score normalization are shown in (1) and (2).

$$Z = \frac{X - \mu}{\sigma} \tag{1}$$

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}|x_i - \mu|^2} \tag{2}$$

where μ and σ are the mean and standard deviation of the ECG data. Z-score normalization is the conversion of an original score to a standard score, making AI training easier and reducing errors in the data conversion process. The production process of the data set is shown in Fig. 3.

3. 1D-CNN abnormal heart rhythm detection method

Chapter 2 describes the heartbeat classification algorithms proposed by some researchers. Experiments have shown that these methods are effective in recognizing abnormal heart rhythm databases. However, implementing these models in hardware requires a significant resource cost. Therefore, this chapter optimizes the previously proposed models and aims to realize an efficient hardware neural network abnormal heart rhythm detection model.

3.1. Abnormal heart rhythm detection algorithm

We implemented feature extraction and accurate classification of ECG signals based on the classical CNN model in deep neural networks. A classic CNN is arranged layer by layer, and each layer takes the calculation results of the previous layer as input, which can provide data for the calculation of the latter layer. Depending on how the layers are computed, we can classify these layers into three categories: convolutional(CONV) layers, pooling layers and fully connected(FC) layers. In a CNN, the CONV layer must be the most important. In the convolution operation, each CONV layer has a number of trained convolution kernels, with each CONV kernel connected to a small region of the input data. The corresponding pixel positions of the connections are multiplied one by one, and all the multiplier results in that area are summed to obtain a pixel result of the convolution output. The convolution kernel slides over the input data according to a certain stride, and the convolution kernel swipes across all the input data to obtain the output result of the convolution layer. As the number of CONV layers deepens, all partial features are gradually integrated into higher-level features, and finally the result representing the original input data category is achieved. The role of the pooling layer is to compress the features and plan the size of the feature map, and the negligible pixels in the feature map can be removed by pooling. Two types of pooling are widely used in CNNs: Maxpooling and Averpooling. Maxpooling is to retain the largest number of features in the perceptual window, while average pooling is to average the values in the perceptual window. The FC layer is generally in the last part of the entire model process. This part of the operation contains a large number of weights. Its function is to flatten the results obtained from the previous convolution and pooling, and map the weights to the sample labels.

The 1D-CNN model proposed in this paper is shown in Fig. 4. The pre-processed 256 × 1 ECG heartbeat signal is used as input, and the classification results are output, namely five kinds of Normal beat, Atrial premature contraction, Premature ventricular contraction, Left bundle branch, which is one of the heartbeat types of block beat and Right bundle branch block beat. The entire CNN architecture consists of 2 CONV layers, 2 pooling layers and 1 FC layer. The value in the first CONV layer is convolved with the input data via four 21 × 1 large-size convolutional kernels with trained weights. To fully collect the features of the input ECG waveform, we fill the input data with zeros and make the center of the convolution kernel coincide with the edge of the ECG signal. The size of the output data after convolution is the same as that of the input data. In addition, the length of the convolution kernel in the first and second layers of the model proposed in this paper is the same. The purpose of this is to facilitate the module calls in the hardware circuit design while ensuring the classification accuracy.

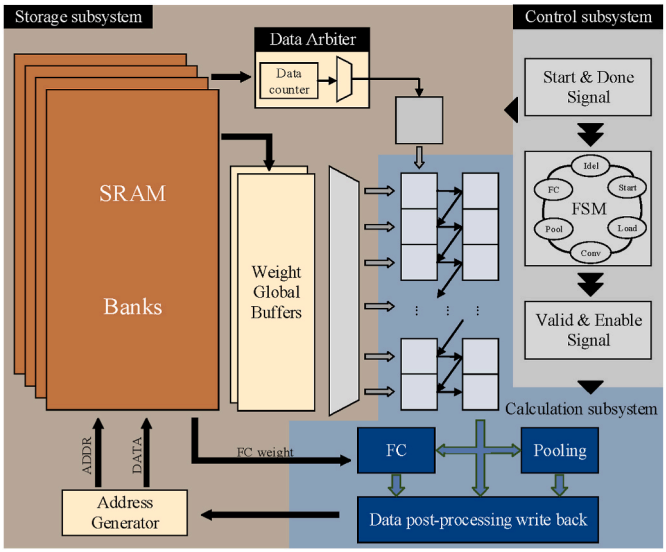


Fig. 5. Abnormal heart rhythm detection hardware architecture.

Table 1
Network data volume statistics.

Data bit width	Data name	Data size
16 bit	Input data	256 × 1
	The weights of the first convolutional layer	21 × 4
	The bias of the first convolutional layer	4
	The weights of the second convolutional layer	21 × 16
	The bias of the second convolutional layer	4
	The result of the first convolutional layer	256 × 4
	The result of the first pooling layer	86 × 4
	The result of the second convolutional layer	86 × 4
	The result of the second pooling layer	29 × 4
	The weights of the fully connected layer	116 × 5
	Summary	3092

3.2. Abnormal heart rhythm detection hardware implementation

The abnormal heart rhythm detection system can be roughly divided into three parts: the storage subsystem, the calculation subsystem and the control subsystem. The overall architecture of the hardware implementation is shown in Fig. 5.

3.2.1. Storage subsystem

In the process of neural network inference, a large amount of storage space is required to cache data during inference, so the case of storage splitting needs to be considered. The data used for inference can be divided into three categories: input data, weights and output data (including the results of each layer's output). The data throughput of each layer is shown in Table 1. In the process of neural network inference, as the number of weights in inference increases, the output data decreases. Therefore, a dual-memory split storage mode is used in this paper. The input data and weights of each layer are placed in one memory, and the temporary calculation results of each layer are placed in another memory to make full use of the storage resources of main memory. It is worth mentioning that the output size of the first CONV layer is the largest, so the size of temporary memory only needs to meet the data size of this layer, and the subsequent calculation results can overwrite the previous calculation cache. In addition, the weights are stored using ping-pong operations to improve the efficiency of data transfer. In this design, all data use the same data structure, and there is no bandwidth wastage due to different bandwidth requirements.

This paper adopts the same padding method, and storing the zero-padding input data in the memory will lead to a waste of storage resources. Therefore, a certain number of idle clock cycles need to be reserved to realize the zero-padding operation before and after the input data.

Fig. 6. Data representation format for 16-bit floating-point numbers.

When the neural network performs inference, there is a huge data operation process inside the model, so the size of the data bit width and the calculation accuracy are issues that must be paid attention to. A reasonable bit width setting not only optimizes the resource allocation of the hardware system, but also meets the precision requirements of the algorithm in the hardware circuit. In GPUs and CPUs, the IEEE-754 data format is commonly used for data types. Its standard was originally developed by William Kahan, a professor of numerical computing and computer science at the University of California, Berkeley, who helped Intel design the 8087 floating-point processing. The IEEE-754 standard was formed on this basis. In IEEE-754, the single-precision floating-point type includes a 1-bit sign (S), a 5-bit biased exponent (E), and a 10-bit trailing significand field (T), representing a variety of data from 10^{38} to 10^{-37}, but with excessive data bit-width in the neural network. The inference process increases the power consumption and area. In addition, data reasoning can also be done using fixed-point methods. The fixed-point number representation has the advantage of being intuitive to operate, with the integer part and the fractional part being represented by truncating a certain length of bit width, respectively. Because of this, the fixed-point arithmetic unit is designed with a small area, which is attractive for processors with low-power design requirements. However, the shortcomings of fixed-point numbers are also obvious. Its representation range is small, and the choice of scale factors in different number operations has to be considered to prevent overflow. Compared to floating-point numbers, fixed-point number operations introduce greater errors. Moreover, processors with floating-point computing capabilities are feasible for processing data from near-sensors using high dynamic range, especially for biosignal processing [28]. In summary, the numerical representation of half-precision floating-point numbers, float16, is used throughout this design. This format is a subset of the IEEE-745 standard and is much simpler to handle, in addition to being able to accomplish the required representation of real numbers. The floating-point format, as shown in Fig. 6, uses a ternary array $\{S,E,T\}$ to represent the number N.

Where S represents the sign bit, and the bit width is 1 bit. When $N < 0$, $S = 1$, and when $N > 0$, S = 1. E represents the exponent bit with a bit width of 5 bits, which can represent the range of 0–31, but the actual exponent range is −15–16. T represents the trailing dominant field of N with a bit width of 10 bits, and the decimal point is to the left of the highest bit of the trailing dominant field, whose high bit has an implicit bit. This implicit bit is fixed to 1 bit, and we call the combination of the trailing explicit field and the implicit bit the normalized mantissa F. Therefore, the calculation formula for the value N (float16) in the IEEE-754 format is,

$$F = \{1, T\} \tag{3}$$

$$N = (-1)^s \times \left(\frac{F}{2^{10}}\right) \times 2^{E-15} \tag{4}$$

3.2.2. Calculation subsystem

3.2.2.1. Floating point multiplier and adder. Data reasoning in CNN is inseparable from floating-point multiplication and addition, so ensuring that floating-point multiplication and addition operations are correct is the most essential task. Calculating floating-point multiplication is more convenient. Let two 16-bit floating-point numbers be N_1 and N_2,

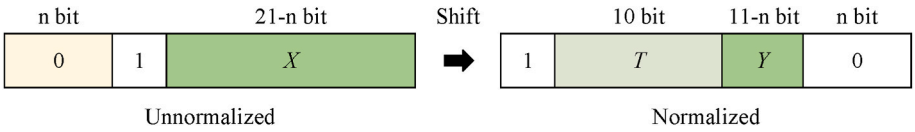


Fig. 7. Normalization method.

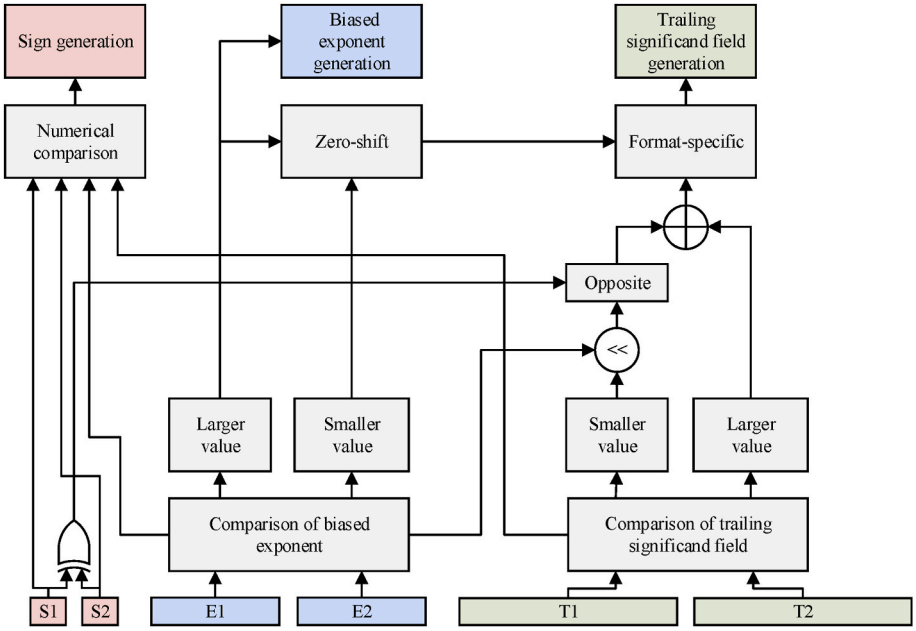


Fig. 8. Floating-point adder hardware operation flow.

respectively. When they are multiplied, there are

$$N_3 = S_1 \otimes S_2 \times 2^{(E_1-15)+(E_2-15)} \times \left(\frac{F_1 \times F_2}{2^{20}}\right) \tag{5}$$

The sign bit of the result of equation (5) N_3 is the XOR of two floating-point sign bits, the exponent bits are added, and the normalized trailing significand field bits are multiplied. The result of multiplying two trailing significand fields of width 11 is 22 bits wide, and we need to normalize this result. Because the trailing number has to be shifted after the first significant digit, the trailing number must be shifted. As shown in Fig. 7, one is the highest significant digit of the two multiplication results, X is the remaining data after shifting the highest significant digit

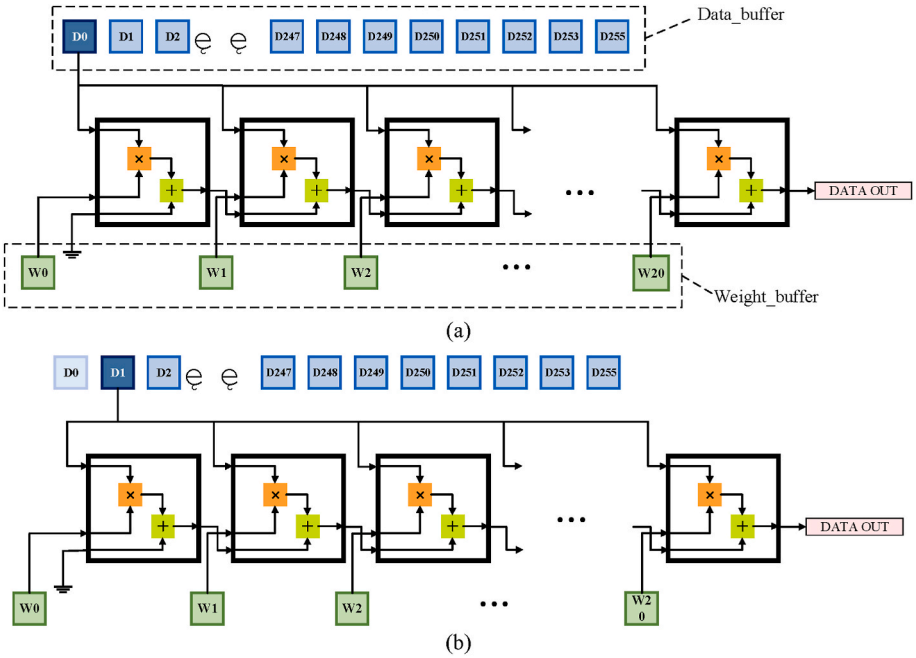


Fig. 9. Schematic diagram of the convolution acceleration hardware calculation process (a) current moment (b) next moment.

to the first of the 22 bits, and the trailing 10 bits of data are the last multiplication result *T*. For each digit shifted to the left, the exponent is subtracted by 1, and for each digit shifted to the right, the exponent is added by 1.

The floating-point adder is more complex than the floating-point multiplier because the IEEE-754 floating-point expression format is based on scientific notation. Therefore, if two numbers are of different orders, they cannot be added directly, but need to be operated on to achieve exponential equality before they can be added.

When adding floating-point numbers, the exponents are first compared and the tails are added directly if the exponents are the same; otherwise, the larger exponent is used as the standard and the smaller exponent is added to the order. The process of factoring is to increase the exponent and shift the trailing significand field at the same time. For each increase of 1 in the exponent, the trailing number is divided by 2, which is expressed as a binary right shift. The exponent and trailing significand field of the results are obtained by adding up the two-digit trailing significand field after the step is completed, and then normalizing and rounding. The sign bit is also obtained for about the size of the two values. When the sign bits of two addends are the same, the sign bit of either addend is taken as the sign bit of the result. When the sign bits of the two addends are different, we need to compare the larger value of the two addends, that is, we compare the two numbers *E* and *T*, and the comparison priority of *E* is higher than that of *T*. The operating mode of floating-point addition is shown in Fig. 8.

3.2.2.2. Convolution. The convolution module is arguably one of the most computationally intensive modules in the CNN, so the design of his acceleration is crucial. The convolution operation can be regarded as a multiply-accumulate (MAC) process. The simplest construction requires only a multiplier, an adder and a register to hold the accumulation of previously calculated values. However this kind of hardware is inefficient and the calculation results also require many clock cycles. In addition, parameter sharing is a major feature of CNNs. Each input data will participate in the operation with multiple weights. If the data needs to be re-entered every time the convolution kernel window slides, a jump in the register will occur. Changing too often can result in very high-power consumption. If the input data can be fully utilized in one clock cycle instead of repeatedly calling these data, it will theoretically reduce the speed and power consumption of the operation.

Here we choose to accelerate the convolutional operation by means of broadcasting the input data, as shown in Fig. 9. In the above designed ECG classification CNN, the convolution kernel of the model is 21×1. Before entering the input data, the 21 accumulation registers are biased. When the input data is input to the multiplying and accumulating module, the number of weights in the multiplying and accumulating module is kept fixed and the input data is broadcasted to these 21 wt, with the calculated value being stored in the computational cache space. Also, when the next clock cycle comes, the second input data is broadcast to the 21 wt, and the value obtained this time is added to the value of the previous multiply-accumulate unit of the previous clock cycle, and after 21 clock cycles, the result of the first convolution is obtained. After each clock cycle, a convolution result is output and a counter is used to count the number of data results. After each output of 256 data, a batch of weights is replaced four times to obtain four cycles. The result is obtained for the convolution of the channels.

Algorithm 1. convolution accelerator.

Algorithm 1 Convolution accelerator

```
1   initialization;
2   dout_reg = 0;
3   dout = 0;
4   output(dout);
5    For i ← 1 to DATA_SIZE do
6        For j ← 1 to WEIGHT_SIZE do
7            dout_reg[j] = DATA_IN * WEIGHT[j];
8        End For
9        dout = dout + dout_reg;
10       output(dout);
11       For k ←WEIGHT_SIZE to 1 do
12           If k = 0 then
13               dout[0] = 0;
14           Else
15               dout[k] = dout[k-1];
16           End If
17       End For
```

The second layer of convolution operation is slightly different from the first layer, as the input data need to be input in parallel to the results of the output of the previous layer, that is, the single input data becomes 4 elements, and the corresponding weights also become 4 elements. Therefore, to obtain the second layer of convolution for an element

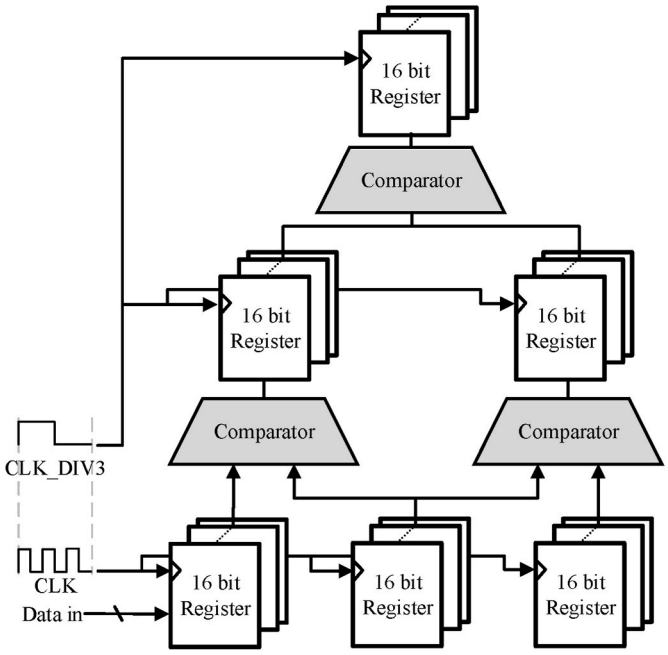


Fig. 10. Pooling architecture diagram.

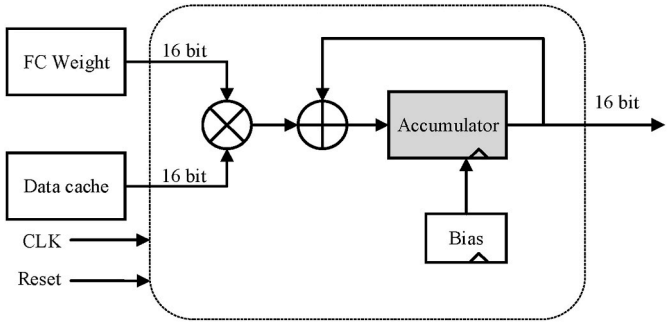


Fig. 11. Fully connected architecture diagram.

requiring 21 × 4 input data and 21 × 4 wt, their corresponding positions are subject to multiplicative accumulation operations, and the second layer of an element is calculated as shown in formula (6), which can be easily seen that the amount of data operations is much larger.

$$R_{1,1} = \mathrm{SUM}\left(\begin{pmatrix} D_{1,,1} & D_{1,2} & \cdots & D_{1,4} \\ D_{2,1} & D_{2,2} & \cdots & D_{2,4} \\ \vdots & \vdots & \ddots & \vdots \\ D_{21,,1} & D_{21,2} & \cdots & D_{21,4} \end{pmatrix} \begin{pmatrix} W_{1,1,1} & W_{2,1,1} & \cdots & W_{21,1,1} \\ W_{1,1,2} & W_{2,1,2} & \cdots & W_{21,1,2} \\ \vdots & \vdots & \ddots & \vdots \\ W_{1,1,4} & W_{2,1,4} & \cdots & W_{21,1,4} \end{pmatrix} \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 1 \end{pmatrix}_{21\times 21} \right) \tag{6}$$

where D, W and R are the input data, weights and output data, respectively.

Considering the above, we rearranged the weights by splitting the 256 × 4 input data into four 256 × 1 data sequences, extracting the weights related to a single input sequence in the four channels, and calling the 4th A multiply-accumulate module used by one layer to compute the convolution of the 4 channels with a single data sequence in parallel. To compute the second layer of convolution in this mode, each data sequence only needs to be passed once. The calculation results of sub-channels will be saved in the FIFO. When the next set of sub-channel calculation results are valid, the data saved in the FIFO will be read out one by one and accumulated with the newly generated results, and the accumulated values will be fed back into the FIFO. Therefore, the computation of sub-channels is performed in parallel and no additional storage space is wasted.

3.2.2.3. Rectified linear unit. We set the Rectified Linear Unit (ReLU) after the convolution module, which is used to activate a certain part of neurons in the neural network, with the activated part retaining its original value and the inactive part assigned a value of 0. The activation information continues to be involved in the inference of subsequent parts of the neural network. The significance of the activation setting activation function is to introduce nonlinear operations into the highly linear neural network, which gives the neural network a strong fitting ability. The dimensions of the input and output do not change after being processed by the activation function.

$$y = \max\{0, x\} \tag{7}$$

In the hardware implementation, it is only necessary to determine whether the highest sign bit S of N is 1. If it is 1, then N takes 0 after being activated, otherwise the original value is retained.

3.2.2.4. Pooling. The various computations that reduce the dimension of the feature map are called pooling, which is a downsampling process that achieves higher accuracy than average pooling and saves more texture information by reducing the offset error of the evaluation values from the CONV layers [29,30]. The pooling operation plays a role of de-redundancy in the entire network. The pooling operation is performed by dividing the input data into multiple partial windows. In this paper, the maximum pooling method is used, that is, the largest data in the local window is taken as the sampling value.

Here we use a pooling window of size 3 × 1 for the pooling operation. The hardware design of the pooling module is implemented via a two-layer comparator path, three comparators, and a three-register comparator tree pipeline, as shown in Fig. 10. During the pooling operation, the SRAM clock is reduced by one-third making the two comparators in the first layer ready to start comparing in parallel when the rising edge of the divider clock arrives and output the maximum value in the pooling window is output after two clock cycles, after which output from the pooling window arrives at each rising edge of the divider clock.

3.2.2.5. Fully connected. A FC layer is similar to a multilayer perceptron (MLP) in that there are two layers of neurons in the FC layer, the first being the input layer and the second being the output layer. The two layers of neurons are connected and all output results are a weighted sum of the inputs. Therefore, the fully connected module [31] requires a large amount of computation and storage. In this way, FC layers can integrate information with similar distinctions in the convolutional and pooling layers.

The FC layer is implemented similarly to matrix-vector multiplication. Fig. 11 shows the fully connected neuron architecture designed in this paper, where at each clock cycle, all neurons have the same input data and the corresponding weights for each channel are read from the weight memory. An accumulator requires as many clock cycles as the number of input data to calculate the output result for that channel.

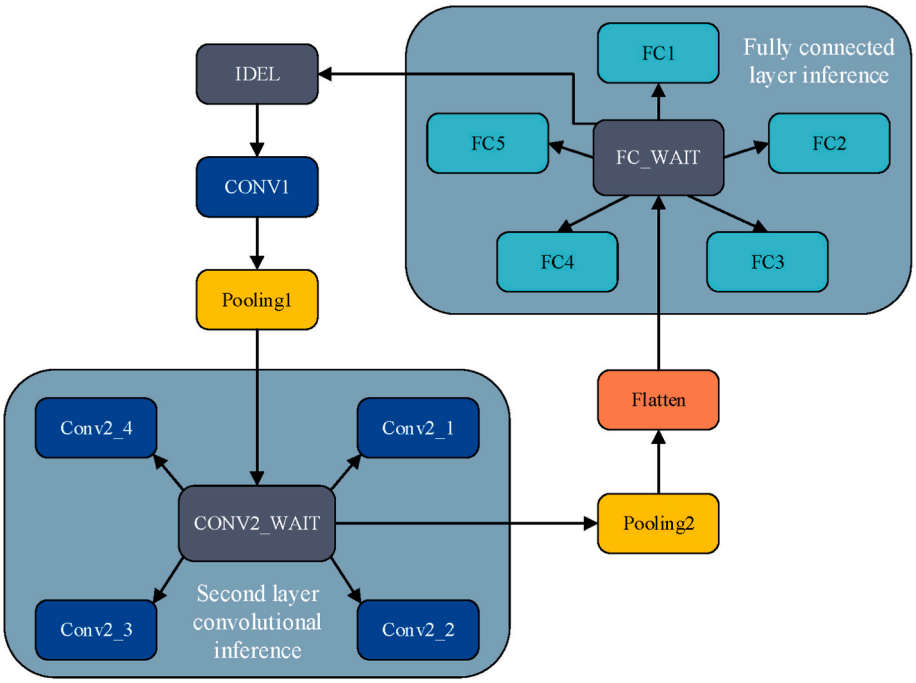


Fig. 12. Convolutional neural network hardware IP control unit state transfer diagram.

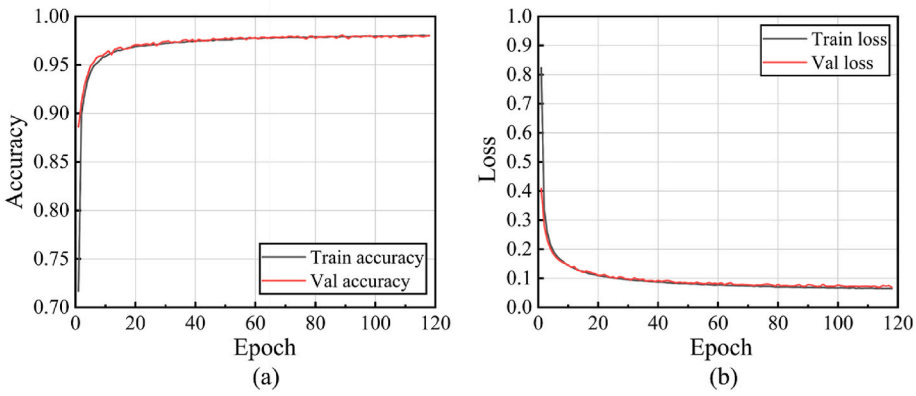


Fig. 13. Accuracy and loss plots of the abnormal heart rhythm detection model. (a)Accuracy curves; (b) Loss curves.

3.2.2.6. Softmax. The Softmax activation function is often used in the output layer of neural networks to handle multi-classification tasks. The data can be transformed into a probability distribution from 0 to 1 with a sum of 1 by the Softmax function. The larger the difference, the larger the distance. Related studies are using Taylor expansion [32] or inverse Softmax method [33] to implement function operations. The expression of the Softmax function is given in the following formula. As can be seen from the formula, exponential operations and division are required to implement the Softmax function.

$$S_i = \frac{e^{a_i}}{\sum_{j=0}^{R} e^{A_j}} \tag{8}$$

In this study, the probability of a classification result can be obtained without implementing the Softmax function in hardware. The label corresponding to the maximum value of the output data of the FC layer is the result we want, so the classification can be achieved directly via the comparator tree mentioned in pooling, and the tedious operations of taking the exponent and power-dividing module can be skipped.

3.2.3. Control subsystem

The operation of the entire control system requires a central control unit and multiple controllers based on finite-state machines. Start and end signals need to be sent to the control unit at the beginning and end of each layer of operation, with the state machine determining the next state and outputting valid and enable signals. In addition, the controller needs to generate addresses for reads and stores to facilitate weight loads and data storage in the SRAM.

The control unit state transfer diagram is shown in Fig. 12. The hardware IP start data inference occurs after the weight data and ECG signals are loaded, and after the IDEL state receives the start signal, the classification circuit enters the operation of convolution and then pooling. When in the state of the second layer of convolutional inference, the convolutional computation module needs to perform the convolutional operations in each sub-channel in turn, so the CONV_WAIT state determination branch is used here, as well as a counter to determine which sub-channel convolutional operation is performed. The pooling and Flatten operations are performed after the second layer of convolutional computation is completed. The control of the state

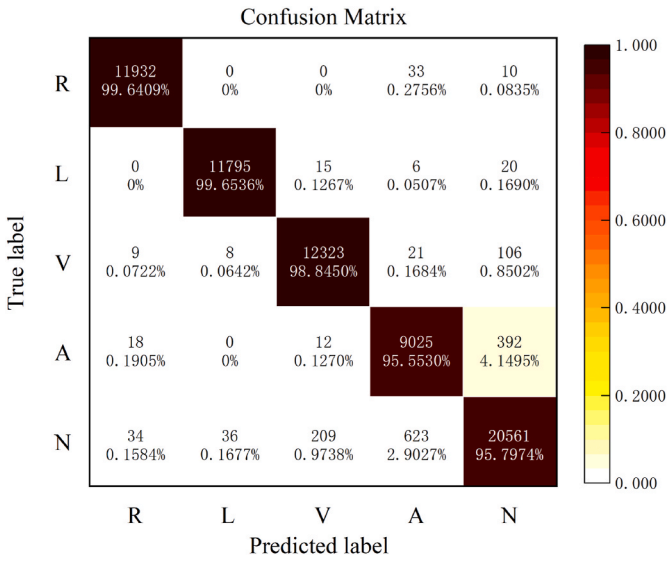


Fig. 14. Normalized confusion matrix for hardware classification results.

machine for the fully connected layer is also implemented by the state judgment branch FC_WAIT and the counter. Return to IDEL after all computation tasks are completed and wait for the next ECG signal to finish loading.

4. Results and discussion

In this paper, the proposed 1D-CNN is implemented in Anaconda3, Tensorflow2.1, and python3.7 (CPU version) environments for modeling. In addition, the EDA tools used for the design of the entire hardware IP core are the simulation tool VCS, the logic synthesis tool Design Compiler, the layout and routing tool IC Compiler, Virtuoso Layout Editor IC5141. The dataset used is from MIT-BIH, from which a total of 200,000 ECG beats were extracted and expanded for training and validation. We randomly split 70% of the data in each class for training in order to obtain good training parameters. The remaining data were assigned to the validation and test sets to examine the classification effect of the abnormal heart rhythm detection CNN model. The Adam optimizer is used to speed up the training process during model backpropagation.

From the perspective of software model training and validation, the proposed 1D CNN model was trained on a laptop computer equipped with an Intel Core i5-8300 CPU 2.3 GHz processor. The model accuracy

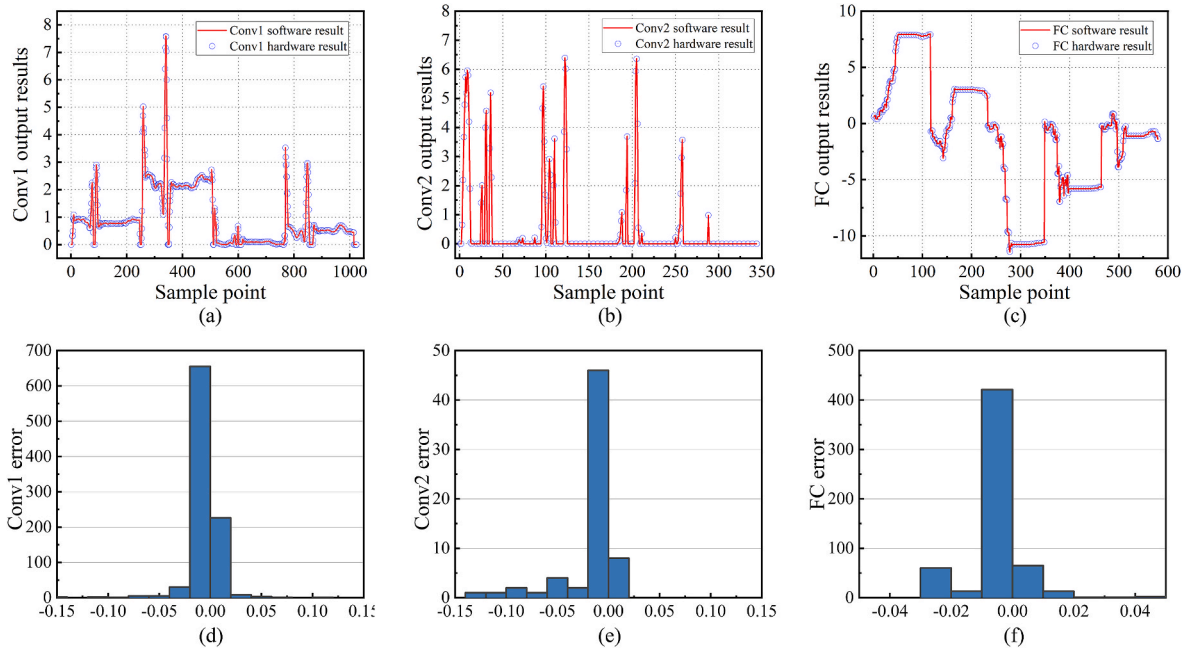


Fig. 15. Software and hardware calculation process error. (a–c) Software and hardware calculation results, (d–f) Software and hardware calculation error.

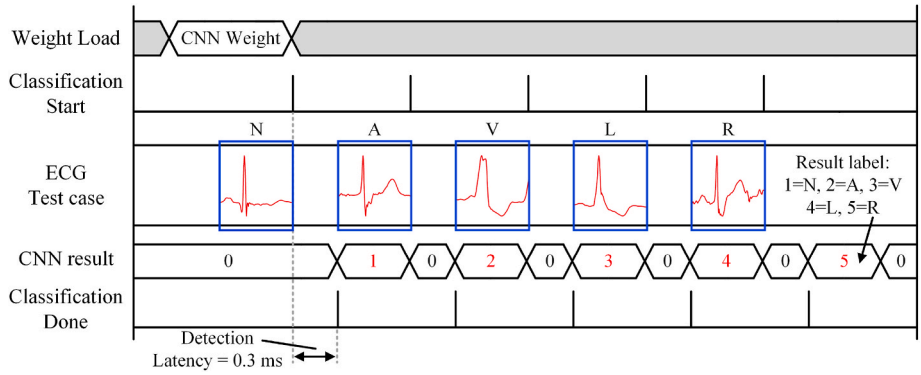
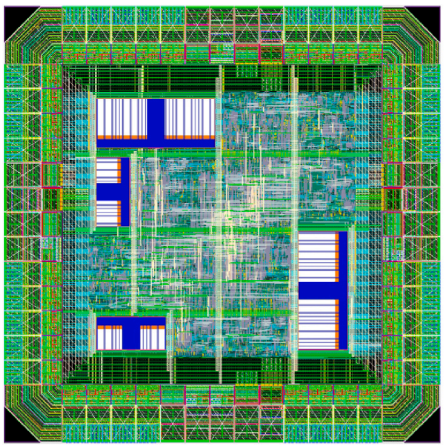


Fig. 16. Results of ECG waveform measurements during cardiac arrhythmia.



ECG classification core feature	
Technology	TSMC 65nm 1P9M
Area	0.191 mm^2
Memory Size	5.12 kB
Equivalent Gates (NAND)	36.9 kgates
Voltage	1 V
Frequency	20 MHz
Power	1.1419 mW
Accuracy	97.69%
Sensitivity	98.67%

Fig. 17. ECG classification chip core layout and performance summary.

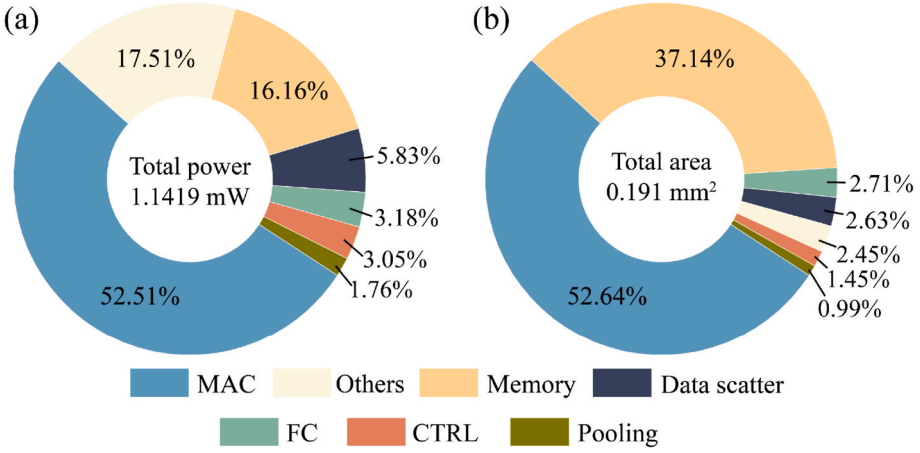


Fig. 18. (a) Power consumption and (b) area consumption breakdown of the ECG classification chip core.

and loss curves for the training and validation sets are shown in Fig. 13. The model accuracy curves grow rapidly at the beginning of the iterations. After 40 iterations, the accuracy curve leveled off and showed good convergence. The proposed heart rhythm abnormality detection model achieves 98.06% and 98.02% accuracy on the training and validation sets, respectively, after 120 iterations.

Abnormal heart rhythm identification is a classification challenge. Here we use the most commonly used indicators to evaluate the performance of the classifier designed in this paper: confusion matrix, accuracy (Acc), sensitivity (Sen), specificity (Spe), and positive predictive ratio (Ppr). Accuracy represents the proportion of events correctly classified among all events, sensitivity indicates the rate of successfully classifying positive events, specificity is the rate of successfully classifying all negative events as negative, and positive classification rate indicates the rate of all detected abnormal heart rhythm events being correctly classified. They are obtained from formulas consisting of the True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN), corresponding to the following mathematical descriptions.

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

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

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

Table 2
ECG classification hardware design performance comparison.

	[34]	[35]	[36]	[37]	This work
Technology/ FPGA	40 nm	65 nm	22 nm	Virtex7	65 nm
Frequency	10 kHz	10 kHz	100 MHz	324 kHz	20 MHz
VDD	1	1	0.8	1	1
Power	14.14 mW	2.78 μW	0.783 μW	12.33 mW	1.1419 mW
Area	0.135 mm^2	0.112 mm^2	1.27 mm^2	NA	0.191 mm^2
Model Utilized	SVM	Navie Bayes	1D-CNN	2D-CNN	1D-CNN
Accuracy	88.06%	86%	78.01%	95.2%	97.69%
Memory (KByte)	12	NA	NA	41.3	5.12
Number of Classes	3	2	2	2	5

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

The classification results of classification problems *TP*, *TN*, *FP* and *FN* are defined as follows. *TP* denotes the positive sample in the actual and predicted samples, *TN* denotes that the actual and predicted samples are both negative samples, *FP* denotes the actual negative samples but predicted positive samples, and *FN* denotes the actual positive sample but predicted negative sample. In the arrhythmia classification problem,

the value of the accuracy rate reflects the performance of the model in classifying ECG signals, while the sensitivity, specificity and prediction rate are the effects of the model on the recognition of various cardiac signals.

The confusion matrix for cardiac arrhythmia classification is shown in Fig. 14, with the following accuracy rates for all categories. Normal heartbeat 95.80%, premature atrial beat 95.55%, premature ventricular beat 98.85%, left bundle branch block fluctuation 99.65%, and left bundle branch block fluctuation 99.64%.

The error of the data in the network comes from the multiplication and addition operation of the value, and in this model, only the CONV layer and the FC layer have the multiplication and addition operation. A randomly selected ECG signal is entered into the model, and the results of comparing the hardware and software output results of the CONV and FC layers are shown in Fig. 15, and more than 85% of the values are within a numerical gap of 0.02. Since the neural network has the characteristics of good robustness, we believe that errors of this scale are acceptable in hardware model inference.

The timing diagram, shown in Fig. 16, depicts the key signals and the measurement results of the neural network hardware detection IP core for arrhythmia during the processing of ECG signals. The Weight Load port is used to input the weight information required by the arrhythmia detection neural network, after which we feed the five classes of ECG signals into the classification circuit and initiate a pulse signal by Classification start after each heart beat input is completed. The arrhythmia convolutional neural network detection circuit outputs classification results from 1 to 5 after 0.3 ms of data inference, and they correspond to five types of heart beats, N, A, V, L, and R, respectively, while a pulse signal indicating the end is sent by Classification Done.

Fig. 17 shows the layout details of the ECG classification chip core. The hardware IP core is implemented on TSMC 65-nm 1P9M CMOS process. 36.9 k equivalent logic gates (minimum sized NAND) and a memory size of 5.12 kB are integrated.

The breakdown of power consumption and area for the entire rhythm classification core is shown in Fig. 18. The power pie chart is estimated at a clock frequency of 20 MHz, with a single voltage of 1V (from a single 1-V supply) and a total power consumption of 1.1419 mW. The MAC unit accounts for 52.51% of the power consumption, the storage unit accounts for 16.16%, and other parts containing, for example, metal alignment, buffering, and unused locations also account for 17.5%. The data routing module, the fully connected computing module, the control unit(CTRL) and the pooling module account for 5.83%, 3.18%, 3.05% and 1.76% of the area, respectively. The core area of the IP core (excluding I/O pads) is 0.191 mm^2, with the MAC unit and memory unit occupying 52.64% and 37.14% of the IP core area.

Table 2 summarizes the performance and hardware resource parameters of the proposed CNN accelerator [34]. uses SVM to classify three heartbeats with a clock frequency of 10k. The hardware implementation has a small area, requires less storage resources but low accuracy, and also consumes 14.14 mW of power, which is better than that reported in Refs. [34,35], and [36] using low-power technology to reduce the power consumption to 2.78 W and 0.783 W; however, the accuracy and number of classifications of the classification model in this paper are significantly higher. Compared with [37], this paper achieves lower power consumption, smaller area and less storage resources under 65 nm process library due to the lightweight model structure design. Therefore, the hardware architecture is suitable for portable and wearable devices.

5. Conclusion

This paper presents a lightweight CNN model and an efficient hardware architecture designed for cardiac arrhythmia classification of wearable medical devices. The model consists of 2 CONV layers, 2 pooling layers, 1 FC layer, and finally the classification label with the highest number of outputs from the FC layer is obtained. The data in hardware network inference is expressed in a 16-bit floating-point format in IEEE-754, which completes the hardware acceleration of the convolutional, pooling, and fully connected layers. The front-end and back-end design of the hardware implementation of abnormal heart rhythm detection is based on TSMC 65 nm process. The IP core has an area of 0.191 mm^2, an operating voltage of 1 V, an operating frequency of 20 MHz, a power consumption of 1.1419 mW, and a classification accuracy of 97.69%. The work presented in this paper demonstrates the feasibility of lightweight 1D-CNN methods in the classification of heart rhythm abnormalities, and we hope that the proposed heart rate abnormality detection model will help medical professionals reduce their workload when faced with a large number of ECG signals for analysis.

Declaration of competing interest

None Declared.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (62174121, 61871244, 62134002), The Fundamental Research Funds for the Provincial Universities of Zhejiang (SJLY2020015), The S&T Plan of Ningbo Science and Technology Department (202002N3134), the K. C. Wong Magna Fund in Ningbo University and the Science, The General Research Project for Education Department of Zhejiang Province (Y202249923), The Fresh Talent Programme for Science and Technology Department of Zhejiang Province (2022R405B082).

References

[1] J. Fayn, P. Rubel, Toward a personal health society in cardiology, IEEE Trans. Inf. Technol. Biomed. 14 (2) (2009) 401–409, https://doi.org/10.1109/TITB.2009.2037616.
[2] M. Shannon, R. Quinn, J. Randal, et al., Participation in cardiac rehabilitation, readmissions, and death after acute myocardial infarction, Am. J. Med. 127 (6) (2014) 538–546, https://doi.org/10.1016/j.amjmed.2014.02.008.
[3] B.F. Cuneo, S.E. Sonesson, S. Levasseur, et al., Home monitoring for fetal heart rhythm during anti-ro pregnancies, J. Am. Coll. Cardiol. 72 (16) (2018) 1940–1951, https://doi.org/10.1016/j.jacc.2018.07.076.
[4] S. Ramasamy, A. Balan, Wearable sensors for ECG measurement: a review, Sens. Rev. 38 (4) (2018) 412–419, https://doi.org/10.1108/SR-06-2017-0110.
[5] Y. Chen, X. Yang, Z. Wei, et al., Generative adversarial networks in medical image augmentation: a review, Comput. Biol. Med. 144 (2022) (2022), 105382, https://doi.org/10.1016/j.compbiomed.2022.105382.
[6] Y. Chen, H. Gan, H. Chen, et al., Accurate iris segmentation and recognition using an end-to-end unified framework based on MADNet and DSANet, Neurocomputing 517 (2023) (2023) 264–278, https://doi.org/10.1016/j.neucom.2022.10.064.
[7] M. Yu, M. Han, X. Li, et al., Adaptive soft erasure with edge self-attention for weakly supervised semantic segmentation: thyroid ultrasound image case study, Comput. Biol. Med. 144 (2022) (2022), 105347, https://doi.org/10.1016/j.compbiomed.2022.105347.
[8] Q. Guan, Y. Chen, Z. Wei, et al., Medical image augmentation for lesion detection using a texture-constrained multichannel progressive GAN, Comput. Biol. Med. 145 (2022) (2022), 105444, https://doi.org/10.1016/j.compbiomed.2022.105444.
[9] Y. Liu, A. Heidari, Z. Cai, et al., Simulated annealing-based dynamic step shuffled frog leaping algorithm: optimal performance design and feature selection, Neurocomputing 503 (2022) (2022) 325–362, https://doi.org/10.1016/j.neucom.2022.06.075.
[10] Y. Wen, Y. Zhang, L. Wen, et al., A 65nm/0.448 mW EEG processor with parallel architecture SVM and lifting wavelet transform for high-performance and low-power epilepsy detection, Comput. Biol. Med. 144 (2022) 105366–105390, https://doi.org/10.1016/j.compbiomed.2022.105366.
[11] X. Yang, Q. Yu, L. He, et al., The one-against-all partition based binary tree support vector machine algorithms for multi-class classification, Neurocomputing 113 (3) (2013) 1–7, https://doi.org/10.1016/j.neucom.2012.12.048.
[12] L. Ma, L. Liang, A regularization method to improve adversarial robustness of neural networks for ECG signal classification, Comput. Biol. Med. 144 (2022) (2022), 105345, https://doi.org/10.1016/j.compbiomed.2022.105345.
[13] R. Hu, J. Chen, L. Zhou, A transformer-based deep neural network for arrhythmia detection using continuous ECG signals, Comput. Biol. Med. 144 (2022) (2022), 105325, https://doi.org/10.1016/j.compbiomed.2022.105325.
[14] T.P. Exarchos, M.G. Tsipouras, C.P. Exarchos, et al., A methodology for the automated creation of fuzzy expert systems for ischaemic and arrhythmic beat classification based on a set of rules obtained by a decision tree, Artif. Intell. Med. 40 (3) (2007) 187–200, https://doi.org/10.1016/j.artmed.2007.04.001.

[15] G. Carrault, M.O. Cordier, R. Quiniou, et al., Temporal abstraction and inductive logic programming for arrhythmia recognition from electrocardiograms, Artif. Intell. Med. 28 (3) (2003) 231–263, https://doi.org/10.1016/s0933-3657(03)00066-6.
[16] C. Shen, W. Kao, Y. Yang, et al., Detection of cardiac arrhythmia in electrocardiograms using adaptive feature extraction and modified support vector machines, Expert Syst. Appl. 39 (9) (2012) 7845–7852, https://doi.org/10.1016/j.eswa.2012.01.093.
[17] R.J. Martis, C. Chakraborty, A.K. Ray, A two-stage mechanism for registration and classification of ECG using Gaussian mixture model, Pattern Recogn. 42 (11) (2009) 2979–2988, https://doi.org/10.1016/j.patcog.2009.02.008.
[18] T. Liu, Y. Si, D. Wen, et al., Dictionary learning for VQ feature extraction in ECG beats classification, Expert Syst. Appl. 53 (2016) (2016) 129–137, https://doi.org/10.1016/j.eswa.2016.01.031.
[19] S. Kiranyaz, T. Ince, M. Gabbouj, Real-time patient-specific ECG classification by 1-D convolutional neural networks, IEEE (Inst. Electr. Electron. Eng.) Trans. Biomed. Eng. 63 (3) (2016) 664–675, https://doi.org/10.1109/tbme.2015.2468589.
[20] M. Apu, F. Akter, M. Lubna, et al., ECG arrhythmia classification using 1D CNN leveraging the resampling technique and Gaussian mixture model. International Conference on Informatics, Electronics & Vision (ICIEV) and International Conference on Imaging, Vision & Pattern Recognition (icIVPR) (2021) 1–8, https://doi.org/10.1109/ICIEVicIVPR52578.2021.9564201.
[21] L. Wei, D. Liu, J. Lu, et al., A low-cost hardware architecture of convolutional neural network for ECG classification, in: International Symposium on Next Generation Electronics, ISNE), 2021, pp. 1–4, https://doi.org/10.1109/isne48910.2021.9493657.
[22] Y. Ji, S. Zhang, W. Xiao, Electrocardiogram classification based on faster regions with convolutional neural network, Sensors 19 (11) (2019) 2558–2576, https://doi.org/10.3390/s19112558.
[23] Q. Yao, R. Wang, X. Fan, et al., Multi-class arrhythmia detection from 12-lead varied-length ECG using attention-based time-incremental convolutional neural network, Inf. Fusion 53 (2020) 174–182, https://doi.org/10.1016/j.inffus.2019.06.024.
[24] T. Emil, D.S. Debjit, V. Ganesh, et al., Compute solution for tesla's full self-driving computer, IEEE Micro 40 (2) (2020) 25–35, https://doi.org/10.1109/MM.2020.2975764.
[25] R.A. Mostafa, L. Corey, K.E. Jason, et al., Hardware implementation of deep network accelerators towards healthcare and biomedical applications, IEEE Transactions Biomed. Circuits Syst. 14 (6) (2020) 1138–1159, https://doi.org/10.1109/TBCAS.2020.3036081.
[26] A.L. Goldberger, L.A. Amaral, L. Glass, et al., Physiotoolkit and physionet: components of a new research resource for complex physiologic signals, Circulation 101 (23) (2000) e215–e220, https://doi.org/10.1161/01.cir.101.23.e215.
[27] H. Wang, H. Shi, X. Chen, et al., An improved convolutional neural network based approach for automated heartbeat classification, J. Med. Syst. 44 (2) (2020) 1–9, https://doi.org/10.1007/s10916-019-151c1-2.
[28] A. Pullini, D. Rossi, I. Loi, , et al.Wolf, An energy-precision scalable parallel ultra low power SoC for IoT edge processing, J. Solid-State Circuits 54 (7) (2019) 1970–1981, https://doi.org/10.1109/JSSC.2019.2912307.
[29] V. Sze, Y. Chen, T. Yang, et al., Efficient processing of deep neural networks: a tutorial and survey, Proc. IEEE 105 (12) (2017) 2295–2329, https://doi.org/10.1109/JPROC.2017.2761740.
[30] K. Khalil, O. Eldash, A. Kumar, et al., Designing novel AAD pooling in hardware for a convolutional neural network accelerator, IEEE Trans. Very Large Scale Integr. Syst. 30 (3) (2022) 303–314, https://doi.org/10.1109/tvlsi.2021.3139904.
[31] A.M. Abdelsalam, F. Boulet, G. Demers, et al., An efficient FPGA-based overlay inference architecture for fully connected DNNs, in: International Conference on ReConFigurable Computing and FPGAs (ReConFig), 2018, pp. 1–6, https://doi.org/10.1109/reconfig.2018.8641735.
[32] Y. Cao, J. Jia, D. Wu, et al., Cordic-based softmax acceleration method of convolution neural network on FPGA, in: International Conference on Artificial Intelligence and Information Systems, ICAIIS), 2020, pp. 66–70, https://doi.org/10.1109/icaiis49377.2020.9194894.
[33] Y. Gao, W. Liu, F. Lombardi, Design and implementation of an approximate softmax layer for deep neural networks, in: IEEE International Symposium on Circuits and Systems (ISCAS), 2020, pp. 1–5, https://doi.org/10.1109/ISCAS45731.2020.9180870.
[34] Y. Xu, Z. Chen, F. Li, J. Meng, A granular resampling method and adaptive speculative mechanism-based energy-efficient architecture for multiclass heartbeat classification, IEEE Trans. Comput. Aided Des. Integrated Circ. Syst. 38 (11) (2018) 2172–2176, https://doi.org/10.1109/tcad.2018.2871819.
[35] N. Bayasi, T. Tekeste, H. Saleh, et al., Low-power ECG-based processor for predicting ventricular arrhythmia, IEEE Trans. Very Large Scale Integr. Syst. 24 (5) (2015) 1962–1974, https://doi.org/10.1109/tvlsi.2015.2475119.
[36] J. Loh, J. Wen, T. Gemmeke, Low-cost DNN hardware accelerator for wearable, high-quality cardiac arrythmia detection, Application-specific Syst. Architectures Processors (ASAP) (2020) 213–216, https://doi.org/10.1109/asap49362.2020.00042.
[37] Y. Ting, F. Lo, P. Tsai, Implementation for fetal ECG detection from multi-channel abdominal recordings with 2D convolutional neural network, J. Signal Process. Syst. 93 (9) (2021) 1101–1113, https://doi.org/10.1007/s11265-021-01676-w.