Optimization and Implementation of a Collaborative Learning Algorithm for an AI-Enabled Real-time Biomedical System

Sinchhean Phea,*, *Zhishang* Wang, *Jiangkun* Wang, and *Abderazek* Ben Abdallah

Adaptive Systems Laboratory, School of Computer Science and Engineering, University of Aizu, Aizu-Wakamatsu, Japan

Abstract. Recent years have witnessed a rapid growth of Artificial Intelligence (AI) in biomedical fields. However, an accurate and secure system for pneumonia detection and diagnosis is urgently needed. We present the optimization and implementation of a collaborative learning algorithm for an AI-Enabled Real-time Biomedical System (AIRBiS), where a convolution neural network is deployed for pneumonia (i.e., COVID-19) image classification. With augmentation optimization, the federated learning (FL) approach achieves a high accuracy of 95.66%, which outperforms the conventional learning approach with an accuracy of 94.08%. Using multiple edge devices also reduces overall training time.

1 Introduction

Efficient and accurate diagnosis of biomedical signals and images plays a significant role in health care systems [1, 2]. In the conventional medical system, diagnosis is managed by doctors or experts. Such procedures usually take long time, and when there is rapid increase in the number of patients, the human experts face a big challenge of working effectively while avoiding medical errors [3, 4]. Particularly, the spread of COVID-19 disease has shown that a more reliable and fast diagnosis system is urgently needed [5, 6].

Convolution neural networks (CNNs) have been widely used in many studies, e.g., speech recognition and image classification [7–9]. As CNNs show promising performance in feature extraction and representation, they have become a favourable choice in current biomedical area [10, 11].

While the aforementioned studies have achieved promising results, there remain some disadvantages of the state-of-the-art approaches. First, collection and labeling of medical data are highly-cost and time-consuming, thus the size of data set is limited. Besides, due to privacy concerns of medical data, sharing data among hospitals is barely possible, which greatly influences the accuracy of diagnosis. Moreover, the training process is usually slow, for the reason that it takes time to gather the distributed data together and performs training on a single machine.

To address these issues, we propose to deploy collaborative learning algorithm for AI Enabled Real-Time Biomedical System (AIRBiS) [12]. Considering the limited amount of medical image samples, augmentation technique is used to increase the size of data set. We then integrate the federated learning (FL) approach [13] into a CNN. The FL algorithm aims to secure the data privacy while accelerating the overall learning process. Additionally, the communication protocol between the aggregator and clients is implemented.

Our main contributions are: (1) Augmentation technique on chest X-ray data set is carefully investigated. (2) Implementation and optimization of collaborative/federated learning scheme for AIRBiS system. (3) A client-server protocol for real application of FL on pneumonia is studied.

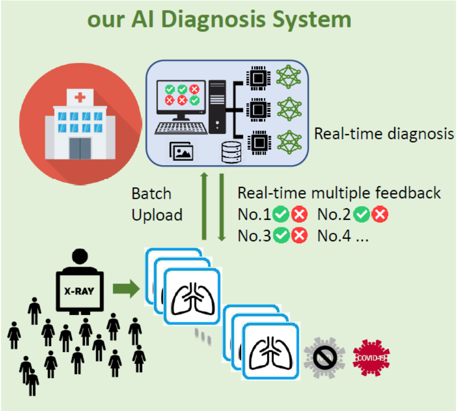

Figure 1. AIRBiS Overall Organization: Chest X-ray images are diagnosed by a automated system and results are then shown on an interactive user interface [12].

1.1 AIRBiS Overview

Fig. 1 shows the AIRBiS high-level view. The system aims to build an automatic pneumonia detection and diagnosis system to achieve highly efficient and accurate performance. In the detection process, CNN is used on chest X-ray images to predict whether a chest is "normal" or "abnormal." If the prediction result shows "abnormal,"

*e-mail: s1250250@u-aizu.ac.jp

the diagnosis process gives a detailed analysis of the disease. The system consists of a CNN-based detection platform, user interface, and AI hardware that helps accelerate the inference process. The present study is a part of the AIRBiS project and focuses on the detection procedure's training process.

2 Collaborative Learning

In this section, we present a detailed description of the implemented federated learning procedure together with an augmentation technique and the client-server protocol. Federated Learning [14] is a machine learning approach that trains using local data residing across multiple edge nodes or clients without transferring the data.

2.1 Federated Learning (FL) Procedure

In the FL architecture, each hospital or clinic acts as a client. Each client shares and updates the local model via an aggregator, as shown in Fig. 2. The FL algorithm mainly consists of the following five steps:

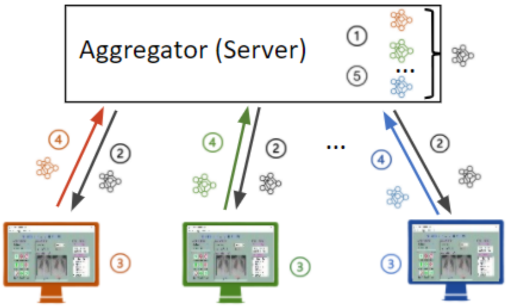

Figure 2. Federated Learning Overview. The computers represent the systems residing in hospital facilities which act as edge devices to perform the training process. The aggregator collects and updates the models.

1) The global model M_G is initiated by the aggregator.

2) The model is sent from the aggregator to the clients:

$$M_{C_i} \leftarrow M_G \tag{1}$$

where M_C denotes the local model, and i denotes the identification (ID) of the client.

3) For each local model, the parameters are updated during training using the local data set:

$$w_i \leftarrow w_i - \eta \frac{\delta Error}{\delta w_i} \tag{2}$$

$$b_i \leftarrow b_i - \eta \frac{\delta Error}{\delta b_i} \tag{3}$$

where w and b denote the weights and biases of the model respectively, and η is the learning rate. $Error$ denotes the cost function.

4) All the trained models are uploaded from clients to the aggregator. Also, the aggregator is informed of the size of data on each node.

5) The aggregator collects all local models, resulting in an updated global model:

$$w \leftarrow \sum_{i=1}^{K} \frac{n_i}{n} w_i \tag{4}$$

$$b \leftarrow \sum_{i=1}^{K} \frac{n_i}{n} b_i \tag{5}$$

where n denotes the total amount of data sets, n_i denotes the number of data on client i, and K is the total number of clients.

Steps 2 to 5 should be repeated multiple times until the model converges and performance stops improving after a number of rounds.

2.2 Real-time Augmentation

When the number and quality of training data increase, the performance of the CNN model also increases [15]. Data augmentation helps increase the training data size by creating modified versions of the existing data. However, for FL system, the local participants' storage space might not be enough. Due to this reason, a real-time augmentation needs to be applied on the client side. The Real-time augmentation means the augmented images are created dynamically and only produced during training, which will not be stored on the client side afterward. Thus, the clients' data can be increased while making no changes to the local data.

2.3 Client-Server Protocol

In this study, the socket programming is used. On the client side, the server IP address and port must be specified in advance. On the server side, if clients' IP addresses are recorded beforehand, the server can decline the clients that are not in the list. Fig. 3 illustrates the FL protocol for multiple clients.

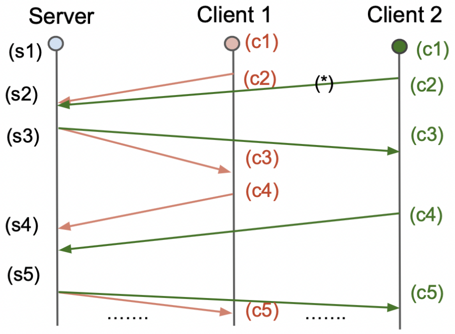

Figure 3. FL Protocol for Multiple Clients.

*Note: For efficiency, the connection initialization for both the server and clients should start at the same time, which can be done using scheduling.

The detailed communication process is as follows:

s1	- Initialize a global model - Create and bind the socket to an IP address (of the server) and port number - Listen for the connections
c1	- Create a socket and connect to the server (indicate server IP and port number)
c2	- Request to join or connect to the server
s2	- Accept the incoming connections
s3	- Send the global model to the clients
c3	- Accept and train the model using client's local data
c4	- Send the locally trained model (and data size) to the server
s4	- Accept the data from clients
s5	- Aggregate and update local models to a global model - Evaluate the model: If it reaches threshold accuracy (e.g., 90%), save the model and also send the model and "Done" message to the clients and stop. Otherwise, repeat from step s2
c5	- Receive the new global model and stop if "Done" message is included, otherwise, continue the training

In order to handle multiple incoming connections from clients at the same time, multi-threading execution is used on the server. We use scheduling to start initialization for both server and client side. This reduces resource consumption on the server side because the server does not have to always be running to wait for the clients.

As shown in Fig. 3, suppose that the training time on client 1 and client 2 is different, thus the server will also receive the local models at different time. One solution to this synchronization problem is to compare the the number of clients connecting at step s4 and the number of clients that have joint at step s2. The number of clients at step s2 will always be greater or equal to the number of clients at step s4:

$$N_r^{s2} \geq N_r^{s4} \tag{6}$$

where N denotes the number of clients, and r denotes the round number. Once all clients' trained models have been received, the server will start the aggregation task.

To ensure the efficiency of the protocol, the timeout setting is used to alleviate late response. If one client's connection has timeout at step s4, the server will disregard that client in that round.

The server has to wait for all clients' response. A same timeout is set for all clients. Server side's timeout is the sum of training time and transmission time:

$$T_s = T_{ct} + T_{tr} \tag{7}$$

where T_s denotes the timeout for server, T_{ct} is the average training time taken by clients, and T_{tr} denotes the data transmission time.

The clients, on the other hand, have to wait for server response. The client side's timeout is less than or equal to the sum of the aggregation time, transmission time, and the server timeout:

$$T_c \leq T_a + T_{tr} + T_s \tag{8}$$

where T_c denotes the timeout for clients, T_a is the aggregation time,
The other perspective of the protocol is shown in Fig. 4, which summarizes the entire communication process.

3 Evaluation

3.1 Evaluation Methodology

We collected the chest X-ray image data set from two sources [16, 17]. The data set contains two categories: normal (healthy) and abnormal (afflicted with many types of pneumonia including COVID-19). As shown in Table 1, there is an unbalanced size between normal and abnormal data. We noticed that for different chest X-ray images, the position of lung and the contrast of image varied. Therefore, we applied fixed augmentation of contrasting and shifting to the normal data.

		Training set	Testing set
ABNORMAL	pneumonia	3875	390
	COVID-19	400	68
NORMAL	original	1341	234
	augmented	2934	-
Total		8550	692

Table 1. Chest X-ray Data Sets.

We considered five clients, and the data set was allocated equally to each client. The number of epochs and batch-size were tuned in the experiments. Moreover, a real-time augmentation of zooming was used during training. Zoom augmentation resizes the key area of the chest X-ray images, thus the features can be better extracted. The data produced by real-time augmentation was not stored on the client side after training, which was a countermeasure against limitation of client's storage space.

The CNN model's architecture in the experiment is presented in Table 2. Adam optimizer is used on batch data set. A rectified linear unit activation (ReLU) function is applied after each convolutional layers. Sigmoid function is used at the last fully-connected layer for binary classification.

Layer	Input	Output
Conv1	(64,64,3)	(62,62,32)
Max-pool1	(62,62,32)	(31,31,32)
Conv2	(31,31,32)	(29,29,32)
Max-pool2	(29,29,32)	(14,14,32)
Conv3	(14,14,32)	(12,12,32)
Max-pool1	(12,12,32)	(6,6,32)
FC1	1152	128
FC2	128	1

Table 2. CNN Model for Evaluation

https://doi.org/10.1051/shsconf/202110204017

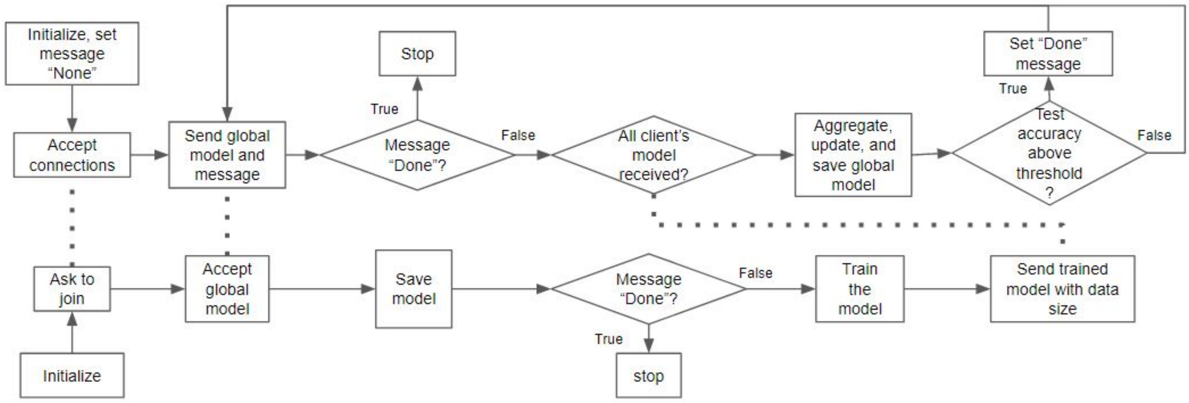
Initialize, set message "None"
Accept connections
Send global model and message
Stop
True
Message "Done"?
False
All client's model received?
Aggregate, update, and save global model
Set "Done" message
True
Test accuracy above threshold ?
False
Ask to join
Initialize
Accept global model
Save model
Message "Done"?
False
True
stop
Train the model
Send trained model with data size

Figure 4. Communication process between the server and clients. The dotted lines represent the message exchanged between the server and clients, while the solid lines correspond to the internal process. The "Done" message is created only when the test accuracy exceeds a satisfactory threshold.

3.2 Evaluation Results

For FL implementation, we conducted two evaluations. Firstly, we compared the performance with and without real-time zoom augmentation from 1 client's data to total 5 clients' data. Secondly, we compared the impact of using different number of epochs and batch size. Among these results, the setting that achieved best accuracy was used to compare with the conventional learning approach.

As shown in Fig. 5, training on a single node using real-time augmentation outperforms the collaborative learning without real-time augmentation. Besides, the accuracy increases when more clients join the learning process. As illustrated in Fig. 6 and 7, when training on the client side, using more training epochs or smaller batch size leads to fewer rounds of communication. The averaging method of the collaborative learning scheme achieves competitive performance compared to the conventional approach as shown in Fig. 8.

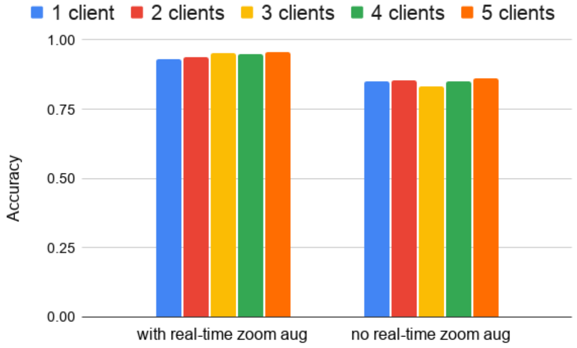

Figure 5. Performance of collaborative (FL) with and without real-time zoom augmentation. The real-time augmentation increases accuracy of the model.

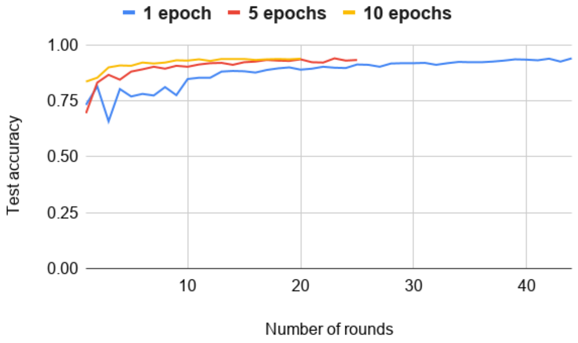

Figure 6. Performance of FL with different number of epochs. Running more epochs in each round leads to faster convergence.

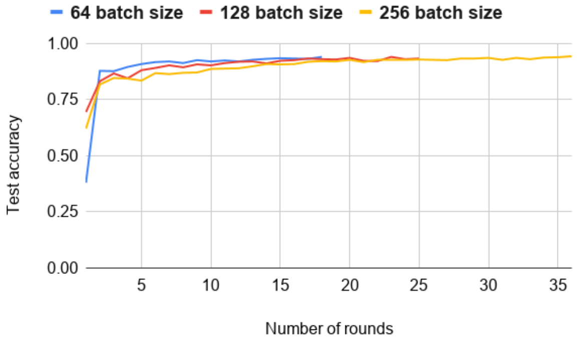

Figure 7. Performance of FL with different batch-size. The training contained 5 epochs in each round. Smaller batch size resulted in more number of batch, which leads to faster convergence.

4 Discussion

The collaborative learning approach has shown satisfactory performance for chest X-ray image classification. With the real-time augmentation in the training stage, the performance of the training process was significantly improved. In the FL scheme, the model instead of raw data is exchanged, which not only protects the data privacy and speeds up the communication. The simple CNN architecture greatly reduces training time and communication delay.

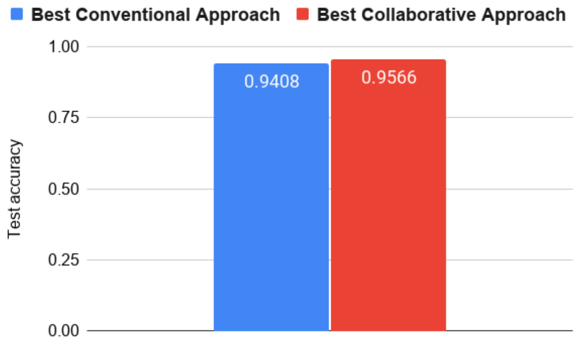

Figure 8. Comparison between the collaborative (this work) and conventional learning approach. Both uses the same data and training parameters.

In our experiments, the data is independent and identically distributed (IID). In practice, however, for the reason that the amount and healthy status of patients vary in each hospital, local data sets may have different characteristics, leading to not independent and identically distributed (non-IID) type of data. Moreover, different training time at each client side remains a challenge for the server to coordinate.

When data is equally distributed, FL can be considered to be a map-reduce distributed processing scheme. Though, map-reduce goal is to reduce the training time using parallelism, which is one of the characteristics of FL.

5 Conclusion and Future Work

In this paper, we have implemented a collaborative learning approach in CNN implementation for pneumonia image detection. We applied the augmentation technique to the federated-learning scheme, achieving high accuracy for a binary classification task. The accuracy results are very competitive compared to the conventional approaches, while data privacy is also preserved.
Future work is to optimize the communication protocol, investigate other learning approach using non-IID data over real-world experiments.

References

[1] Achraf Ben Ahmed, Yumiko Kimezawa, Abderazek Ben Abdallah, "Hardware/software prototyping of dependable real-time system for elderly health monitoring", World Congress on Computer and Information Technology (WCCIT) 2013, pp. 1-6, 2013.

[2] Achraf Ben Ahmed, A. Ben Abdallah, ”Architecture and Design of Real-Time Systems for Elderly Health Monitoring,” Journal of Embedded Systems, 2017, Vol.9, No.5, pp.484 – 494, DOI: 10.1504/IJES.2017.10007717

[3] G. C. Peng, “Moving toward model reproducibility and reusability,” *IEEE Trans. on Biomedical Engineering*, vol. 63, no. 10, pp. 1997–1998, 2016.

[4] S. L. Grimes, “The challenge of integrating the healthcare enterprise,” *IEEE Engineering in Medicine and Biology Magazine*, vol. 24, no. 2, pp. 122–124, 2005.

[5] Q.-V. Pham, D. C. Nguyen, W.-J. Hwang, P. N. Pathirana *et al.*, “Artificial intelligence (ai) and big data for coronavirus (covid-19) pandemic: A survey on the state-of-the-arts,” 2020.

[6] B. Benreguia, H. Moumen, and M. A. Merzoug, “Tracking covid-19 by tracking infectious trajectories,” *IEEE Access*, vol. 8, pp. 145 242–145 255, 2020.

[7] Y. LeCun, Y. Bengio *et al.*, “Convolutional networks for images, speech, and time series,” *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[8] The H. Vu, Ryunosuke Murakami, Yuichi Okuyama, and Abderazek Ben Abdallah, ”Efficient Optimization and Hardware Acceleration of CNNs towards the Design of a Scalable Neuro-inspired Architecture in Hardware,” IEEE Int. Conf. on Big Data and Smart Computing (BigComp 2018), Shanghai, China, January 15-18, 2018.

[9] Tomohide Fukuchi, Ogbodo Mark Ikechukwu, and Abderazek Ben Abdallah. ”Design and Optimization of a Deep Neural Network Architecture for Traffic Light Detection,” ACM Chapter Int. Conf. on Educational Technology, Language and Technical Communication (ETLTC), January 27-31, 2020, Aizuwakamatsu, Japan.

[10] M. Sun, F. Wang, T. Min, T. Zang, and Y. Wang, “Prediction for high risk clinical symptoms of epilepsy based on deep learning algorithm,” *IEEE Access*, vol. 6, pp. 77 596–77 605, 2018.

[11] L. Liu, F.-X. Wu, Y.-P. Wang, and J. Wang, “Multi-receptive-field cnn for semantic segmentation of medical images,” *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 11, pp. 3215–3225, 2020.

[12] Abderazek Ben Abdallah, Huakun Huang, Nam Khanh Dang, and Jiangning Song, "AI processor", Japanese Patent Application Laid-Open No 2020-194733.

[13] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[14] T. Li, A. K. Sahu, A. Talwalkar and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions," in IEEE Signal Processing Magazine, vol. 37, no. 3, pp. 50-60, May 2020, doi: 10.1109/MSP.2020.2975749.

[15] W. Li, C. Chen, M. Zhang, H. Li, and Q. Du, “Data augmentation for hyperspectral image classification with deep cnn,” *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 4, pp. 593–597, 2018.

[16] D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan *et al.*, “Identifying medical diagnoses and treatable diseases by image-based deep learning,” *Cell*, vol. 172, no. 5, pp. 1122–1131, 2018.

[17] J. P. Cohen, P. Morrison, L. Dao, K. Roth, T. Q. Duong, and M. Ghassemi, “Covid-19 image data collection: Prospective predictions are the future,” *arXiv preprint arXiv:2006.11988*, 2020.