Portable ECG Monitor on Edge Device using TensorFlow Lite

*Cheonghwan Hwang, **Nagyoum An and *Jaehyun Park
*Department of Electronics and Computer Engineering
**Department of Computer Engineering
Inha University, Korea
chhwang@emcl.org, ymj08073@gmail.com, jhyun@inha.ac.kr

Abstract

In recent years, the healthcare and wellness industry has grown dramatically due to the increased focus on health. Early detection of cardiovascular disease through electrocardiogram monitoring is becoming increasingly important. Recent advances in artificial intelligence have made great strides in the early detection of cardiovascular disease through electrocardiogram. This paper presents a method to monitor electrocardiogram in real time by implementing artificial intelligence technology on edge device using IoT technology, unlike the existing server-based artificial intelligence technology. Since the edge device used in this paper has limited resources, the TensorFlow Lite that is specially designed to support real-time machine learning (ML) inference on low-power and resource-constrained edge devices was used. The performance of the pre-trained model and the lightweight model was compared through TensorFlow Lite to confirm whether the application was suitable.

Topics: Embedded System, Healthcare, electrocardiogram, Deep Learning, TensorFlow Lite Micro

Introduction

Health and wellness are paramount concerns in modern society, leading to significant growth in the healthcare industry. Among various health threats, cardiovascular disease (CVD) stands out as one of the leading causes of death globally. This trend, exacerbated by an aging population, underscores the importance of early and accurate diagnosis. The primary diagnostic tool for detecting heart rhythm disorders, a leading cause of cardiovascular disease, is the electrocardiogram (ECG) which records the heart's electrical activity at specific moments [1], [2].

Myocardial cells, stimulated by the heart's conductive system, undergo cycles of contraction and relaxation. An electrocardiogram captures this electrical activity, with waveforms typically divided into P, QRS Complex, T, and (U) sequences. A normal electrocardiogram cycles in the P-QRS-T-(U) order. The QRS complex represents the electrical activity of the ventricles: The Q waveform starts with the first negative waveform, the R waveform is the first positive waveform representing the polarization of the ventricle, and the S waveform is the second negative waveform representing the non-polarization of the ventricle.

Traditional algorithmic methods utilized in conventional electrocardiogram monitoring systems have been proven as a standard diagnosis method. However, while they showed relatively good performance for the standardized scenarios and data, they showed the limited capacity to the range of an individual's attributes such as heart condition, or changes in response to certain disease states. Recent research has explored the use of machine learning techniques, including deep learning, to identify and analyze individual characteristics and variability in electrocardiogram data with improved accuracy. Deep learning enables automatic identification of complex patterns from large datasets, which results in precise characterization of an individual's electrocardiogram and faster detection of early signs or abnormalities in heart disease [3].

Advancements in computing devices and high-speed mobile networking have enabled global deployment of deep learning applications via the cloud services. Nevertheless, large-scale distributed applications have not been suitable for cloud computing due to the network capacity [4] and latency produced when the application is far from the cloud [5]. To overcome these issues of cloud computing, the edge computing was proposed [6], which carries out tasks at the edge of the network. Machine learning applications including deep learning also likely to use these edge computing concept to expand the real-time and on-line services. One of the widely used deep learning frameworks is TensorFlow, an open source based deep learning framework. To optimize inference speed and enhance performance with reduced memory usage to improve efficiency and minimize size, TensorFlow Lite uses FlatBuffers rather than the standard protocols used in TensorFlow. TensorFlow Lite Micro

is designed for ultra-low power microcontrollers and other devices with limited memory, enabling machine learning models to run effectively with only a few kilobytes of memory [7].

To adopt a deep learning algorithm for cardiovascular disease diagnosis, we utilized the MIT-BIH arrhythmia database as the electrocardiogram dataset for our experiment. This database has been the standard for arrhythmia detector evaluation since 1980. It has been employed in various studies, including foundational research on cardiac mechanics. The data in this database was collected from 47 subjects using two lead configurations: V1 (which can occasionally be V2, V4, or V5, depending on the subject) and Modified Limb Lead II (MLII)[8]. V1 is the precordial (or thoracic) lead, which is in the fourth intercostal space to the right of the sternum. The electrical activity of the heart can be seen in the precordial position. Modified limb leads II (MLII) records the electrical difference between the right arm (cathode) and left leg (anode).

In this paper, we aimed to detect arrhythmia by implementing CNN-based deep learning models specialized in electrocardiogram analysis for embedded devices. Lighten the model for use in microcontrollers, deploy the model with Arduino Nano 33 BLE Sense as Edge Node, and analyze the signal measured by the electrocardiogram sensor in real time.

Proposed method

The proposed hardware system configuration is shown in Figure 1. The electrocardiogram monitoring device utilizes an Arduino Nano 33 BLE Sense supplied by Nordic Semiconductors. The device has a nRF52840 SoC powered by a 32-bit ARM Cortex™-M4 architecture CPU with a floating-point unit, capable of running at 64 MHz [9]. It is energy-efficient, making it ideal for low power usage, and is compatible with the TensorFlow Lite Micro deep learning framework to infer electrocardiogram data as an edge device. With one megabyte of internal flash memory, the device can execute models that are 500-600 KB in size, not including the API programs necessary to operate deep learning models, thus classify three types of images.

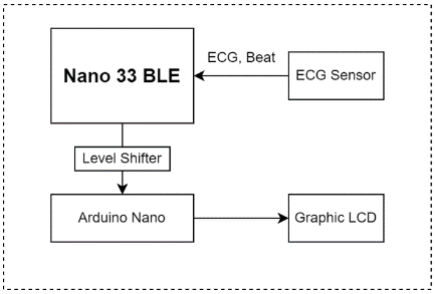

Fig.1: System Block diagram

The MIT-BIH arrhythmia dataset consists of Normal(N), Atrial premature(S), Premature ventricular contraction(V), Fusion of ventricular and normal(F), and Paced(Q) depending on the arrhythmia state [8]. Due to the limited size of the model that can be deployed due to the constraints of device memory space, separate data were used for training and inference. The selected training data was extracted from the MIT-BIH arrhythmia database, categorized as noise data including steady-state electrocardiogram (N) and ventricular premature contraction (PVC) data during arrhythmia and time series data. the QRS waveform is a crucial feature in electrocardiogram. data, providing essential information on the heart's electrical activity. In our dataset, we identified QRS waveforms using vertices with heights ranging from 1250 to 1500 and a 200-unit distance. To optimize the data input size when the model is run on the hardware device, we narrowed our focus to the QRS and P waves by cropping 85 pixels from the left and 15 pixels from the right around the vertices of the identified QRS waveforms. To facilitate data analysis, the image was resized to 96 x 96 using the matplotlib module. Additionally, the data was processed to emphasize the distinguishing characteristics of normal and arrhythmic patterns.

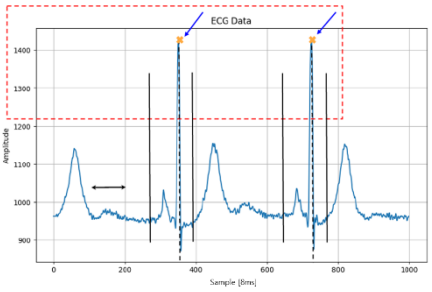

Fig.2: Data Preprocessing

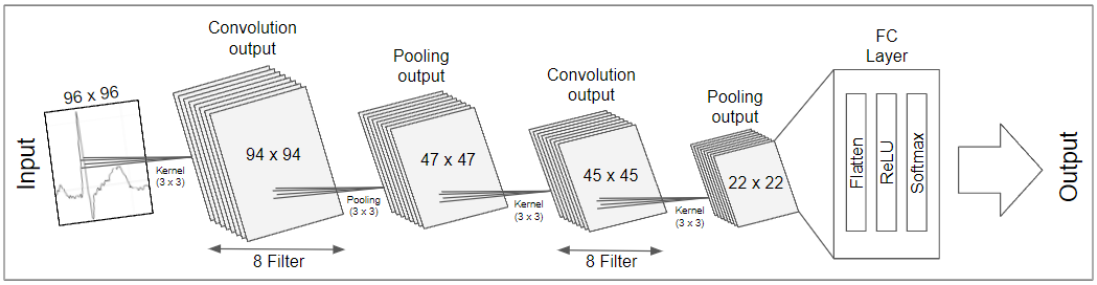

Fig.3: Implemented Model Configuration

The CNN model implemented comprises two convolutional layers, two pooling layers, and one fully connected layer. The fully connected layer is made up of Flatten, ReLU activation function, and Softmax classification. The ReLU activation function was chosen to address the gradient vanishing issue, as it can return 0 for negative input values and its corresponding value for positive ones. While some nodes in the hidden layer may have zero gradients, this does not seem to be a significant issue as the majority have values higher than zero.

As a technique for model optimization, we selected the Adam optimization method, which combines the strengths of Momentum and RMSprop to offer a more effective learning process. We opted for 'categorical_crossentropy' as the loss function, which is appropriate for multi-class classification, considering the scalability of the project to classify input data into more precise categories. 'categorical_crossentropy' is a frequently used loss function that trains a model by minimizing the difference between the probability distributions of each class.

The commonly implemented TensorFlow model has capacity issues that make it difficult to apply to small edge devices using microcontrollers, so there is a need to apply model lightweighting and optimization techniques. When we initially implemented the model with a focus on performance and used 32 filters, the size of the converted TensorFlow Lite model was 2.4MB, which exceeded the flash memory. To solve this problem, we limited the size of the input layer and adjusted the number of filters in the internal convolution layer to implement a model that can be accommodated by limited resources. This approach risks impacting performance, so we conducted several experiments to find the optimal balance between lightweight design and optimal performance, setting the input image to 96 x 96 and using three filters to achieve the best compromise between model accuracy and lightweight structure.

The model was trained using TensorFlow and converted to TensorFlow Lite for microcontrollers for inference on the Arduino Nano 33 BLE Sense. In this process, the training model, consisting of weights and biases in floating point format, is converted to a TensorFlow Lite FlatBuffer file. This file has the weights converted from traditional floating-point weights to an array of C bytes in 8-bit integer format with quantization [10]. The implemented model was 503 KB and deployed normally to the device.

In addition, to improve the model implemented in this way, re-learning was conducted using individual electrocardiogram data. Personal data was collected in a variety of environments and conditions, which were used to improve the generalization ability of the model. The model after re-learning evaluated the difference in accuracy compared to the original model.

	Base model	Re-trained model
	Accuracy	Accuracy
32 Layer	99.23%	99.99%
8 Layer	88.72%	99.99%
TF Lite	53.75%	83.55%

Table 1: Model Performance Comparison

Table 1 displays the accuracy achieved during the model lightening process and its conversion to TensorFlow Lite, as well as the improved accuracy obtained after retraining with individual electrocardiogram data. The results underscore the potential to enhance the model's predictive capabilities in electrocardiogram analysis by incorporating personal data for retraining.

Experiments

The MIT-BIH electrocardiogram database includes two types of leads: Lead I/II (VI) and Modified Lead II (MLII). MLII is a frequently used type that allows for observation of the heart's overall electrical characteristics by measuring electrical activity in a transverse direction. Therefore, we utilized the MLII lead type to capture and analyze the characteristics of the QRS complex more effectively.

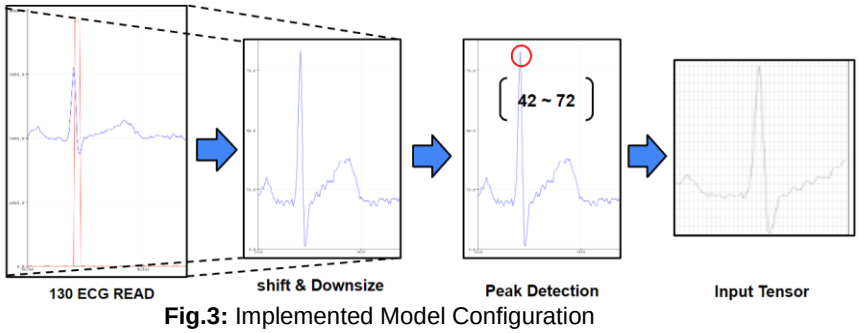

Fig.3: Implemented Model Configuration

Electrocardiogram data were sampled using the MLII lead method. Each sample included 130 values at intervals of 8.3ms (70 SPS). To adjust the range of the electrocardiogram data, the minimum value was shifted closer to zero. The 12-bit resolution data values were then reduced from a maximum of 4095 to 96 to boost processing efficiency. The R-wave position of the electrocardiogram signal was processed to be centered by verifying whether the signal values above 70 were between 48 and 72 and centered accordingly. To enhance computational efficiency, the electrocardiogram values at the (x + peak - 48)th electrocardiogram array location were converted to grayscale. This reduced the data complexity and minimized the model computation. Using this experiment method, the data was processed in a way that captured the ventricular arrhythmias' features and maximized the model's computational efficiency.

Since direct experiments on arrhythmia patients are limited, we developed a validation process utilizing genuine patient electrocardiogram data. The following section provides a detailed description of the methodology used to validate the model. To minimize the data size while maintaining the original electrocardiogram waveform characteristics, the 96 x 96 electrocardiogram waveform images (.bmp) utilized for training were converted to 1-bit values using a tool. Instead of using the signal from the actual electrocardiogram sensor, we directly input the 1-bit converted electrocardiogram waveform data that was preprocessed above into the model. This serves as a proxy for the absence of real patient data, and we use it to validate the model's ability to make predictions on the data it was trained on. We cross-fed the arrhythmia and normal heart rate data to evaluate the model's performance and ensure its ability to make predictions in diverse situations. This provided an initial evaluation of the functionality and user-friendliness of the model in a practical context on an edge device.

Conclusion

In this paper, we proposed a deep learning-based edge computing node that can be used for the diagnosis of Cardiovascular disease with real-time monitoring an electrocardiogram. To implement real-time inference, the proposed system utilized TensorFlow Lite Micro engine on a Arduino Nano 33 BLE. We used a CNN-based model to convert electrocardiogram waveforms into images of heart rate units and the converted images were trained. The learned model was optimized for a tiny edge computing device where the real-time electrocardiogram signal was captured, and inference of diagnosis was performed using TensorFlow Lite Micro.

Since the electrocardiogram was learned via a method based on deep learning, it can easily analyze diverse environments and complex data patterns with high accuracy, depending on the model and trained data. Furthermore, by retraining the individual's heartbeat data, it is expected to exhibit high accuracy that is specific to the user.

Acknowledgment

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Innovative Human Resource Development for Local Intellectualization support program(IITP-2023-RS-2023000259678) supervised by the IITP(Institute for Information & communications Technology Planning & Evaluation)

The authors would like to express their appreciation to Hoyoung Son and Minsub Kim at Inha University for helping experiments to improve the quality of this paper.

References

[1] D. Lloyd-Jones "Heart disease and stroke statistics 2009 update, A report from the American Heart Association statistics committee and stroke statistics subcommittee", Circulation, vol. 119, no. 3, pp. 21 - 181 2009.

[2] D. Buxi et al., "Correlation Between Electrode-Tissue Impedance and Motion Artifact in Biopotential Recordings," in IEEE Sensors Journal, vol. 12, no. 12, pp. 3373-3383, Dec. 2012.

[3] P. Rajpurkar, et., “Cardiologist-level arrhythmia detection with convolutional neural networks,” arXiv preprint arXiv:1707.01836, 2017.

[4] X. Ke, W. Hou and L. Meng, "Research on Pet Recognition Algorithm With Dual Attention GhostNet-SSD and Edge Devices," in IEEE Access, vol. 10, pp. 131469-131480, 2022.

[5] C. A. A. Valencia, R. S. S. Suliva and J. F. Villaverde, "Hardware Performance Evaluation of Different Computing Devices on YOLOv5 Ship Detection Model," 2022 IEEE 14th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), Boracay Island, Philippines, pp. 1-5, 2022.

[6] N. T. Ha et al., "Leveraging Deep Learning Model for Emergency Situations Detection On Urban Road Using Images From CCTV Cameras," 2022 International Conference on Engineering and Emerging Technologies (ICEET), Kuala Lumpur, Malaysia, pp. 1-5, 2022.

[7] TensorFlow Lite for Microcontrollers, Aug 2023, [online] Available: ttps://www.tensorflow.org/lite/microcontrollers.

[8] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH Arrhythmia Database," in IEEE Engineering in Medicine and Biology Magazine, vol. 20, no. 3, pp. 45-50, May-June 2001.

[9] nRF52840, Aug 2023, [online] Available: https://www.nordicsemi.com/products/nrf52840.

jhyun@inha.ac.kr