# Multi-diseases Classification from Chest-X-ray: A Federated Deep Learning Approach

Sourasekhar Banerjee[1][0000−0002−3451−2851], Rajiv Misra[2], Mukesh Prasad[3], Erik Elmroth[1], and Monowar H. Bhuyan[1]

[1] Department of Computing Science, Umeå University, SE-901 87 Umeå, Sweden
`{sourasb, elmroth, monowar}@cs.umu.se`
[2] Dept. of Computer Science and Engg., Indian Institute of Technology Patna, India
`rajivm@iitp.ac.in`
[3] School of Computer Science, FEIT, University of Technology Sydney, Australia
`mukesh.prasad@uts.edu.au`

**Abstract.** Data plays a vital role in deep learning model training. In large-scale medical image analysis, data privacy and ownership make data gathering challenging in a centralized location. Hence, federated learning has been shown as successful in alleviating both problems for the last few years. In this work, we have proposed multi-diseases classification from chest-X-ray using Federated Deep Learning (FDL). The FDL approach detects pneumonia from chest-X-ray and also identify viral and bacterial pneumonia. Without submitting the chest-X-ray images to a central server, clients train the local models with limited private data at the edge server and send them to the central server for global aggregation. We have used four pre-trained models such as ResNet18, ResNet50, DenseNet121, and MobileNetV2 and applied transfer learning on them at each edge server. The learned models in the federated setting have compared with centrally trained deep learning models. It has been observed that the models trained using the ResNet18 in federated environment produce accuracy up to 98.3% for pneumonia detection and up to 87.3% accuracy for viral and bacterial pneumonia detection. We have compared the performance of adaptive learning rate based optimizers such as Adam and Adamax with Momentum based Stochastic Gradient Descent (SGD) and found out that Momentum SGD yields better results than others. Lastly, for visualization, we have used Class Activation Mapping (CAM) approaches such as Grad-CAM, Grad-CAM++, and Score-CAM to identify pneumonia affected regions in a chest-X-ray.

**Keywords:** Federated learning, Optimization, Transfer learning, Medical imagery analysis, Class activation mapping

## 1 Introduction

Deep learning creates a massive impact on the data-intensive applications in diverse domains from healthcare [10] to autonomous systems [4] to disaster management [11]. The availability of biomedical and healthcare-related data brings opportunities and challenges to the health care research [10]. In this domain, the application of Deep Neural Network (DNN) is required for the analysis of the Clinical Imaging [14] and Electronic Health Records [7].

In medical imagery analysis, applying deep learning methods can produce significant results but requires a considerable amount of quality data. An individual client may not have sufficient data to train and build a quality model, so, collaborating with different clients may solve the data insufficiency problems in deep learning but introduce privacy-constraints. Federated learning plays an essential role in solving such issues. Instead of storing highly sensitive patients personal and medical data to a centralized server, keep it at the individual client. Each client separately trains models based on locally available data and shares local models with a centralized server for global aggregation [17].

This paper proposes a federated deep learning approach for multi-diseases classification from chest-X-ray, specifically pneumonia. We have taken the following assumptions to alleviate this problem such as (a) The Client use their private data for training (b) Imbalanced distribution of data across edge servers, (c) The communication between clients/hospitals with the central server is stable, (d) The learning follows a cross-silo federated learning architecture where each edge server actively participates in training for every round. The main contributions are as follows.

- The training of the Convolutional Neural Network (CNN) on chest-X-ray images using transfer learning to classify different kind of pneumonia at the client-side in federated setting. The training includes the client-side local update and central server-side global aggregation.
- The impact analysis of different adaptive learning rate-based optimization methods, including Adam, Adamax, and momentum-based SGD, to train deep neural network model training in a federated environment.
- Visualization using CAM based methods (Grad-CAM, Grad-CAM++, and Score-CAM) to show the impact of the proposed model for detecting viral and bacterial pneumonia from chest-X-ray images.

The rest of the paper is organized as follows. In Section 2, a brief literature review is reported. The problem formulation is given in Section 3, whereas in Section 4, we describes the proposed model. Experimental results provided in section 5, and finally, Section 6 concludes the work.

## 2   Related Work

Deep learning has been played a significant role in medical imagery analysis. It was used in various areas such as brain-tumors detection [17], electronic health records analysis [3], pneumonia detection from chest-X-ray images [14, 2]. For these tasks, the goal was to develop a medical decision support system by employing data-driven machine learning-based modeling on patient data. A recent work [19] on Covid19 detection from chest-X-ray images using the deep convolutional neural network reported promising results. In [14], the model can detect pneumonia from chest-X-ray and produce the radiologist level outcome.

Training of deep neural networks requires a massive amount of data. Different clients work collaboratively to address this problem. However, putting sensitive information of patients in a centralized location may violate privacy constraints. Federated learning, introduced by google [9] as a replacement of traditional centralized

learning solutions can alleviate this problem. In the traditional learning method, different clients store data in a centralized server for training. However, in the federated learning, each client trains its model locally. Instead of sharing data, they only send the parameter updates or models to the centralized server. The server aggregates the parameter updates and sends the recent updates to the clients for further training [9]. By preserving privacy, federated learning is showing a great impact on the healthcare analysis domain [20]. In [17], the performance of federated semantic segmentation models on multimodal brain scans are similar to models trained by data sharing. Authors in [8] has prepared a model using federated learning based on Covid19 and other pneumonia chest-X-ray images. In [12], authors applied split learning for the collaborative neural network in health care.

In federated deep learning, optimization plays an important role. Federated averaging (FedAvg), a stochastic gradient descent (SGD) based optimization method [9], is massively used in federated learning. Now, momentum [13] accelerates SGD in the proper direction and dampens oscillations. Adaptive Moment Estimation (Adam) [6] computes adaptive learning rate for each training parameter. Adamax [6] is a variant of Adam optimizer. In adaptive federated optimization [15], the federated version of the adaptive optimizers such as Adagrad, Adam, and Yogi have been proposed.

Visual interpretation of the result is one of the important part of medical imagery analysis. Class Activation Mapping (CAM) based methods are useful here. Three types of visual explanation methods are very popular such as Grad-CAM [16], Grad-CAM++ [1], and Score-CAM [18]. One application of Grad-Cam has been found in versatile domains including healthcare [14] and disaster management [11].

## 3   Problem Formulation

This paper's main objective is to build an optimal federated deep learning model to detect multiple pneumonia diseases from normal and infected chest-X-ray images. So, the problem turns into an optimization problem with two parts, such as optimize loss function and optimal weights generation.

a. **Optimize loss function:** The training images are consist of two parts, such as batch of input images $X = [X_1, X_2, \ldots, X_N]$ and the label of the images $Y = [Y_1, Y_2, \ldots, Y_N]$ as desired output of the model. The loss function produces the error of the model. The objective here is to minimize the error on a set of data divided into batches. We have assumed that each hospital ($H_i$) has one edge server ($E_{ser\_i}$) and, its local dataset ($D_{E_{ser\_i}}$) to train local models. Edge servers individually train their model for $E$ epochs. The local loss function $F_{E_{ser\_i}}(w_i, X, Y)$ at $E_{ser\_i}$ is estimated using Equation (1).

$$F_{E_{ser\_i}}(w_i, X, Y) = \frac{\sum_{j=1}^{|D_{E_{ser\_i}}|} f_j(w_j, X_j, Y_j)}{|D_{E_{ser\_i}}|} \tag{1}$$

where loss per data point at $E_{ser\_i}$ is $f_j(w_j, X_j, Y_j)$ and updated weight is $w_i$. Here $w_j$ refers to the parameter update per iteration at $E_{ser\_i}$.

The global loss function on distributed datasets is computed based on the Equation (2).

$$F(w^*) = \min_{t=1,2,\dots,T} (\ \underset{i=1,\dots,N}{avg}\ F_{E_{ser\_i}}(w_i^t)) \tag{2}$$

where $T$ is the total number of rounds for federated training and $N$ is the total number of participating edge server.

In this paper, we attempt to solve this problem not only by binary classification (i.e., normal and pneumonia) but also with multi-class classification (i.e., normal, bacterial pneumonia, and viral pneumonia). For these two problems, we have used categorical cross-entropy [21] as loss function.

b. **Learning problem:** The learning problem is to minimize the loss function and converge towards a pre-defined threshold. Each edge server optimizes the loss function locally; then, the loss function will be globally optimized in the global update step. Each edge device eventually gets the best-optimized model ($w^*$) from the central server after $T$ rounds of iteration.

$$w^* = \underset{where\ t=1,2,\dots,T}{\arg\min}\ F(w^t) \tag{3}$$

Here, $w^t$ indicates model weights at $t^{th}$ round.

## 4   System Model

A typical Cross-silo federated learning involves the participation of all clients / hospitals and a central server to learn a model. Clients train models locally using their private data and upload the model to the central server. The central server orchestrates the training and generates global models.

Training of deep neural network models in federated learning needs a large amount of data and computational power at the client level. So, clients ( e.g., edge servers, micro data-center) with high computing and storage power may involve in training. The trained model can be deployed at devices (e.g., smartphone, laptop, tablet) registered with the central server. This paper assumes that each client contains one edge server, and each client participates in model training. Figure 1 describes the architecture of the proposed system.

1. **Training:**
   (a) Each client/hospital is an autonomous entity, and they are registered with the central server. Each hospital has its own set of chest-X-ray images and uses them for local training at the edge server.
   (b) At the initial phase of federated deep learning, each registered edge server downloads the pre-trained model and weights from the central server.
   (c) Each edge server then re-train the model using transfer learning according to their private chest-X-ray images and uploads the model updates to the central server.
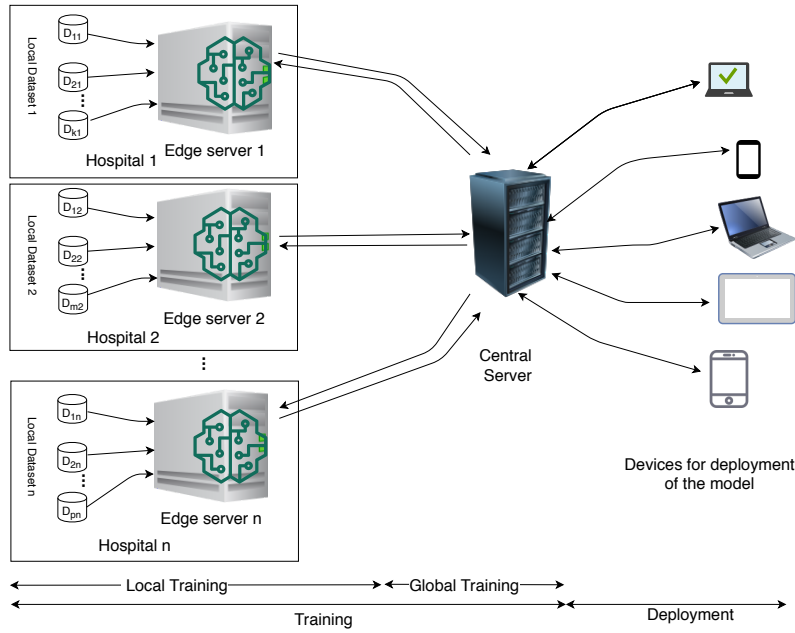   (d) The central server performs the global update using the federated averaging method.

Fig. 1: System architecture

(e) The edge server of each hospital download the globally updated model and initiate the next round of training.

(f) The above procedure will continue until the convergence of the model.

(g) CAM based methods (Grad-CAM, Grad-CAM++, and Score-CAM) would be applied to the optimal global model to detect pneumonia infected regions.

2. **Inference:**

(a) The optimal trained model would be inferred by the devices registered with the central server for real-time pneumonia prediction.

### 4.1 Proposed FDL approach

The proposed FDL approach involves Edge Server ($E_{ser}$) and Central Server ($C_{ser}$) for learning. Each edge server is responsible for learning locally, and the central server performs model aggregation. Algorithm 1 describes the local update at the $E_{ser}$ and Algorithm 2 refers the global update of the model.

1. **Client Side Local Update:** In Algorithm 1, each edge server ($E_{ser}$) parallelly trains their model on local data ($D_{E_{ser}}$) . The model computes loss for each epoch ($E_i$) and selects the best model with minimum validation loss. At local training, the algorithm uses any of the three optimizers such as Momentum with SGD, Adam and, Adamax. Momentum ($\beta$) accelerates the performance of SGD and calculates model weights using Equation (8) and (9) . Adam uses exponentially moving

averages on a mini-batch $B$ using Equation (4) and (5) to calculate the moments.

$$m^b = \beta_1 m^{b-1} + (1 - b_1)\nabla^b \tag{4}$$

$$v^b = \beta_2 m^{b-1} + (1 - b_2)[\nabla^2]^b \tag{5}$$

Then it calculates the bias regulated estimators using Equation (6) and (7).

$$\bar{m}^b = \frac{m^b}{(1 - b_1{}^b)} \tag{6}$$

$$\bar{v}^b = \frac{v^b}{(1 - b_2{}^b)} \tag{7}$$

Apply value of $\bar{m}^b$ and $\bar{v}^b$ in equation (10) to generate the parameter weights of the model. Adamax also use same approach but use the value $v$ as the $L2$ norm of the gradient ($\nabla$). Using Equations (11) and (12), it calculates model weights. After all epochs, the best model will be selected using Equations (13) and (14) and send the local model to the $C_{ser}$ for generating global model.

2. **Global Update at Central Server:** Algorithm 2 contains two parts: (a) Central Server-side Global Update, and (b) Optimal Global Model generation.

   (a) **Central Server side Global Update:** Each $E_{ser}$ train model parallelly and returns the local models to the $C_{ser}$ for global aggregation. $C_{ser}$ applies a federated averaging(FedAvg) method to update the global model. Each $E_{ser}$ downloads the global model for the next round of training.

   (b) **Finding Optimal Global Model:** For each round, $C_{ser}$ stores the global model, so after $T$ rounds, it will contain $T$ global models. The global model with minimum loss will be the optimal global model.

The CAM based techniques to be applied to the optimal model to visualize the pneumonia infected areas using heat maps. The optimal global model will be deployed at the edge for real-time processing and detection of diseases.

## 5  Evaluation

We divide the experiments into three parts. In the first part, we detect pneumonia from chest-X-ray using federated and centralized learning. In the second part, we have extended the experiments towards a different type of pneumonia detection (Bacterial and Viral) from chest-X-ray images using federated deep learning. In the final part, we will use CAM based techniques (e.g., Grad-CAM, Grad-CAM++, and Score-CAM) to detect pneumonia infected regions from chest-X-ray.

In Section 5.1, we have given a detailed description of the dataset. The simulation setup and the performance metrics have given in Section 5.2. In Section 5.3, we have shown corresponding results.

---

**Algorithm 1** Client Side Local Update

---

    **Input :** $D_{Local}$, $B$ = mini batch Size, $E$ = Total Epochs
    **Output :** Optimal model parameters $w_e$ , Optimal local loss function $F(w_e)$

1: **procedure** CLIENTSIDE\_LOCAL\_UPDATE($D_{local}$, $B$, $E$)
2:     **Initialize :** $w_e^0 = w_e^1$, $\lambda$, $\beta$, $\beta 1$, $\beta 2$
3:     **for** $e = 1; e \leq E; e+ = 1$ **do**
4:         **for** $b = 1; b \leq |\frac{D_{Local}}{B}|; b+ = 1$ **do**
5:             compute local update at each $E_{ser}$ using any of the following optimizers
6:             **Momentum + SGD :**

$$V^b = \beta V^{b-1} + (1-\beta)\nabla_{w_e^b} F_{E_{ser}}(w_e, X, Y) \tag{8}$$

$$w_e^b = w_e^{b-1} - \lambda V^b \tag{9}$$

7:             **Adam :** calculating bias regulated first and second moment using Equation 4, 5, 6, and 7 and then calculate

$$w_e^b = w_e^{b-1} - \lambda \frac{\bar{m}^b}{\sqrt{\bar{v}^b} + \epsilon} \tag{10}$$

8:             **Adamax :**

$$v^b = \max(\beta_2 v^{b-1}, |\nabla^b|) \tag{11}$$

$$w_e^b = w_e^{b-1} - \lambda \frac{\bar{m}^b}{v^b} \tag{12}$$

9:             compute loss using Equation 1
10:     select the local model using

$$F_{E_{ser}}(w_e) = \min_{e=1...E}(F_{E_{ser}}(w_e)) \tag{13}$$

$$w_e = \arg\min_{e=1...E}(F_{E_{ser}}(w_e)) \tag{14}$$

11:     **return** $w_e, F(w_e)$         ▷ The best local model to be returned

---

**Algorithm 2** Central Server Side Global Update

---

    **Input :** $E_{ser} = [E_{ser_1}, E_{ser_2} \dots E_{ser_N}]$, total rounds = $T$
    **Output :** final model parameters $w^*$ , Global optimal loss function $F(w^*)$

1: **procedure** CENTRALSERVER\_SIDE\_GLOBAL\_UPDATE($E_{ser}$)
2:     **for** $t = 1, 2, \dots T$ **do**
3:         **for** each $E_{ser_i} \in E_{ser}$ in parallel Compute: **do**
4:             ClientSide\_Local\_Update()
5:         $w^t$ = FedAvg(all clients local updates)  ▷ Calculate weights and loss using federated averaging
6:         Send $w^t$ to the $E_{ser}$s
7:     **Optimal Global Model**

$$w^* = \underset{where\ t=1,2,\dots T}{\arg\min} F(w^t) \tag{15}$$

8:     apply CAM based methods on optimal global model for visualization
9:     infer optimal global model for real-time processing

---

### 5.1   Datasets

We consider a public dataset of chest-X-ray [5] to perform experiments. The dataset contains 5,232 labeled images of chest-X-ray from children. It includes 3,883 pneumonia (including 2,530 Viral pneumonia and 1,353 bacterial pneumonia) and 1,349 normal chest-X-ray images. In the beginning, we only use this dataset to classify pneumonia and normal chest-X-ray images using federated deep learning. The dataset is divided into three hospitals/clients, so that each Hospital can train its model locally. In this experiment, the whole data is distributed into 30%, 32%, and 38% between Hospital 1, Hospital 2, and Hospital 3, respectively. Figure 2a describes the distribution of pneumonia and normal chest-X-ray, and Figure 2b depicts the distribution of viral pneumonia, bacterial pneumonia, and normal chest-X-rays among three hospitals.
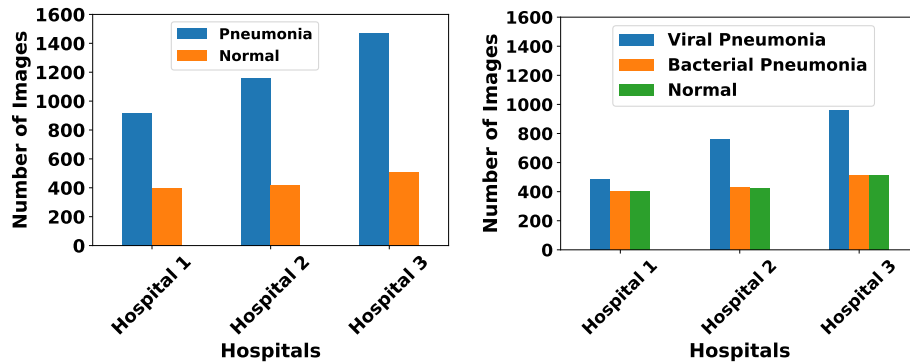


(a) Distribution of Pneumonia and Normal chest-X-ray images among hospitals

(b) Distribution of Viral Pneumonia, Bacterial Pneumonia and Normal chest-X-ray images among hospitals

Fig. 2: Characteristics of data

The dataset in each edge server further is divided into training, validation and test set with a ratio of $6:3:1$.

### 5.2   Simulation setup

We have taken four pre-trained models for the experiments including ResNet18, ResNet50, DenseNet121, and MobileNetV2. Each edge server first downloads the pre-trained model with pre-defined imagenet weights from the central server. The edge server in each hospital applies transfer learning here and trains the model with their local data. The edge server sends the updated model to the central server for model aggregation. This process will continue until the optimal model is produced. The description of the federated setting is described in Table 1.

Table 1: Parameters description in federated setting

| Parameter(s) | Value | Description |
|---|---|---|
| Clients | 3 | Edge servers for model training. |
| Client participation | 100% | Full participation of each client |
| Total epochs | 10 | Train model locally |
| Batch size | 16 | Batch size per round |
| Rounds | 5 | Number of interaction between hospitals (clients) and central server |
| Optimizer | 3 | Adam, Adamax, and SGD+Momentum |
| $\lambda$ | 0.001 | Learning rate for optimizer |
| $\beta$ | 0.9 | Momentum term |
| $\beta_1$ | 0.9 | The exponential decay rate for the first moment estimates |
| $\beta_2$ | 0.999 | The exponential decay rate for the second moment estimates |
| $\epsilon$ | $10E-8$ | A small number to prevent the division by zero |
| Momentum | 0.9 | |
| Learning rate decay | 0.1 | After 7 epochs learning rate will be decayed. |
| Performance metrics | 5 | Loss, Accuracy, F1-score, Precision, Recall |

### 5.3   Results and analysis

We have performed 5 consecutive rounds of model training of ResNet18, ResNet50, MobileNetV2, and DenseNet121 using transfer learning in federated domain using datasets of Figure 2a and Figure 2b, respectively and compared the loss convergence among them. Figure 3a describes the performance of every model on dataset Figure 2a, the ResNet18 model produces the loss 0.052 which is minimum compared to other methods. In Figure 3b, loss convergence of ResNet18 and ResNet50 model have carried out on dataset 2b. ResNet18 model gives better convergence than ResNet50 model. It produces the minimum loss 0.365 whereas ResNet50 model provide minimum loss 0.381.

Next, in Figure 3c, we have compared FedAvg method with centralized method. We have trained the ResNet18 model in both centralized and federated setting on the chest-X-ray dataset given in Figure 2a. The performance of the model in both environments is equivalent.

In Table 2, we made a comparative study between the performance of different optimizers in this problem. We compared adaptive learning rate based optimization techniques such as Adam and Adamax with momentum-based SGD technique in the federated setting. We test it for both datasets in Figure 2a and Figure 2b, respectively. In both cases, ResNet18 based model gives the best performance, and Momentum + SGD outperforms adaptive learning rate based methods.

To visualize pneumonia from chest-X-ray images, we have applied three different CAM based methods such as Grad-CAM, Grad-CAM++, and Score-CAM on the ResNet18 model. It produces the heatmap for the chest-X-ray images. Figure 4a shows the normal chest-X-ray. Figure 4b shows the the viral pneumonia affected regions and Figure 4c depicts the bacterial pneumonia infected regions.
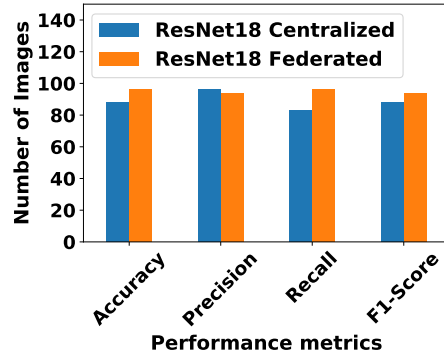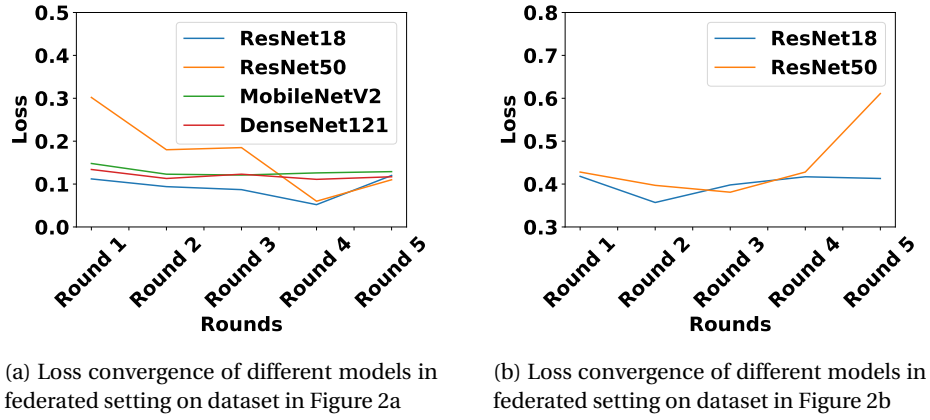
(a) Loss convergence of different models in federated setting on dataset in Figure 2a

(b) Loss convergence of different models in federated setting on dataset in Figure 2b



(c) Performance evaluation between centralized and federated deep learning models

Fig. 3: Performance evaluation between centralized and federated deep learning models



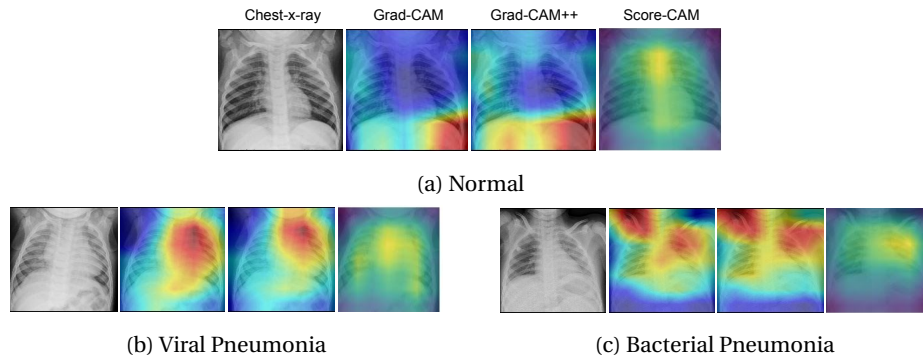(a) Normal

(b) Viral Pneumonia

(c) Bacterial Pneumonia

Fig. 4: Visualization of Pneumonia and Normal chest-X-ray

Table 2: Comparison among optimizers

| Classification of Pneumonia and Normal chest-X-ray | | | | | | |
|---|---|---|---|---|---|---|
| **Methods** | **Adam** | | **Adamax** | | **Momentum + SGD** | |
| | **Loss** | **Accuracy** | **Loss** | **Accuracy** | **Loss** | **Accuracy** |
| ResNet18 | 0.13 | 95.8% | 0.146 | 95.8% | 0.052 | 98.3% |
| ResNet50 | 0.12 | 95.6% | 0.126 | 95.6% | 0.06 | 97.6% |
| **Classification of Viral Pneumonia , bacterial Pneumonia, and Normal chest-X-ray** | | | | | | |
| ResNet18 | 0.486 | 81% | 0.517 | 79.4% | 0.365 | 87.3% |
| ResNet50 | 0.533 | 78.6% | 0.529 | 79.2% | 0.381 | 84.9% |

## 6   Conclusion and Future Work

This paper has shown a CNN-based deep learning model for efficiently detecting pneumonia from chest-X-ray images in a federated setting. The proposed approach is validated by employing four pre-trained models such as ResNet18, ResNet50, Dense-Net121, and, MobileNetV2 and re-trained using transfer learning on the chest-X-ray dataset in the federated as well as centralized environments. In both settings, the performance of ResNet18 based model with Momentum + SGD optimizer performs better than other models. ResNet18 achieves 98.3% accuracy for a pneumonia detection problem and 87.3% accuracy for classifying viral, bacterial pneumonia, and normal chest-X-ray images in federated settings. To visualization of pneumonia infected regions in a chest-X-ray, we used CAM based methods including Grad-CAM, Grad-CAM++, and Score-CAM on trained ResNet18 model.

In the future, we will extend this problem to classify different types of chest diseases efficiently from chest-X-ray images in a hybrid federated setting.

## Acknowledgement

## References

1. Chattopadhay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N.: Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 839–847. IEEE (2018)
2. Chouhan, V., Singh, S.K., Khamparia, A., Gupta, D., Tiwari, P., Moreira, C., Damaševičius, R., De Albuquerque, V.H.C.: A novel transfer learning based approach for pneumonia detection in chest x-ray images. Applied Sciences **10**(2),  559 (2020)

3. Farahani, B., Barzegari, M., Aliee, F.S.: Towards collaborative machine learning driven healthcare internet of things. In: Proceedings of the International Conference on Omni-Layer Intelligent Systems. p. 134–140. COINS '19, Association for Computing Machinery, New York, NY, USA (2019)
4. Grigorescu, S., Trasnea, B., Cocias, T., Macesanu, G.: A survey of deep learning techniques for autonomous driving. Journal of Field Robotics **37**(3), 362–386 (2020)
5. Kermany, D., Zhang, K., Goldbaum, M.: Labeled optical coherence tomography (oct) and chest x-ray images for classification. Mendeley data **2** (2018)
6. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
7. Lauritsen, S.M., Kalør, M.E., Kongsgaard, E.L., Lauritsen, K.M., Jørgensen, M.J., Lange, J., Thiesson, B.: Early detection of sepsis utilizing deep learning on electronic health record event sequences. Artificial Intelligence in Medicine p. 101820 (2020)
8. Liu, B., Yan, B., Zhou, Y., Yang, Y., Zhang, Y.: Experiments of federated learning for covid-19 chest x-ray images. arXiv preprint arXiv:2007.05592 (2020)
9. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics. pp. 1273–1282 (2017)
10. Miotto, R., Wang, F., Wang, S., Jiang, X., Dudley, J.T.: Deep learning for healthcare: review, opportunities and challenges. Briefings in bioinformatics **19**(6), 1236–1246 (2018)
11. Patel, Y.S., Banerjee, S., Misra, R., Das, S.K.: Low-latency energy-efficient cyber-physical disaster system using edge deep learning. In: Proceedings of the 21st International Conference on Distributed Computing and Networking. pp. 1–6 (2020)
12. Poirot, M.G., Vepakomma, P., Chang, K., Kalpathy-Cramer, J., Gupta, R., Raskar, R.: Split learning for collaborative deep learning in healthcare. arXiv preprint arXiv:1912.12115 (2019)
13. Qian, N.: On the momentum term in gradient descent learning algorithms. Neural Networks **12**(1), 145 – 151 (1999)
14. Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., et al.: Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. arXiv preprint arXiv:1711.05225 (2017)
15. Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., McMahan, H.B.: Adaptive federated optimization. arXiv preprint arXiv:2003.00295 (2020)
16. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. International Journal of Computer Vision **128**(2), 336–359 (Oct 2019)
17. Sheller, M.J., Reina, G.A., Edwards, B., Martin, J., Bakas, S.: Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. In: International MICCAI Brainlesion Workshop. pp. 92–104. Springer (2018)
18. Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z., Ding, S., Mardziel, P., Hu, X.: Score-cam: Score-weighted visual explanations for convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 24–25 (2020)
19. Wang, L., Wong, A.: Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. arXiv preprint arXiv:2003.09871 (2020)
20. Xu, J., Wang, F.: Federated learning for healthcare informatics. arXiv preprint arXiv:1911.06270 (2019)
21. Zhang, Z., Sabuncu, M.: Generalized cross entropy loss for training deep neural networks with noisy labels. In: Advances in neural information processing systems. pp. 8778–8788 (2018)