Federated Learning for Potato Leaf Disease Detection using CNN

Rakib Hossain Rifat
Dept. of CSE
BRAC University
Dhaka, Bangladesh
rakib.hossain.rifat@g.bracu.ac.bd
ID: 22366030

Marufa Kamal
Dept. of CSE
BRAC University
Dhaka, Bangladesh
marufa.kamal1@g.bracu.ac.bd
ID: 22366033

Abanti Chakraborty Shruti
Dept. of CSE
BRAC University
Dhaka, Bangladesh
abanti.chakraborty.shruti@g.bracu.ac.bd
ID: 22366034

Ehsanur Rahman Rhythm
Dept. of CSE
BRAC University
Dhaka, Bangladesh
ehsanur.rahman.rhythm@g.bracu.ac.bd

Humaion Kabir Mehedi
Dept. of CSE
BRAC University
Dhaka, Bangladesh
humaion.kabir.mehedi@g.bracu.ac.bd

Annajiat Alim Rasel
Dept. of CSE
BRAC University
Dhaka, Bangladesh
annajiat@gmail.com

***Abstract*—One of the most widely cultivated crops in the world is the potato. The spread of diseases such as potato leaf disease can significantly impact its quality and yield worldwide. Federated learning (FL) is a machine learning technique that enables various parties to collectively train a model without sharing individual data with one another. In the context of potato farming, our research proposes federated learning (FL) to detect potato leaf disease across two clients without sharing sensitive data between them. Each client is trained with the CNN model and its different architectures using their own dataset in this approach. Then the models are aggregated to create a global model and this global model is then used to detect potato leaf disease across multiple farms, improving the accuracy of disease detection and enabling early intervention to limit its spread. For every round of the updated global model, the accuracy of our model has improved significantly. After three rounds of communication using the Inception-V3 model on the server, the accuracy and F1-score were 88% and the precision and recall were around 89% respectively. These experiments were conducted using different custom and pre-trained CNN architectures to understand the improvement in results after collaborating with different datasets.**

***Index Terms*—Federated Learning, Potato Disease, Agriculture, Early Blight, Late Blight, CNN, Leaf Disease, InceptionV3**

I. Introduction

Improvements in nutrition and food security are two of the main issues that the agricultural sector must address. Leaf diseases can have a significant impact on plants' growth, production, and general health. The severity and effect of the disease can vary depending on the type of plant and the specific disease [1]. Potatoes are the third-most significant food crop in the world for human consumption, behind rice and wheat. 300 million metric tons of agricultural products are produced globally, and over one billion people eat potatoes. [2]. Several varieties of potatoes are grown in various African and Asian nations, such as Egypt, Pakistan, Bangladesh, and others, depending on the location and the local market [3]. Overall, potato production is quite popular. To ensure the proper growth of potatoes during the growing period, it is crucial to be able to detect any diseases that may occur. Late identification of diseases is one of the prime reasons for damaged crops.

Despite the advancement of technology, people are still dependent on traditional methods of disease detection, which often rely on visual inspection by experts and can be time-consuming and expensive. This research focuses on the two most harmful foliar diseases for potato crops: **late blight** (*Phytophthora infestans*) and **early blight** (*Alternaria solani*), which significantly reduce yields in the majority of potato-growing regions worldwide [4]. As mentioned earlier, potatoes are grown in different regions of the world with differences in soil, water, and other factors during their production. Introducing a technique that allows researchers to collect data from multiple production regions and train models on this collaborative data could result in more generalized disease identification that can be applied anywhere. A machine learning technique named federated learning (FL) enables many participants to jointly train a model without disclosing their individual data. [5]. This approach is particularly relevant in agriculture, where data privacy and ownership are of utmost importance to farmers. Farmers are often reluctant to share information about the soil, weather, crop yield, and other agricultural practices as it can be valuable to competitors or other parties who have sensitive or confidential knowledge. Thus, using federated learning allows multiple clients to maintain the confidentiality of their crops while creating a more efficient and accurate system that can provide early detection of diseases, which can help reduce yield losses and economic damage. By leveraging the collective knowledge of multiple farmers, robust and more accurate models can be used to monitor and manage disease outbreaks in real-time.

Our proposed research method is based on using custom

CNN architecture and the transfer learning approach to find early blight, late blight, and healthy potato leaves. Transfer learning has also been used to train and test CNN models that have already been trained. This has been done on both the client and server sides. Training has been done using data collected from two different datasets of two regions: Bangladesh and Pakistan referred to as client one and client two. And later on, these trained models were tested on the global server, preserving the privacy of the data using a federated learning approach. The architectural layers allow us to study effective features for distinguishing both diseases without any exchange of information between the clients.

II. Related Works

Over time, agriculture and technology have converged, and numerous studies have been proposed to identify and prevent crop diseases. As previously mentioned, potatoes are a significant crop for food production, and several studies have been conducted on the recognition and prevention of potato leaf diseases. Here are some of the key findings.

In the year 2020, Tiwari et al. [6] proposed to use transfer learning models to extract features and then use multiple classifiers to classify the three leaf diseases as Early blight, Late blight, and healthy. For feature extraction, VGG19, VGG16, and InceptionV3 were used. As classifiers, they used models like SVM, KNN, neural networks, and logistic regression. Among all the models, they achieved the best results with the VGG19+Logistic Regression model, which was 97.8% accuracy. Another work in the same year by Iqbal et al. [7] showed that potato disease classifications can also be done using a combination of image processing and machine learning techniques. First, image segmentation was used then global feature descriptors were used for feature extraction like colors, texture, and shape. Then multiple machine learning models like SVM, K-NN, LDA, Random Forest, Decision Tree, Naive Bayes, etc. were used to classify the data. The Random Forest classifier showed the best performance among all of them. Agarwal and his team proposed another CNN model with four convolutional models in the year 2020 [8]. Their experimental findings demonstrate that the proposed model works effectively under a variety of scenarios, including shifting backgrounds, varying image sizes, spatial differentiation, frequent level changes in lighting, and real pictures. In 2021, two primary convolutional layers for feature extraction with various convolution window sizes are included in the proposed architecture's 14 layers, which are followed by two fully connected layers for classification [9]. The authors have also used data augmentation for data increment. They have achieved an accuracy of 98% with their proposed architecture. Sholihati et al. [10] in the year 2020 presented a model using deep learning to identify four diseases of potato leaves. They also used data augmentation in their method, as they acquired a total of 5,100 data. As classifiers, the authors used the pre-trained VGG16 and VGG19 models. After the 250th epoch, their model achieved an accuracy of 91%. To detect early blight disease from the potato leaves, research was done in 2021 by Afzaal et al. [11]. The authors experimented with a machine vision model with a combination of deep learning models. GoogleNet, VGGNet, and EfficientNet were used for the experiments, where EfficientNet with four class CNN showed better performance. Another CNN-based model was proposed, and the authors came up with a new sequential model to identify various diseases [12]. They experimented with AlexNet, VGGNet, ResNet, and LeNet and their proposed sequential model. They achieved a precision of 97%. However, the size of their dataset was not very large. Also in 2019, Suttapakti et al. presented research to identify potato diseases by leveraging color and texture features [13]. K-means clustering and the maximum-minimum color difference method were used for image segmentation and feature extraction. They experimented with 300 potato leaf images and achieved satisfying performance.

Although many studies have been done on potato leaf disease detection, none of them were based on a federated learning approach. We want to start a new research subject on federated learning because it can protect data privacy in the agriculture industry. Moreover, our model deals with much more data than others.

III. Methodology

In this paper, convolutional neural network architecture is used to train our models for both the clients and the global server as shown in figure 1. We have also used the Federated Learning Approach to aggregate the locally trained weights into the global server model.

A. Data Augmentation

All of the images in both client one and client two are re-scaled to keep them in the range of 0 and 1. We randomly rotated the images to 20 degrees, zoomed with 0.05 percentage, shifted the width and the height by 0.05 and horizontally flipped the images, and augmented both datasets in order to increase the amount of training data in the dataset with randomly changed updates. We also shuffle the data, and we also use class_weights to avoid overfitting and generalizing the model.

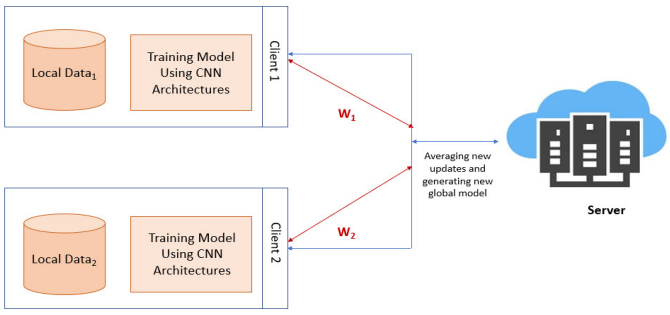

Fig. 1. Overall Framework of the Proposed Methodology

B. Client-Side Model

For our client side, we have used pre-trained CNN architectures to train the local clients and generate the model for the global server.

1) Custom CNN Model: Convolutional neural networks are made to take advantage of the fact that images are 2D, which lets them learn hierarchical representations of image features through convolutional layers. Using a pre-trained model built using transfer learning with CNN helps achieve better performance with less training. The CNN architecture proposed in this research consists of three convolution layers as shown in figure 2 with max pooling, flattening the images, and two dense layers with the ReLU activation function and softmax function, converting the numbers to probabilities. This architecture and set of hyperparameters are used to train both the client and the model.

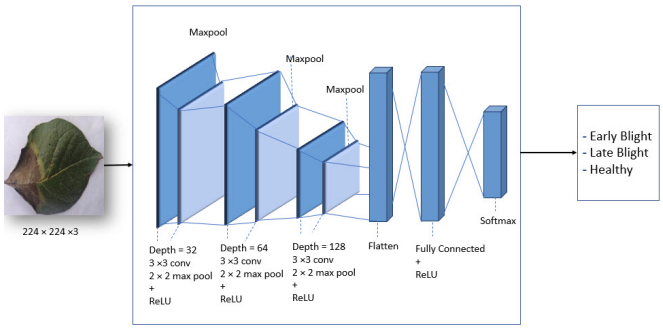

Fig. 2. The Custom CNN Architecture used in Client Side and Server Side Model

2) VGG19: VGG19 is broadly used in image recognition and classification tasks, such as object detection and scene recognition, and has achieved state-of-the-art performance on several benchmark datasets [14]. It was first proposed in 2014 [15] being trained on about 1.2 million images. We have also experimented with this pre-trained model to train our client datasets. The network consists of 19 convolutional layers, followed by max-pooling layers and fully connected layers, and uses small 3x3 filters to extract features from the input images. This model consists of some common hyperparameters, like a learning rate of 0.0001, using the ReLU activation function, a dropout rate of 0.2, and three dense layers of the Softmax function to generate the probabilities and the Adam optimizer.

3) VGG16: We have also experimented with VGG16 created by the Visual Geometry Group (VGG) at the University of Oxford [15], a convolutional neural network (CNN) architecture consisting of 16 layers, including 13 convolutional layers and 3 fully connected layers. Similar to VGG19, it attained state-of-the-art performance on the ImageNet image classification task and employs a 3x3 filter for each convolutional layer and a 2x2 max pooling window with a stride of 2. In contrast to VGG19, VGG16 has fewer layers, and thus fewer parameters, resulting in a relatively quicker training and application process.

4) InceptionV3: Another pre-trained model developed by Google [16] used in this research is InceptionV3, a convolutional neural network (CNN) architecture that utilizes "inception modules" to capture features at various scales by utilizing multiple filter sizes within the same layer. InceptionV3 obtained state-of-the-art accuracy With 48 layers on the ImageNet image classification task and includes advancements over previous Inception architecture versions, such as factorized 7x7 convolutions and batch normalization. Its combination of accuracy and efficiency has made it a prevalent option for numerous computer vision applications.

5) EfficientNet B1: EfficientNet [17] is a pure ConvNet that is trained on ImageNet-1k 240*240 resolution and is a technique that equally scales all dimensions of the network with a simple and effective coefficient. The model achieved state-of-the-art performance on the CIFAR-100. Depth-wise separable convolution is used in this model which uses fewer parameters and is a more coherent computation that justifies its name EfficientNet.

C. Server-Side Model

Step 1: In the beginning the global model has initial weights and the training starts on the client side.

Step 2: Both the clients perform training on their local dataset. Clients aim to improve the model by reducing the categorical cross-entropy loss, which involves classifying the data into categories. After training and receiving the best model, the client side sends these model weights to the server.

Step 3: Global model parameters are then updated once the server gets updates from both clients and computes an average weight and generates a new model in accordance with the following equations :

$$F_x(w) = \frac{1}{n_x} \sum_{i \in P_k} f_i(w) \tag{1}$$

Using equation 2,

$$f(w) = \sum_{x=1}^{X} \frac{n_x}{n} F_x(w) \tag{2}$$

Based on the scaling factor retrieved from each dataset the local model weights are updated and then all the client weights are summed and passed to the global model. Similar to the client models, we have used Custom CNN, VGG16, VGG19, and InceptionV3 for the global model. This process gets repeated for 4 rounds in total. Having no weights initially in the global model, rather updating the weights after each round.

IV. Dataset

This research has been executed using two different datasets considered as clients. Both of these datasets contain three classes of images, namely early blight, late blight, and healthy leaf, as depicted in figure 3. The first dataset is collected from crops in the Bangladesh region [18], [19]. It contains images for different crop diseases having 14 class labels and we have specifically used the images for the 3 classes mentioned

above. The second dataset contains images collected from potato production in Pakistan [20]. The following table I demonstrates the data distribution of client one and two.

TABLE I
DATA DISTRIBUTION OF TRAINING DATASET

Dataset	Class Label			Total
	Early Blight	Late Blight	Healthy	
Client One [18]	1000	1000	152	2152
Client Two [20]	1628	1424	1020	4072

Fig. 3. Example of Images in the dataset containing 3 class labels

From both the client datasets, using random splitting 10% of data is stored for testing the global server, and among the rest of 90%, 10% is used for validation and the remaining 80% of data is used for training the model.

V. EXPERIMENTAL RESULTS AND DISCUSSION

A. Experimental Setup

We have used Google Colab free GPU and Python 3 as the language. Pandas, NumPy, sklearn, and TensorFlow keras libraries are used in this experiment. For deep learning models, using optimized hyperparameters is crucial. For training our client-side models we have used a batch size of 8 and trained the model for 20 epochs. The learning rate adjusted with each epoch which was finalized at 0.0001. For the dense layer, ReLU is used as the activation function and a softmax activation function is used at the output layer.

B. Results

We used a total of four rounds (0, 1, 2, 3) of the client training phase in our experiment and achieved the final results in the fourth round (round 3). The result section discusses the experimental results of all experimental models individual results and tries to demonstrate or draw a comparison between them.

1) Custom CNN Model: As mentioned earlier, custom designed CNN model is used for our experiment. The results of the models are discussed here.

From figure 4, we can see the training loss of the client models. For round 0, we observe that the training loss was very high **0.87** for **client 1** and **0.97** for **client 2**, and with each epoch, the loss came to 0.11 and 0.13 for client 1 and 2 respectively. In round 3 the loss was minimized to a great extent and at the final 20th epoch the loss became nearly 0.03 and 0.05 for client 1 and client 2 respectively.

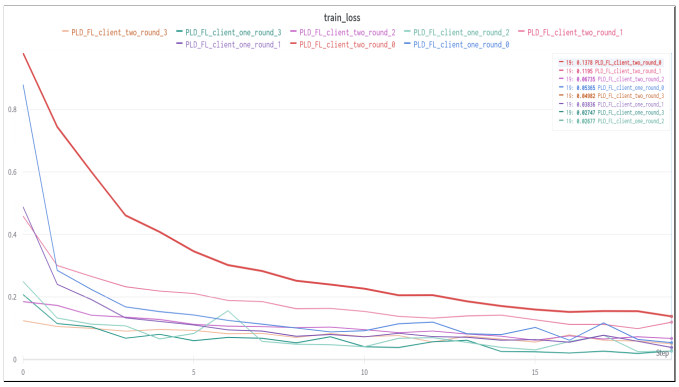

Fig. 4. Training Loss on Custom CNN Model

TABLE II
TEST ACCURACY ON CUSTOM CNN MODEL

Communication Round	Test Accuracy		
	Client1	Client2	Server
0	99.485	96.721	52.564
1	97.938	95.628	78.045
2	96.392	96.175	88.462
3	97.938	97.814	87.5

From Table II, we can see the test results on our client and server models for each round of communication. On the first round of communication although the accuracy of the client side is high (99% and 96% respectively) the server side accuracy is very low 52% only. The accuracy increased along with the round of communications. In the third round, we can see that the server accuracy reached the highest which is 88.46% but upon training for another round of communication, it is seen that there is a decreasing trend of accuracy (87.5%). Therefore, we stopped our training at this point.

2) Pretrained Transfer Learning Models: In our study, we have used four pre-trained models - VGG16, VGG19, InceptionV3, and EfficientNet B1. We will be discussing the results of the individual model in this subsection.

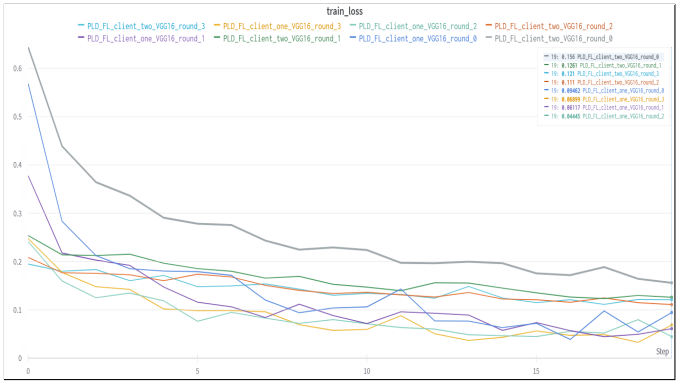

Fig. 5. Training Loss on VGG16 Model

We can see the training losses of VGG16, VGG19, Incep-

Fig. 6. Training Loss on VGG19 Model

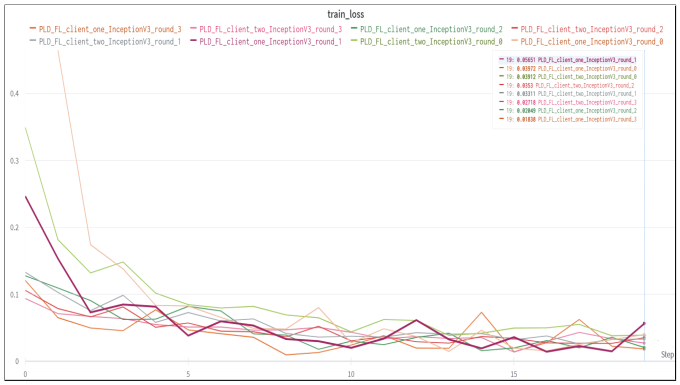

Fig. 7. Training Loss on Inception Model

tionV3, and EfficientNet models in the figure 5, 6 , 7 and 8 respectively. The graphs show us that the InceptionV3 model gets the lowest training loss of 0.056 after the 20th epoch followed by 0.159 loss of the VGG16 model. VGG19 has a higher loss score of 0.195 while EfficientNet has the highest training loss of 1.1.

TABLE III
TEST RESULTS ON PRE-TRAINED MODELS

Model	CommR	Test Accuracy		
		Client1	Client2	Server
VGG16	0	97.938	93.989	66.987
	1	97.938	94.262	76.763
	2	98.454	96.175	78.205
	3	96.907	97.541	73.718
VGG19	0	96.907	89.617	65.385
	1	97.423	90.143	68.429
	2	99.485	95.902	71.154
	3	95.902	98.454	72.115
InceptionV3	0	98.969	98.634	75.16
	1	97.938	98.307	83.013
	2	98.969	98.696	88.642
	3	98.969	98.907	**88.141**
EfficientNet B1	0	44.845	40.437	41.667
	1	53.093	38.7798	41.667
	2	53.608	40.984	41.667
	3	47.938	41.530	41.667

Table III shows the test accuracy of the four used pre-trained models and the accuracy of their details on the client

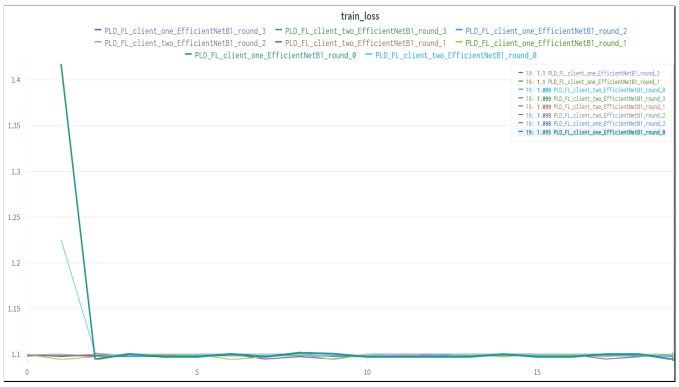

Fig. 8. Training Loss on EfficientNet Model

and server sides. From the table, it is evident that Inception-V3 outperforms the VGG16, VGG19, and EfficientNet models from the first round of communication. Over the communication round the accuracy is increased on all of the models showing the impact of federated learning. VGG16 has a better performance than VGG19 in this experiment. EfficientNet however performs very poorly from the very first communication round and does not increase much with the rounds. It achieves the lowest accuracy of nearly 42% on the last round.

3) Comparison between Different Models: In this section, we compare our four experimented models and discuss their results in terms of evaluation matrices.

TABLE IV
TEST RESULTS ON DIFFERENT MODELS

Model	Accuracy	Precision	Recall	F1-Score
Custom CNN	0.875	0.88	0.86	0.87
VGG16	0.731	0.73	0.74	0.73
VGG19	0.721	0.71	0.73	0.71
InceptionV3	**0.881**	**0.89**	**0.89**	**0.88**
EfficientNet B1	0.42	0.14	0.33	0.20

From the table IV we can see the test results of different models. Among the four models, we can see Inception-V3 and our proposed custom CNN model shows the best performance which is nearly 88% accuracy. The F1 score of the Inception-v3 model is the highest which is 0.88. Even the F1-score of our proposed model is satisfactory which is 87%. The VGG16 model slightly performed better in comparison to the VGG19 model with a 73.718% of accuracy whereas, the VGG19 model achieved an accuracy of 72.115%. The F1 score difference is also seen here where VGG16 has a score of 0.73 compared to the 0.71 scores of VGG19. In terms of all the evaluation matrices EfficientNet performs worst with only 42% of accuracy and only 14% of precision which is very dissatisfactory.

From the figure 9, we can see that 250 data are correctly classified as Early Blight and 190 data are correct as Late Blight. Some of the late blights are wrongly classified into

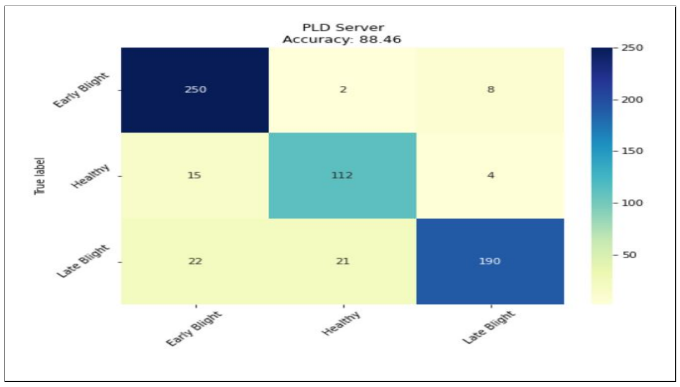

Fig. 9. Confusion Matrix of Custom CNN Model

early blight and healthy classes but most of the early blights are correctly classified. In the healthy class, some of the healthy class is misclassified as early blight. Although most amounts of the healthy classes (112) are correctly classified.

As the confusion matrices of the pre-trained models are shown in figure 10, 11, 12 and 13, we can see that Inception model and VGG19 model correctly identifies the nearly same amount of Early Blight disease and the amount is 214 and 217 respectively.

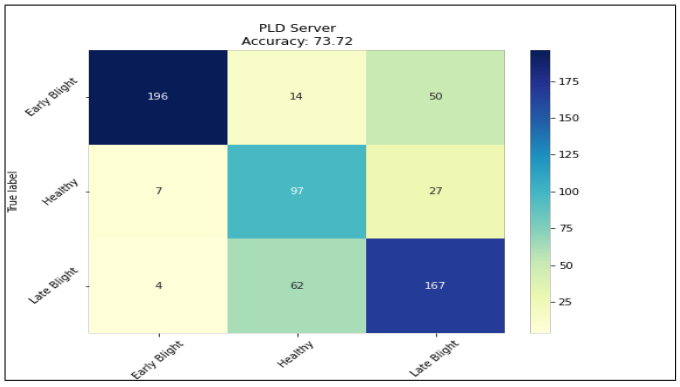

Fig. 10. Confusion Matrix of VGG16 Model

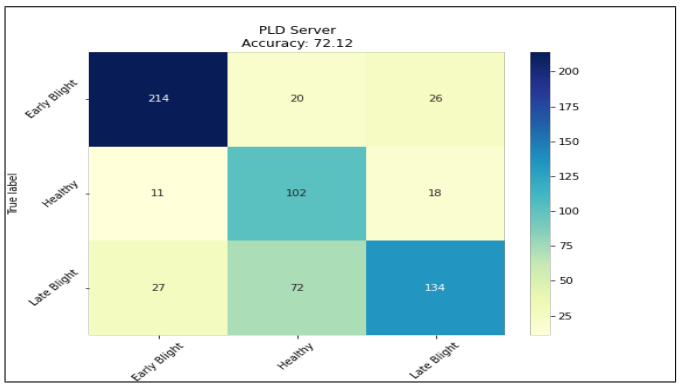

Fig. 11. Confusion Matrix of VGG19 Model

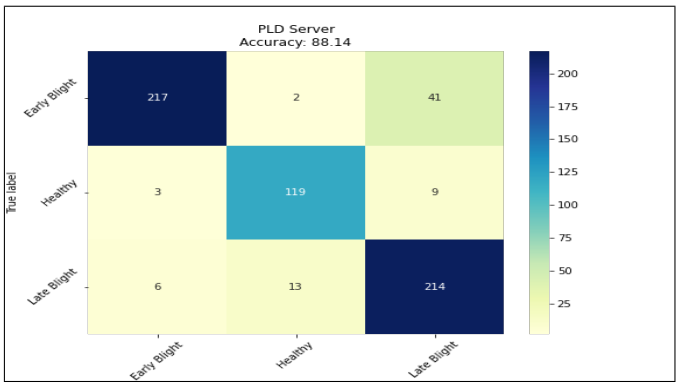

Fig. 12. Confusion Matrix of Inception-v3 Model

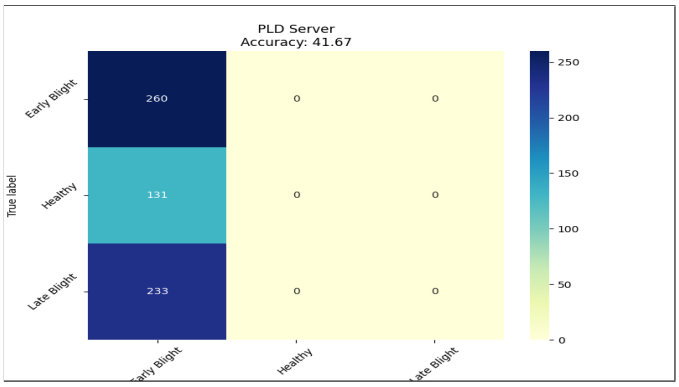

Fig. 13. Confusion Matrix of EfficientNet B1 Model

The Late Blight is mostly correctly classified by the Inception-v3 model while VGG19 misclassified 99 of them. All of the models misclassify some healthy classes to the late blight class mostly. In figure 13, we the EfficientNet confusion matrix and how the model classifies all of the data into the Early Blight class. It explains the reason for the poor performance of the model.

TABLE V
COMPUTATION RESULTS ON DIFFERENT MODELS

Model	Memory Usage(%)	Computation Time(min)
Custom CNN	9.58	177.583
VGG16	10.25	160.067
VGG19	14.54	153.666
Inception-V3	26.76	198.3
EfficientNet B1	33.28	223.75

In Table V, the computational difference between the four models is shown. From the table, it is evident that even if the Inception-V3 model is the most accurate but it takes the most time for computation after EfficientNet. On the other hand, our custom CNN model takes less time 177.6 min approximately compared to the Inception-V3 model but shows nearly the same accuracy. In terms of memory usage too, our proposed custom model outstands the InceptionV3 model by using only 9.58% of memory compared to the 26.76% usage of the Inception model. Most memory(33.28%) is again

used by the worst-performing model - EfficientNet. However, there's not much difference seen between the VGG16 and VGG19 model's memory usage and computation time.

VI. Conclusion and Future Work

A federated learning-based framework is presented in this research to identify the two most familiar diseases of potato leaves early blight and late blight disease, during their production and growth. This approach helps to maintain data privacy which is usually a concerning point in the agricultural sector. We have used the CNN model and its different architectures for both the client and server sides. Through extensive testing of these models on the two datasets we have received the best accuracy through Inception-V3 architecture and our proposed custom CNN architecture. Furthermore, we demonstrate the advantages of collaborating on multiple agricultural crops using the federated learning framework in the context of potato leaf analysis. In the future we hope to add more clients as the increase in the dataset with the incorporation of more clients, will demonstrate more potential to achieve better accuracy and generalization in identifying crop diseases. We also hope to add multimodal data such as climate and soil data, to provide a more comprehensive understanding of crop health.

References

[1] Z. Iqbal, M. A. Khan, M. Sharif, J. H. Shah, M. H. ur Rehman, and K. Javed, "An automated detection and classification of citrus plant diseases using image processing techniques: A review," *Computers and electronics in agriculture*, vol. 153, pp. 12–32, 2018.

[2] "Potato facts and figures," https://cipotato.org/potato/potato-facts-and-figures/, accessed: February 23, 2023.

[3] "Potato — diseases and pests, description, uses, propagation," https://plantvillage.psu.edu/topics/potato/infos, accessed: February 23, 2023.

[4] C. Hou, J. Zhuang, Y. Tang, Y. He, A. Miao, H. Huang, and S. Luo, "Recognition of early blight and late blight diseases on potato leaves based on graph cut segmentation," *Journal of Agriculture and Food Research*, vol. 5, p. 100154, 2021.

[5] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, and Y. Zhou, "A hybrid approach to privacy-preserving federated learning," in *Proceedings of the 12th ACM workshop on artificial intelligence and security*, 2019, pp. 1–11.

[6] D. Tiwari, M. Ashish, N. Gangwar, A. Sharma, S. Patel, and S. Bhardwaj, "Potato leaf diseases detection using deep learning," in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, 2020, pp. 461–466.

[7] M. A. Iqbal and K. H. Talukder, "Detection of potato disease using image segmentation and machine learning," in *2020 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*. IEEE, 2020, pp. 43–47.

[8] M. Agarwal, A. Sinha, S. K. Gupta, D. Mishra, and R. Mishra, "Potato crop disease classification using convolutional neural network," in *Smart Systems and IoT: Innovations in Computing: Proceeding of SSIC 2019*. Springer, 2020, pp. 391–400.

[9] N. E. M. Khalifa, M. H. N. Taha, L. M. Abou El-Maged, and A. E. Hassanien, "Artificial intelligence in potato leaf disease classification: a deep learning approach," *Machine Learning and Big Data Analytics Paradigms: Analysis, Applications and Challenges*, pp. 63–79, 2021.

[10] R. A. Sholihati, I. A. Sulistijono, A. Risnumawan, and E. Kusumawati, "Potato leaf disease classification using deep learning approach," in *2020 international electronics symposium (IES)*. IEEE, 2020, pp. 392–397.

[11] H. Afzaal, A. A. Farooque, A. W. Schumann, N. Hussain, A. McKenzie-Gopsill, T. Esau, F. Abbas, and B. Acharya, "Detection of a potato disease (early blight) using artificial intelligence," *Remote Sensing*, vol. 13, no. 3, p. 411, 2021.

[12] M. K. R. Asif, M. A. Rahman, and M. H. Hena, "Cnn based disease detection approach on potato leaves," in *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*. IEEE, 2020, pp. 428–432.

[13] U. Suttapakti and A. Bunpeng, "Potato leaf disease classification based on distinct color and texture feature extraction," in *2019 19th International Symposium on Communications and Information Technologies (ISCIT)*. IEEE, 2019, pp. 82–85.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[17] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *ArXiv*, vol. abs/1905.11946, 2019.

[18] D. Hughes, M. Salathé *et al.*, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *arXiv preprint arXiv:1511.08060*, 2015.

[19] "New bangladeshi crop disease," https://www.kaggle.com/datasets/nafishamoin/new-bangladeshi-crop-disease.

[20] J. Rashid, I. Khan, G. Ali, S. H. Almotiri, M. A. AlGhamdi, and K. Masood, "Multi-level deep learning model for potato leaf disease recognition," *Electronics*, vol. 10, no. 17, p. 2064, 2021.