Machine Learning-based Arrhythmia Diagnosis Algorithm

Dong Hyun Kim, Jae Min Lee, and Jong Deok Kim*
Department of Computer Science and Engineering
Pusan National University
Pusan, South Korea
e-mail: {dhkim1106, ashesmin, kimjd}@pusan.ac.kr

Yeong Joon Gil
Department of Research and Development
HUINNO
Seoul, South Korea
e-mail: kyzoon@huinno.com

***Abstract*— As interest in cardiovascular disease increases, there is significant development in real-time healthcare services and devices. This paper suggests a machine learning-based arrhythmia diagnosis algorithm for mobile healthcare systems linked to a wearable electrocardiogram measurement device. The system monitors electrocardiograms in real time and distinguishes among arrhythmia, normal, and noise signals. By regular monitoring using a mobile healthcare system linked to a wearable bio-signal measurement device, users can minimize the risk of chronic disease and maintain a healthy standard of living. In this paper, an arrhythmia diagnosis algorithm is suggested, realized, and evaluated based on bio-signals collected from a watch-type electrocardiogram device. In the process, the efficacy of the algorithm is established**

Keywords-* *Arrhythmia; Atrial fibrillation; Machine Learning.

I. INTRODUCTION

The U-health care technology has allowed health management that used to be performed in a hospital to become a daily standard of care. This concept has been actively developed along with the dissemination of smart devices. In particular, U-health care devices that assist in the prevention of chronic diseases, such as obesity, diabetes mellitus, and cardiovascular diseases are in active development.

The risk of incidence of cardiovascular diseases demands the need for regular monitoring of electrocardiograms (ECGs), especially for those individuals who have medical history require continuous management regardless of time and place. U healthcare is the newest collection of digital-convergence that combined Information Technology (IT) and Biology Technology (BT) to provide remote services of health and medical care without any limitations of time or place. As real-time electrocardiogram signal processing and classification technologies are realized, early diagnosis of cardiovascular diseases that required real-time monitoring became possible, which is a great achievement [1]-[3].

The electrocardiogram signals are electric waves that are generated by a sequence of depolarization and repolarization of the atria and ventricles, which form repetitive periodic curves. These signals can express various abnormalities of the heart, and each condition exhibits characteristic waveforms. In this paper, a low-layer deep learning algorithm that distinguishes Atrial fibrillation(A-fib), Sinus rhythm, and noise is designed, realized and evaluated, based on data collected from watch-type electrocardiogram devices.

II. RELATED WORK

In this section, we describe existed machine learning platform and machine learning algorithm

A. Machine learning platform

In November 2015, Google released its open-source library TensorFlow that was developed for its vital technology of machine learning. Source code and API are open sources built on top of TensorFlow, which can be accessible under the Apache License 2.0 [4].

Anyone with Apache License 2.0 can create new software and transfer its copyright by utilizing the TensorFlow software. In addition, the software developed by utilizing TensorFlow can be used for commercialization purpose.

The engine built for TensorFlow is significantly flexible and can be executed on a CPU that is computer-sourced, thus enabling data analysis and GPU. In addition, it utilizes a multi-core processor of a desktop PC, such as dual-core or quad-core. It is accessible in a server environment that is built by virtualization technology or thousands of big data sensors [5].

TensorFlow works on both a PC architecture, as well as on mobile devices. Google reported that the relative API of this algorithm uses the same code regardless of the device. TensorFlow can be utilized with Python, which is often used in C++ and machine learning. However, there are several limitations of the algorithm. Firstly, it does not facilitate Cloud service [6]. Google has combined TensorFlow with various services and applied machine learning methods, but there are still some systems that cannot utilize this package. For devices that do not support TensorFlow, their systems should be independently developed. Secondly, Google does not provide any guideline regarding TensorFlow or any data to utilize using the algorithm, except the frame that can analyze data. Creation and application of modes are thus realized by individual programmers. TensorFlow is a computing library that realizes an artificial neural network of data through data flow graphs. The data flow graphs consist of nodes and edges. Nodes calculate the output and edges express the relation between the input and output of the nodes. The values calculated from a node are transferred to a multidimensional arrange through an edge, while nodes read the values from the multidimensional arrange and calculate them in parallel [7].

B. Machine learning algorithm

Convolution is an operation that is used to extract a feature of interest in a given image using certain filters. s(t) is referred to as a feature map of data xx. Prior to the introduction of deep learning, in order to process an image as an input in other machine learning frameworks, a filter was chosen, preprocessing was performed which involved the convolution of an image with the chosen filter to produce a proper feature map. Then, this result was used as an input to the machine learning framework. This feature engineering process significantly impacted overall performance. Since the choice of filters or the number of filters used are part of the feature engineering process and not a theoretical construct, this area was not viewed as an interesting area in machine learning [8].

The critical idea of Convolution Neural Network (CNN) is to create a model that learns the best convolution filter that can extract the best feature map, and effectively performs preprocessing, since processing significantly affects overall performance. Therefore, there are three key ideas in CNN to realize the best filter with minimum complexity and include; sparse interactions (or sparse weight), parameter sharing (or dense weight), and equivariant representations. In other words, CNN connects every layer but only partially, which is described as the sparse weight. These weights are considered different random variables and updated respectively, while a certain weight group shares parameter to balance the weight value (parameter sharing). Then, based on this idea, a new model that is going to learn representation equivariant to transform such a shift is constructed [9].

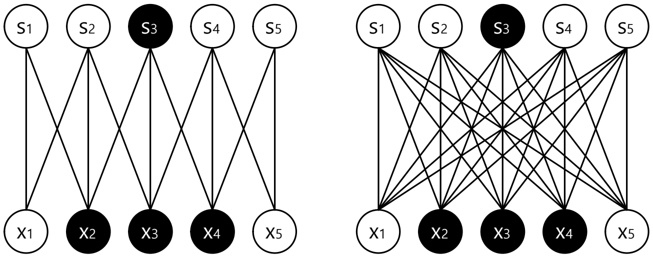

Figure 1. edges and input nodes influence output s3

Figure 1 illustrates the edges that influence the same output s3 and input nodes. The left diagram shows that not every connection is available and only inputs x2, x3, x4 that influence s3. However, in the right diagram, there is a difference of model parameters in available connection and every input influences s3 [10].

However, in the right diagram, there is a difference of model parameters in available connection and every input influences s3.

To process the real image data, the left model works better because only part of information of the image is needed to decide a certain feature. Equivalent representations mean that the changing method of output along with changes of the input becomes equivariant when sparse weight and tied weight are effectively arranged in a certain form. For example, if function f is equivariant to function g, this can be expressed as f(g(x)) = g(f(x)). For image processing, g means a random value of a linear transform. This represents the transform of an image such as shifting, rotation and scaling. The image can still be recognized if it is rotated, moved, or scaled, but a computer cannot identify this same image since the transform results in changes to individual pixel values. If we could create a network *f* that makes an equivariant representation to a certain transform *g*, this network could have a fixed representation as the input regardless of its shift or rotation. Previously, shared parameters were set to process the same filter for each patch. If the image was shifted, then feature map is not distorted but shifted along with the image [10].

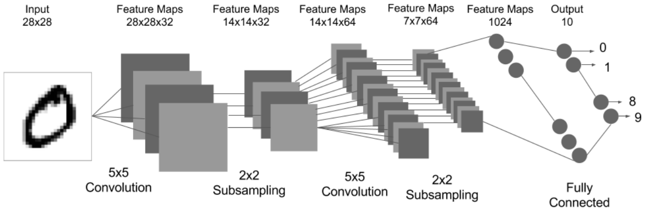

Figure 2. Construction map of Convolutional Neural Network (CNN)[11]

However, every image might need more than one filter. The previous convolution layer expresses only one convolution filter, but several convolution filters might be needed to concatenate and create a feature map. Therefore, the CNN model is in the form of several convolution layers that are combined as subsequently indicated. For reference, each layer or filter is officially called a kernel, and the set of kernels are called one layer [12].

III. DESIGN OF PREPROCESSING AND ARRHYTHMIA ALGORITHM

In this section, we describe suggested ECG data format and deep learning network structure

A. Preprocessing

The data used in this research were collected from watch-type wearable devices of HUINNO Co. To use ECG data set stored as time series form in a CNN algorithm, the ECG data stored every 15 min (900 s) were set as 10 s unit of sampling form. The frequency of the watch-type ECG sampling is 300 Hz.

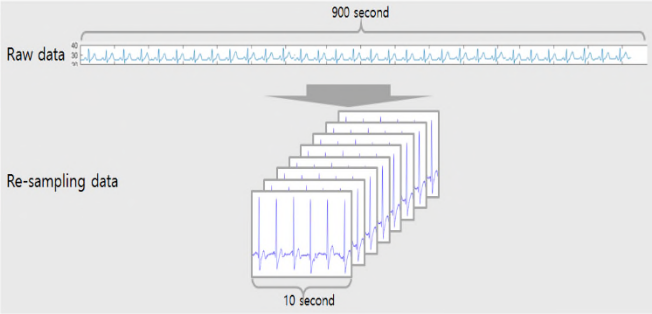

Figure 3. Re-sampling structure with existing ECG data format adjusted to DL engine

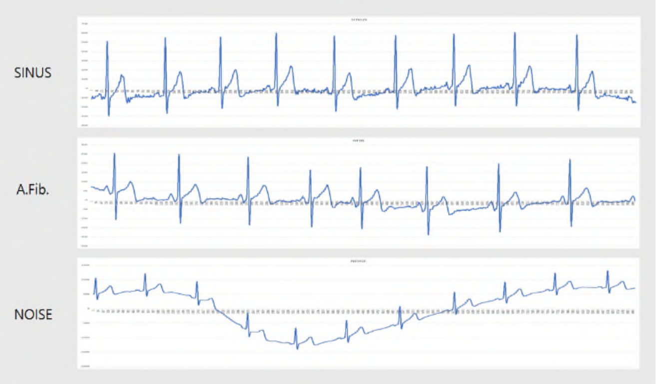

Figure 4. HUINNO SINUS ECG, A.Fib ECG, Noise ECG data sample for HUINNO.

B. Design of arrhythmia algorithm

The neural network comprises seven layers: five convolutional layers and two fully connected layers. On every convolutional layer, batch normalization and max-pooling are executed. The first convolutional layer is set with 48 filters and 11 kernels while the second one consists of 128 filters and 5 kernels.

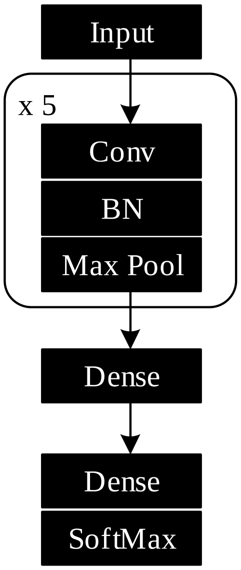

Figure 5. Suggested deep learning network structure

TABLE I. THE NUMBER OF SINUS, A-FIB, NOISE DATA CREATED THROUGH PREPROCESSING

	Sinus	A-fib	Noise	Total
Number of data set	130,496	22,390	18,720	171,606

The first convolutional layer is set with 48 filters and 11 kernels while the second one consists of 128 filters and 5 kernels. The third one contains 192 filters and 3 kernels, the fourth one has 192 filters and 3 kernels, while the last one consists of 128 filters and 3 kernels The Max polling layer moves in between 3 kernels and 2 strides to collect the maximum values in every layer. The two final fully connected layer comprises of 2048 filters. As previously explained, the configuration of each layer is described in Table 2.

TABLE II. THE CONFIGURATION OF EACH LAYER

Layer	Output Shape	Param#
Conv1	(2997, 48)	576
Pooling1	(1498, 48)	-
Conv2	(1494, 128)	30,848
Pooling2	(746, 128)	-
Conv3	(744, 192)	73,920
Pooling3	(371, 192)	-
Conv4	(369, 192)	110,784
Pooling4	(184, 192)	-
Conv5	(182, 128)	73,856
Pooling5	(90, 128)	-
Flatten	(11, 520)	-
Dense6	(2048)	23,595,008
Dense7	(2048)	4,196,352
Dense8	(3)	6,147

IV. PERFORMANCE EVALUATION ENVIRONMENT AND ANALYSIS

In the overall data, the number of data classified into Noise was the smallest. Therefore, the algorithm was created with these 15,000 Noise data, and the rate of Sinus and the A-Fib dataset was adjusted to that of the noise data.

Figure 6 shows the confusion matrix of the sinus, A-fib, noise of the suggested model. It is important to apply the algorithm in a remote medical area to determine the accuracy of the approach in determining the A-fib signal as A-fib. In 3000 test sets, this algorithm identified A-fib signals with 100% accuracy, noise at 82.4%, and sinus at 39.4%. Only three classes were expressed using a simple algorithm to increase efficiency.

TABLE III. TRAINING, VALIDATION, TEST DATA CONDITION

	Group	Sinus	A.Fib	Noise
Training	A group	9,000	9,000	9,000
Validation		3,000	3,000	3,000
Test	B group	3,000	3,000	3,000
Total	-	15,000	15,000	15,000

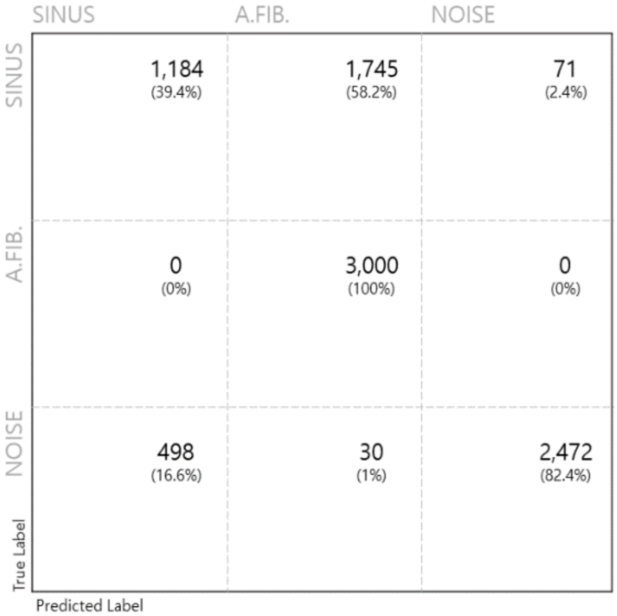

Figure 6. Confusion matrix for the model predictions on the test set.

V. CONCLUSION

In this paper, we designed and realized a deep learning algorithm in a mobile healthcare system for real-time processing of electrocardiogram data. A simple deep learning network was designed with Sinus, A-fib, and noise.

Then, the collected information from watch-type electrocardiogram device was learned, evaluated and tested. The result showed a high accuracy for A-fib signal detection, which could be applied to a remote medical system. There are approximately 12 signals, which indicate an abnormality in the functioning of the heart. Based on our findings, a deep learning algorithm to detect these signals could be designed and developed and its accuracy may be improved through further research.

ACKNOWLEDGMENT

"This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (No. 2016R1A2B4013150)."

* : Corresponding Author

REFERENCES

[1] K Itao, T Umeda, G Lopez, and M Kinjo, "Human recorder system development for sensing the autonomic nervous system", IEEE Sensors, pp423-426, 2008.

[2] Corventis Corporation. corventis products [Internet]. Available: http://www.corventis.com. [retrieved: Jun, 2018]

[3] Isansys Lifecare Corporation. isansys products [Internet]. Available: http://isansys.com/. [retrieved: Jun, 2018]

[4] Tensorflow [Internet]. Available : http://www.tensorflow.org/ [retrieved: Jun, 2018]

[5] Jeffrey Dean and Rajat Monga, TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, Preliminary White Paper, November 9, 2015.

[6] Automatic differention [Internet]. Available : https://en.wikipedia. org/wiki/Automatic_diff erention [retrieved: Jun, 2018]

[7] Four Reasons Why Google's Tensor Flow Disclosure Is Important [Internet] Available : http://www.itworld.co.kr/news/96498 [retrieved: Jun, 2018]

[8] R. Collobert et al, "Natural language processing (almost) from scratch," Journal of Machine Learning Research 12(Aug), pp. 2493-2537, 2011

[9] S. Joty and E. Hoque, "Speech act modeling of written asynchronous conversations with task-specific embeddings and conditional structured models," Proc. of the 54th Annual Meeting of the Association for Computational Linguistics, ACL, pp. 7-12, 2016.

[10] M. Kim and H. Kim, "User Utterance Analysis using Convolutional Neural Network," Journal of Telecommunications and information, Vol. 21, pp. 5-7, 2017.

[11] Implement Convolutional Neural Networks (CNNs) for MNIST number classification using TensorFlow [Internet] Available : http://solarisailab.com/archives/1308 [retrieved: Jun, 2018]

[12] X. Wang, W. Jiang, and Z. Luo, "Combination of convolutional and recurrent neural network for sentiment analysis of short texts," Proc. of the 26th International Conference on Computational Linguistics, pp. 2428-2437, 2016.