

A Novel Defibrillator-Specific Coprocessor **Capable of Running Entropy and CNN Integration Algorithms**

Xuelong Wang¹, Peng Xia¹, Changjiang Zhou², Zhenyi Huang², Hao Zhao², Haipo Cui¹, Shiju Yan^{1*}

¹School of Health Science and Engineering, University of Shanghai for Science and Technology, Shanghai, China ²Department of Ultrasound, Jinan City People's Hospital (People's Hospital Affiliated to Shandong First Medical University), Jinan, China

Email: *yanshiju@usst.edu.cn

How to cite this paper: Wang, X.L., Xia, P., Zhou, C.J., Huang, Z.Y., Zhao, H., Cui, H.P. and Yan, S.J. (2024) A Novel Defibrillator-Specific Coprocessor Capable of Running Entropy and CNN Integration Algorithms. Journal of Biosciences and Medicines, 12, 310-322.

https://doi.org/10.4236/jbm.2024.1211026

Received: September 23, 2024 Accepted: November 15, 2024 Published: November 18, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

http://creativecommons.org/licenses/by/4.0/ **Open Access**



Abstract

It is difficult for the existing Automated External Defibrillator (AED) onboard microprocessors to accurately classify electrocardiographic signals (ECGs) mixed with Cardiopulmonary Resuscitation artifacts in real-time. In order to improve recognition speed and accuracy of electrocardiographic signals containing Cardiopulmonary Resuscitation artifacts, a new special coprocessor system-on-chip (SoC) for defibrillators was designed. In this study, a microprocessor was designed based on the RISC-V architecture to achieve hardware acceleration for ECGs classification; Besides, an Approximate Entropy (ApEn) and Convolutional neural networks (CNNs) integrated algorithm capable of running on it was designed. The algorithm differs from traditional electrocardiographic (ECG) classification algorithms. It can be used to perform ECG classification while chest compressions are applied. The proposed co-processor can be used to accelerate computation rate of ApEn by 34 times compared with pure software computation. It can also be used to accelerate the speed of CNNs ECG recognition by 33 times. The combined algorithm was used to classify ECGs with CPR artifacts. It achieved a precision of 96%, which was significantly superior to that of simple CNNs. The coprocessor can be used to significantly improve the recognition efficiency and accuracy of ECGs containing CPR artifacts. It is suitable for automatic external defibrillator and other medical devices in which one-dimensional physiological signals.

Keywords

Defibrillator, SoC, Entropy, Vector Multiplication, Coprocessor, RISCV, ECG Classification

1. Introduction

Cardiac arrest may result in severe or even irreversible damage. If not promptly rescued, the patient's life could be put at risk [1]. Out-of-Hospital Heart Arrest (OHCA) is the third-most common cause of death in the industrialized world. OHCA kills about 700,000 people annually in the United States and Europe [2]. High quality Cardiopulmonary Resuscitation (CPR) and defibrillation are crucial for patients in cardiac arrest [3]. Reasonable electrical defibrillation is a crucial element in enhancing rescue rates for patients who have had cardiac arrest [4] [5]. Severe CPR artifacts may result from chest compressions. However, it is difficult for an on-board microprocessor of the current Automated External Defibrillator (AED) product to accurately classify the electrocardiographic signals (ECGs) contaminated by CPR artifacts. Thus, it is difficult to interpret ECGs while chest compressions are applied.

Recently, along with the improvement of computer performance, CPR artifact filtering algorithms have been developed quickly. Thanks to high performance computers, researchers have achieved remarkable accuracy and speed in filtering artifacts. This progress is attributed to deep learning methods, mathematical analysis, and filtering techniques. Deep learning methods include deep convolutional neural networks, K-Nearest Neighbor classification model, and transfer learning, among others [6] [7]. The mathematical analysis methods include analysis of independent components and elimination of coherent lines [8] [9]. The most widely used filters include band pass filter, Kalman filter, adaptive filter and filter combination structure [10]-[12]. However, these algorithms are not suitable for the existing AED airborne microprocessors. Thus, the deployment of a defibrillator rhythm recognition algorithm is essential. It should be capable of filtering out CPR artifacts from the AED onboard microprocessor.

In this project, an integrated algorithm of ApEn and CNNs was proposed to classify the ECGs which are contaminated by CPR artifacts. Moreover, we designed a Defibrillator-Specific co-processor SoC, which can be used to speed up the ECGs classification algorithms. The co-processor meets the need of analyzing ECGs in real time on AEDs. That integrated algorithm was also deployed on the co-processor SoC. This co-processor is designed for the analysis of physiological signals. It is versatile enough for use in various medical devices for which one-dimensional signal processing is necessary, not limited to defibrillators.

2. Materials and Methods

2.1. Experimental Materials

In this project, we used XC7A100T FPGA development board. It features 63,400 lookup tables, six phase-locked loops, and other resources. It is well equipped to satisfy the design requirements of our project. This development board was used to complete the internal logic design and test of the co-processor. Furthermore, STM32F103, STM32G0B1 and CY8C5888 were used to run test cases and record

operating times. Due to their wide usage and variety of capabilities, we used them for comparative testing.

Three lead electrodes and an ECGs collection circuit were used to collect ECGs from a single volunteer separately. These ECGs were used in the grading tests. Electromagnetic interference was introduced by connecting the acquisition circuit to the chest compressor drive circuit, and this electromagnetic interference was used to simulate CPR artifact signals. Without the simulated CPR artifacts, the obtained ECGs QRS complex is intact. As soon as the chest compressor drive was turned on, the ECG showed an apparent artifact disturbance. We reviewed the literature [7] and compared the waveforms. In consultation with an experienced doctor from Shanghai's Sixth People's Hospital, we determined that the disturbance was consistent with CPR artifacts in CPR. Thus, this portion of the data is used to perform tests for classification of ECG signals in the presence of interference was complexient.

2.2. Design of Instruction Set and Instruction Decoding Unit of Coprocessor

The RISC-V architecture has been used widely for developing microprocessors for its open-source nature so far. E203 core meets the design requirements of power consumption and performance. Therefore, we used the E203 core as the central processor CPU of the SOC.

Both power consumption and processing performance are critical for AED. To balance these needs, we employed the NICE interface for communication between the main processor and co-processor. Additionally, We specially designed the coprocessor and employed the tailored cooperative computing unit.

The co-processor includes an instruction decode unit, an execution state machine, an entropy calculation acceleration unit, a CNNs vector calculation unit, and a single cycle fixed-point multiplier. The design block diagram is shown in **Figure 1**.



Figure 1. Overall design block diagram of coprocessor.

The co-processor designed in this study exhibits diverse functionalities with intricate structures. In order to have proper control and scheduling, we control data transfer, data manipulation, and command execution with an execution state machine. The machine switch among 8 different states: Idle, Load Address Operands, Read, Write, Compute, Wait, Write MEM, and Shake Hand.

The instruction decode unit consists of a combined logic circuit and a D trigger. The co-processor supports execution of instructions while prefetching an instruction. The equal comparator determines each component's legitimacy in the instruction. After this judgment, an AND gate retrieves the instruction's prefetch signal. Upon handshake completion. The co-processor transmits the instruction's execution signal using the D trigger. This logic ensures that the co-processor transit to the next instruction seamlessly once the current one concludes.

The co-processor features seven instructions, with three dedicated to the entropy computing accelerator: "custom_1buf_apen", "custom_1th_apen", and "custom_calu_apen". When calculating the program entropy, the "custom_1buf_apen" instruction is called first, writing the initial address of the sequence to the entropy computing accelerator. Then, the entropy computation threshold is set via the "custom_1th_apen" command. Finally, "custom_calu_apen" is performed to retrieve the result of the calculation.

The two instructions that control the vector calculation unit are "custom_1ker_cnn" and "custom_calcu_cnn". When doing convolutional multiplication, the convolution core is loaded into cell using "custom_1ker_cnn" instruction. Then, the convolution calculation is constantly carried out through the "custom_calcu_cnn" instruction.

The CPU core used in this project is a low-power CPU, without single-cycle multiplier and floating point computing unit. But the majority of signal-processing algorithms require a significant number of multiplication operations. In order to prevent multiplication operations from slowing down the overall computation speed of the algorithm and affecting the acceleration effect of coprocessor, two multiplication instructions for fixed-point and floating-point operations are added to the co-processor: "custom_multi", and "custom_multi_f".

3. Design of Accelerated Unit for Tailorable Entropy Computation

3.1. Analysis of Approximate Entropy Calculation Program

Approximate entropy (ApEn) is an algorithm for judging regularity of data. Notably, ApEn enables estimation of a data series' randomness without prior knowledge of the data source [13]. Owing to its simplicity and high applicability, ApEn algorithm has been widely used in various research fields [14]. Its applications span biology and medicine, such as seizure detection [14], quantification of hormone pulsatility [15], depth of analgesia during propofol-remifentanil anesthesia [16], and gene expression data classification [17].

In this paper, a computing library of ApEn algorithm was developed to evaluate

the complexity and to optimize it. The computation library includes five key functions: initialization, data structure reconstruction, vector distance computation in M and M + 1, log mean calculation, and subtraction. Using this library, we evaluated the ApEn algorithm on a 500-point sequence.

We used this library to calculate the approximate entropy of a 500-point sequence on a laptop with an AMD R7-4800U CPU, and we find that the calculation of the vector distances occupies most of the computation time. Vector distance computation takes up most of the computing time due to two nested loops. This leads to an exponential increase in computation with the increase of data points.

The best range of sequence points for calculating approximate entropy is often between 100 and 5000 points. Considering human heartbeat frequency and common device sampling rates, an ideal ECGs sequence length for effective analysis ranges between 200 and 1000 points. Such a range is not ideal for emergency equipment seeking low power consumption. Hence, the design of hardware accelerator should focus on providing optimization for calculating the distance of each vector.

3.2. Design Architecture of Tailorable Entropy Calculation Acceleration Unit

The entropy computing acceleration unit comprises vector distance calculation unit array, parallel calculation controller, vector reconstruction unit, register group, and read-write controller. The function of the vector reconstruction unit is to reconstruct the original data sequence into two-dimensional or three-dimensional vector groups. Based on the control register's calculation number, this unit supplies corresponding data to the vector distance calculation unit. The parallel calculation controller and vector reconstruction unit are used to connect. The register group and the vector distance calculation unit. The architecture of the entropy calculation acceleration unit is shown in **Figure 2**.



Figure 2. The architecture of entropy computing accelerator.

Control of the acceleration unit of entropy computation is done by reading and writing registers. The accelerator inner register group is addressed from 0 to (APEN_DATA_LEN+1). You can change the APEN_DATA_LEN parameter to

trim the entropy computing accelerator.

The address range of the data register group is 0-(APEN_DATA_LEN-1). This register group is used to store the original data sequence to be calculated. The data is not written by CPU in turn, but rather by DMA, in order to guarantee the efficiency of operation.

The control register and the status register share the same address APEN_DATA_LEN+1. The external circuit controls the acceleration unit by writing control signals to this address. The acceleration unit will also write the internal working state to this address for the external circuit to know.

3.3. Design of Vector Distance Calculation Unit

In the approximate entropy algorithm, the definition of vector distance is shown in Formula (1):

$$d\left[X\left[i\right], X\left[j\right]\right] = \max_{k=0-m-1} \left(\left|x\left(i+k\right), x\left(j+k\right)\right|\right)$$
(1)

where *X* is the reconstructed *m*-dimensional vector group, *x* is the original sequence, and *d* is the distance between vectors. The distance is the largest absolute value of the difference between the corresponding elements in the vectors x[i] and x[j].

The vector distance calculation unit consists of a two-level pipeline. The primary stage is a data selector, tasked with selecting data for subsequent output. The secondary pipeline is a calculation circuit responsible for calculating the vector distance of the data provided by the primary pipeline. Sizing of Graphics.

4. Design of Tailorable One-Dimensional CNNs Vector Computing Unit

4.1. Analysis of Convolutional Neural Network Calculation Program

Convolutional Neural Networks (CNNs) are a class of feed-forward neural networks that contain convolutional computation and have deep structure. A CNN includes convolutional layers, pooling layers, and fully connected layers [18]. CNNs have the characteristics of local connections, shared weights, pooling, and the use of many layers [19]. CNNs can automatically extract effective features from data [20]. These benefits of CNNs have led to their widespread application in many different industries. There are successful application cases in computeraided detection of thoraco-abdominal lymph node and classification of interstitial lung disease [21], classification of image and chart types [22], classification of COVID-19 [23], ECG analysis and other fields [24] [25]. Therefore, in this paper, we use CNNs to classify ECG signals and perform targeted optimization.

Inference SDK for CNNs was designed in this project for MCU. The highest processing resources are used by convolutional computing, followed by full connection, and pooling, and the least amount of resources are used for activation functions. According to tests, the accelerator should focus on the calculation of

convolution layer and fully connected layer.

The definition of sum convolution of one-dimensional sequence is shown in Formula (2):

$$y(n) = \sum_{i=0}^{n} x(i) \cdot h(n-i)$$
(2)

In this project, the convolution design is based on Formula (3), which is compatible with the existing CNNs library. The calculation formula of the fully connected layer is shown in Formula (4):

$$y(n) = \sum_{i=0}^{n} x(i) \cdot h(i)$$
(3)

$$y_{1}(i) = x(i) \cdot h_{1}(i)$$

$$y_{2}(i) = x(i) \cdot h_{2}(i)$$

$$\vdots$$
(4)

where represents the network sequence with length n of the previous layer, and represents the parameter sequence with length of n in this layer.

The convolution sequence and the fully connected layer both rely on vector multiplication. To enhance acceleration, a vector multiplication calculation unit was integrated into the coprocessor.

4.2. Architecture Design of One-Dimensional CNN Vector Computing Unit

The E203's ICB bus has a bandwidth restriction, hence parallel computing approach is not effective in improving computational efficiency. Therefore, we abandon the multi-unit parallel computing on the vector computing unit and adopt the long pipeline computing method. The unit is composed of an external interface, a data splitter, a parallel multiplier, a data selector, and an accumulator pipeline.

The structure of the vector computing unit is shown in Figure 3.



Figure 3. The architecture of vector computing unit.

The external interface is responsible for external communication and data exchange. The data splitter is responsible for splitting the serial data. Parallel multiplier, data selector, and accumulator work together to complete CNNs vector calculation.

The unit's external interface extracts data one at a time. The data selector filters valid data and feeds it into the accumulation pipeline after each parallel multiplier operation.

5. Experiments and Results

5.1. ECG Signal Classification Acceleration

A project was written and synthesized on FPGA development board. We also wrote a Convolution Neural Network ECG Test Case and an ApEn Computation Test Case. Test cases come in both software and hardware acceleration versions. The software version and the hardware-accelerated version have exactly the same algorithm, and the only difference is whether hardware acceleration is used or not. The results of the test are given in **Table 1**.

Table 1. Test results of CNNs ECG classification algorithm and ApEn algori
--

Algorithm	Instruction count	Calculate number of cycles						
CNNs ECG classification								
Software-only	2,473,601	3,747,196						
Hardware-accelerated	58,439	114,799						
	ApEn calculation							
Software-only	39,065,597	50,474,100						
Hardware-accelerated	1,042,651	1,486,142						

The CNNs designed in this project is a two-layered network. The raw data are 1000 samples of ECG, sampled at 200 Hz. The first layer of CNNs consists of a convolutional layer, which consists of a convolutional layer with a length of 9. The second level is fully connected, with a parameter sequence of 992. Experimental results indicate that the CNNs test procedure can be accelerated by 33 times with hardware acceleration.

A 250-point sequence is used for the ApEn calculation program. The results show that hardware acceleration can speed up the ApEn test program by about 34 times. At a clock frequency of 16 MHz, the co-processor optimizes the calculation time of 250-point ApEn data from several seconds to several tens of milliseconds.

Additionally, a slightly more intricate routine was designed in this study. Using a common embedded development platform for comparative testing, the computational cycles consumed by each chip model are shown in **Table 2** below.

The results show that the common embedded platform is able to run CNNs and

Hardware platform	Crystal frequency	Number of cycles of CNNs routines	Number of cycles of ApEn routines
E203 co-processor	16 MHz	216,379	1,516,046
E203	16 MHz	3,795,869	51,604,344
STM32F103	72 MHz	1,979,998	44,639,964
STM32G0B1	64 MHz	2,566,400	48,239,961
CY8C5888	24 MHz	1,526,398	120,960,000
STM32F411	25 MHz	2,122,500	65,750,000

Table 2. Test results of different embedded platforms.

ApEn computation, and can also deploy the ECG signal recognition algorithm. But it can not meet the real time demand of synchronous filtering and real time analysis. In this study, the co-processor can speed up the algorithm by accelerating the hardware, and can meet the requirement of real time computation.

5.2. Classification Accuracy of ECG Signal

Three-lead ECG collection line is attached to the body of the volunteer to collect ECGs. Simulated CPR artifacts are created in the same frequency range by means of coupled electromagnetic interference. The electrocardiogram collected is shown in **Figure 4**.



(c) Volunteer ECG signals during chest compressor sustainment



In this study, 54 normal ECG signals of a volunteer were collected independently. Additionally, under conditions simulating CPR artifacts using coupled electromagnetic interference, 50 ECG signals from the same volunteer were collected. These 50 signals are considered as normal ECG data with CPR artifacts. The results are shown in **Table 3**: where N stands for normal, A stands for atrial premature beats, V stands for ventricular premature beats, L stands for left bundle branch block, R stands for right bundle branch block, and F stands for ventricular fibrillation. The accuracy represents the classification accuracy of normal ECG signal n.

The test results indicate that the classification accuracy of normal ECGs under simulated CPR artifacts can be significantly improved by the integrated algorithm. Additionally, the co-processor designed in this study can be used to effectively accelerate the ApEn algorithm and one-dimensional CNNs algorithm. The algorithm can be applied to the analysis of various physiological signals.

 Table 3. Test results of ECGs classification for the algorithm designed in this study.

Algorithm	Ν	А	V	L	R	F	Accuracy	
ECG classification results without interference from analog artifacts								
CNN classification algorithm	53	0	0	0	1	0	98.1%	
Methodology for the design of this study	53	0	0	0	1	0	98.1%	
ECG classification results with simulated artifact interference								
CNN classification algorithm	10	1	0	0	0	40	20%	
Methodology for the design of this study	48	1	0	0	0	2	96%	

N: normal heartbeat A: atrial premature beats; V: ventricular premature beats; L: left bundle branch block; R: right bundle branch block; F: ventricular fibrillation; Accuracy: the classification accuracy of normal ECG signal n.

6. Discussion and Conclusion

6.1. Discussion

An organ's pathology will invariably result in an increase in signal entropy value, which is a crucial performance metric for the human body as an ordered system. Therefore, ApEn is a very important basis in the current physiological signal analysis. In recent years, the neural network algorithm has become one of the important algorithms for analyzing physiological signals. Using neural network algorithm and ApEn algorithm can effectively analyze various physiological lesions.

High-performance ECG signal classification algorithm should be highly efficient and accurate in eliminating CPR artifacts in CPR. However, the integration of such high performance algorithms into the AED microprocessors is a challenge. Common microprocessors like STM32G0B1 and CY8C5888 can not meet the requirement of AEDs' real time signal classification. In order to solve this problem, a co-processor SOC has been developed, which can be used to accelerate the integration algorithm of ApEn and CNNs. To validate the test, we collected and tested ECG signals from volunteer.

In actual testing, we observed that the running speed of common microprocessors for the ApEn algorithm and CNNs algorithm is positively correlated with clock frequency of the chip itself. The running speed of the E203, without hardware acceleration, is slower than that of the STM32F103—which has a higher main frequency—for both algorithms. However, the E203 co-processor, equipped with hardware acceleration, significantly outpaces the other chips.

There are still shortcomings in this design, and future research work includes:

1) Broadening the utility of the calculation unit by enhancing both the approximation entropy calculation library and the associated co-processor circuit.

2) Improving the CNNs inference library to make the SDK more flexible and user-friendly.

6.2. Conclusion

In this study, we presented an AI analysis SoC specifically designed for physiological signal analysis. This SoC incorporates a co-processor to analyze physiological signals via the NICE interface. We developed and optimized an ApEn and CNNs integrated algorithm, leading to a reduction in computational overhead. Subsequent tests validate the design results. Through the integrated software and hardware approach detailed in this study, we achieve a notable acceleration in the speed of ECG classifications. This approach also improves the accuracy of ECG classifications under simulated CPR artifacts. This advancement fulfills the requirements of AEDs for real-time electrocardiogram signal analysis in practical applications.

Funding

Shanghai Municipal Science and Technology Major Project, Grant/Award Number: 2021SHZDZX.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

 Langhelle, A., Nolan, J., Herlitz, J., Castren, M., Wenzel, V., Soreide, E., *et al.* (2005) Recommended Guidelines for Reviewing, Reporting, and Conducting Research on Post-Resuscitation Care: The Utstein Style. *Resuscitation*, **66**, 271-283. <u>https://doi.org/10.1016/j.resuscitation.2005.06.005</u>

- Böttiger, B.W., Becker, L.B., Kern, K.B., Lippert, F., Lockey, A., Ristagno, G., *et al.* (2020) BIG FIVE Strategies for Survival Following Out-of-Hospital Cardiac Arrest. *European Journal of Anaesthesiology*, **37**, 955-958. https://doi.org/10.1097/eja.00000000001247
- [3] Soar, J., Böttiger, B.W., Carli, P., Couper, K., Deakin, C.D., Djärv, T., et al. (2021) European Resuscitation Council Guidelines 2021: Adult Advanced Life Support. *Resuscitation*, 161, 115-151. <u>https://doi.org/10.1016/j.resuscitation.2021.02.010</u>
- [4] Andersen, L.W., Holmberg, M.J., Berg, K.M., Donnino, M.W. and Granfeldt, A. (2019) In-Hospital Cardiac Arrest. *JAMA*, 321, 1200-1210. https://doi.org/10.1001/jama.2019.1696
- [5] Kiguchi, T., Okubo, M., Nishiyama, C., Maconochie, I., Ong, M.E.H., Kern, K.B., *et al.* (2020) Out-of-Hospital Cardiac Arrest across the World: First Report from the International Liaison Committee on Resuscitation (ILCOR). *Resuscitation*, **152**, 39-49. <u>https://doi.org/10.1016/j.resuscitation.2020.02.044</u>
- [6] Hamidi, A.A., Robertson, B. and Ilow, J. (2023) A New Approach for ECG Artifact Detection Using Fine-KNN Classification and Wavelet Scattering Features in Vital Health Applications. *Procedia Computer Science*, 224, 60-67. <u>https://doi.org/10.1016/j.procs.2023.09.011</u>
- [7] Gong, Y., Wei, L., Yan, S., Zuo, F., Zhang, H. and Li, Y. (2023) Transfer Learning Based Deep Network for Signal Restoration and Rhythm Analysis during Cardiopulmonary Resuscitation Using Only the ECG Waveform. *Information Sciences*, 626, 754-772. <u>https://doi.org/10.1016/j.ins.2023.01.055</u>
- [8] Granegger, M., Werther, T. and Gilly, H. (2011) Use of Independent Component Analysis for Reducing CPR Artefacts in Human Emergency ECGS. *Resuscitation*, 82, 79-84. <u>https://doi.org/10.1016/j.resuscitation.2010.08.034</u>
- [9] Amann, A., Klotz, A., Niederklapfer, T., Kupferthaler, A., Werther, T., Granegger, M., et al. (2010) Reduction of CPR Artifacts in the Ventricular Fibrillation ECG by Coherent Line Removal. *BioMedical Engineering OnLine*, 9, Article No. 2. <u>https://doi.org/10.1186/1475-925x-9-2</u>
- [10] Markovsky, I., Amann, A. and Van Huffel, S. (2008) Application of Filtering Methods for Removal of Resuscitation Artifacts from Human ECG Signals. 2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Vancouver, 20-25 August 2008, 13-16. https://doi.org/10.1109/iembs.2008.4649079
- [11] Thannoon, H.H. and Hashim, I.A. (2023) Efficient Enhanced Recursive Least Square Algorithm Adaptive Filtering Scheme for Artifacts Removal in ECG Signals. *e-Prime—Advances in Electrical Engineering, Electronics and Energy*, 6, Article ID: 100318. https://doi.org/10.1016/j.prime.2023.100318
- [12] Alonso, E., Aramendi, E., Irusta, U., Daya, M., Corcuera, C., Lu, Y., *et al.* (2018) Evaluation of Chest Compression Artefact Removal Based on Rhythm Assessments Made by Clinicians. *Resuscitation*, **125**, 104-110. <u>https://doi.org/10.1016/j.resuscitation.2018.01.056</u>
- [13] Delgado-Bonal, A. and Marshak, A. (2019) Approximate Entropy and Sample Entropy: A Comprehensive Tutorial. *Entropy*, 21, Article No. 541. <u>https://doi.org/10.3390/e21060541</u>
- Sharmila, A., Aman Raj, S., Shashank, P. and Mahalakshmi, P. (2017) Epileptic Seizure Detection Using Dwt-Based Approximate Entropy, Shannon Entropy and Support Vector Machine: A Case Study. *Journal of Medical Engineering & Technology*, 42, 1-8. <u>https://doi.org/10.1080/03091902.2017.1394389</u>
- [15] Pincus, S.M. and Keefe, D.L. (1992) Quantification of Hormone Pulsatility via an

Approximate Entropy Algorithm. *American Journal of Physiology-Endocrinology* and Metabolism, **262**, E741-E754. <u>https://doi.org/10.1152/ajpendo.1992.262.5.e741</u>

- [16] Chen, W., Jiang, F., Chen, X., Feng, Y., Miao, J., Chen, S., *et al.* (2020) Photoplethysmography-Derived Approximate Entropy and Sample Entropy as Measures of Analgesia Depth during Propofol-Remifentanil Anesthesia. *Journal of Clinical Monitoring and Computing*, **35**, 297-305. <u>https://doi.org/10.1007/s10877-020-00470-6</u>
- [17] Zhang, H. (2021) Feature Selection Using Approximate Conditional Entropy Based on Fuzzy Information Granule for Gene Expression Data Classification. *Frontiers in Genetics*, **12**, Article ID: 631505. <u>https://doi.org/10.3389/fgene.2021.631505</u>
- [18] Vaz, J.M. and Balaji, S. (2021) Convolutional Neural Networks (CNNs): Concepts and Applications in Pharmacogenomics. *Molecular Diversity*, 25, 1569-1584. <u>https://doi.org/10.1007/s11030-021-10225-3</u>
- [19] LeCun, Y., Bengio, Y. and Hinton, G. (2015) Deep Learning. *Nature*, **521**, 436-444. <u>https://doi.org/10.1038/nature14539</u>
- [20] Khan, S.H., Iqbal, J., Hassnain, S.A., Owais, M., Mostafa, S.M., Hadjouni, M., et al. (2023) COVID-19 Detection and Analysis from Lung CT Images Using Novel Channel Boosted CNNs. *Expert Systems with Applications*, 229, Article ID: 120477. <u>https://doi.org/10.1016/j.eswa.2023.120477</u>
- [21] Shin, H., Roth, H.R., Gao, M., Lu, L., Xu, Z., Nogues, I., et al. (2016) Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. *IEEE Transactions on Medical Imaging*, 35, 1285-1298. <u>https://doi.org/10.1109/tmi.2016.2528162</u>
- [22] Bajić, F. and Job, J. (2021) Chart Classification Using Siamese CNN. *Journal of Im-aging*, 7, Article No. 220. <u>https://doi.org/10.3390/jimaging7110220</u>
- [23] Alhudhaif, A., Polat, K. and Karaman, O. (2021) Determination of COVID-19 Pneumonia Based on Generalized Convolutional Neural Network Model from Chest X-Ray Images. *Expert Systems with Applications*, 180, Article ID: 115141. <u>https://doi.org/10.1016/j.eswa.2021.115141</u>
- [24] AlDuwaile, D.A. and Islam, M.S. (2021) Using Convolutional Neural Network and a Single Heartbeat for ECG Biometric Recognition. *Entropy*, 23, Article No. 733. <u>https://doi.org/10.3390/e23060733</u>
- [25] Huysmans, D., Castro, I., Borzée, P., Patel, A., Torfs, T., Buyse, B., et al. (2021) Capacitively-Coupled ECG and Respiration for Sleep-Wake Prediction and Risk Detection in Sleep Apnea Patients. Sensors, 21, Article No. 6409. https://doi.org/10.3390/s21196409