[1]Jyoti L. Bangare

[2]Nilesh P Sable

[3]Parikshit N. Mahalle

[4]Gitanjali Rahul Shinde

Federated Texture Classification: Implementing Colorectal Histology Image Analysis using Federated Learning

JES

Journal of Electrical Systems

Abstract: - This research explores neural network models' performance and adaptability in the context of the colorectal histology dataset as it pertains to the categorization of textures. Inception, VGG19, and MobileNet, together with their federated variations, are among the models being examined. The study includes a detailed evaluation, parameter analysis, and training information. VGG19 stands out as a particularly noteworthy high performance, with remarkable accuracy, precision, and recall. Due to its lightweight design, MobileNet performs less well, but its potential is enhanced by the addition of federated learning. The accuracy and precision of federated versions of Inception, VGG19, MobileNet, and a Lightweight MobileNet model are competitive, with FL-Lightweight MobileNet achieving outstanding results. The work has important ramifications for the field of medical image analysis since it shows how federated learning may balance the need for data confidentiality and privacy with model performance. This study marks a turning point in the development of medical imaging by opening the door to in-depth investigation into the complex interactions across federated paradigms. Furthermore, these results provide a compelling story in the wider discussion of how cutting-edge technologies and the pressing needs of contemporary healthcare might work together.

Keywords: Neural networks, Texture classification, Colorectal histology, Inception, VGG19, MobileNet, Federated learning, Privacy-preserving, Medical image analysis, Accuracy, Precision, Recall, Model performance, Data security, Distributed learning, Healthcare.

I. INTRODUCTION

With the use of medical image analysis, precise illness diagnosis, treatment planning, and disease monitoring are now possible. Histological pictures are one of the many forms of medical images, and they are extremely important for comprehending microscopic disease development, tissue composition, and cellular architecture. In particular, colorectal histology photos give insights into colorectal illnesses, assisting pathologists in spotting anomalies and selecting the best course of action for patients [1]. Advanced computational approaches that can manage these pictures' complexity and variety while protecting patient privacy and data security are necessary for efficient image analysis, nevertheless [2].

Deep learning algorithms in particular have demonstrated outstanding accomplishments in the processing of medical images in recent years. The subfield of image analysis known as texture classification offers great promise for describing the intricate spatial patterns found in histology pictures [3]. Diagnosis can be greatly aided by patterns seen in cellular configurations, glandular structures, and fibrous components. Manual feature extraction and subsequent classifier training are steps in the traditional texture categorization process [4]. These methods, however, frequently fall short in their capacity to completely and effectively capture complex patterns.

The understanding of the value of privacy and data ownership grows as the area of medical image analysis develops [5]. As sensitive and private information, medical data sharing for research purposes involves issues with patient confidentiality, data security, and adherence to laws like the HIPAA Act. Federated learning, a cutting-edge strategy, offers a resolution to this conundrum by enabling several organisations or businesses to jointly create a global model without disclosing raw data. Decentralised model training, in which each institution trains the model locally using its own data and only exchanges model updates, achieves this [6][7]. In this manner, the data stays inside the confines of the different institutions, all while resolving privacy issues and enabling the development of effective models.

[1]Research Scholar, Bansilal Ramnath Agarwal Charitable Trust's, Vishwakarma Institute of Information Technology, Pune, Maharashtra, India. Email: jyoti.bangare@cumminscollege.in

[2]Bansilal Ramnath Agarwal Charitable Trust's, Vishwakarma Institute of Information Technology, Pune, Maharashtra, India. Email: drsablenilesh@gmail.com

[3]Bansilal Ramnath Agarwal Charitable Trust's, Vishwakarma Institute of Information Technology, Pune, Maharashtra, India. Email: aalborg.pnm@gmail.com

[4]Bansilal Ramnath Agarwal Charitable Trust's, Vishwakarma Institute of Information Technology, Pune, Maharashtra, India. Email: gr83gita@gmail.com

This study's main goal is to investigate how federated learning may be used to the texture categorization of colorectal histology pictures [8]. We intend to create and construct a strong model that can successfully categorise textures in histology pictures while upholding data privacy and security by utilising federated learning. We postulate that federated learning's collaborative features can enhance texture classification performance while preserving patient data's privacy [9][10].

The format of the research paper is as follows: We examine relevant research in the areas of medical image analysis, texture classification, and federated learning in Section 2 of this article. This review sets the context for our study by highlighting the current difficulties and knowledge gaps. The research approach used in our study is described in Section 3. We examine the foundations of federated learning and go into great depth on the design of our suggested framework. We also go through model selection, preprocessing procedures, and texture feature extraction methods used on colorectal histology pictures. In Section 4, the dataset utilised for this study is introduced. We provide light on the colorectal histology pictures' origin, size, and class distribution. We also go through any methods of data augmentation used to improve the generalizability of our models. The presentation of our tests and findings is covered in Section 5. We outline the experimental setting, including how data was distributed among the many participating organisations or devices. The federated training procedure is explained in great detail. Along with visualisations of the learning curves and model convergence behaviours, the quantitative findings are provided. These results include accuracy, precision, recall, and other pertinent measures. We have a full explanation of the findings in Section 6. We analyse the results in light of federated learning and medical image analysis. It is done a performance comparison between our federated models and conventional centralised methods. In addition, we discuss any implementation-related difficulties and provide potential solutions for future research.

II.   RELATED WORK

Through the contributions of various research publications, the field of medical image analysis, texture classification, and federated learning has greatly changed. The basis for our study on "Federated Texture Classification: Implementing Colorectal Histology Image Analysis using Federated Learning" is laid out in this part, which delves into the body of literature that already exists in these fields. Deep learning algorithms have led to a paradigm change in medical image analysis during the past ten years. By incorporating architectures made for medical pictures, studies like "Deep Residual Learning for Image Recognition" by He et al. (2015) and "U-Net: Convolutional Networks for Biomedical Image Segmentation" by Ronneberger et al. (2015) revolutionised the area. These structures have been modified for histology pictures and are useful for cell segmentation and cancer identification. The "DeepFocus" network suggested by Bayramoglu et al. (2016) for tissue segmentation and "HistoNet" by Saltz et al. (2018) for cancer diagnosis are two examples of histopathology-specific CNN models that have developed. The use of deep learning in histological analysis has been made possible thanks to these models. The rich textures exhibited in these photographs continue to be difficult to capture, though.

Texture analysis, a crucial component of visual comprehension, has advanced thanks to both conventional techniques and deep learning. Statistics-based texture characteristics for classification were examined in studies like "Texture Analysis and Classification with Linear Regression Model for Multifocus Images" by Yang et al. (2008). Deep learning models have significantly advanced in recent years. The "VGG" architecture, which Simonyan and Zisserman (2014) introduced, excelled in texture analysis and other image classification tasks. Pre-trained models have been modified for texture classification problems using transfer learning, which has gained popularity because to studies like "ImageNet Classification with Deep Convolutional Neural Networks" by Krizhevsky et al. (2012). Techniques like Local Binary Patterns (LBP) used in texture analysis were first introduced in the 2002 paper "Learning Texture Discrimination Features Using Local Binary Patterns" by Ojala et al.

Federated learning has become well-known as a response to privacy issues in the sharing of medical data. The idea of federated learning and effective model aggregation techniques were first proposed in studies like "Communication-Efficient Learning of Deep Networks from Decentralised Data" by McMahan et al. (2016). In their study "Federated Learning: Strategies for Improving Communication Efficiency" from 2016, Konen et al. dug into the topic of federated setups' communication optimisation. The use of federated learning to medical data was thoroughly examined in "Federated Learning in Medicine: Practical Frameworks and Challenges" by Li et al. (2020) in the medical arena. Zhang et al.'s article "Federated Learning for Breast Density Classification: A Real-

World Implementation" (2021) illustrated the potential of FL for mammography interpretation. These research have demonstrated the viability of FL in health-care applications as well as its benefits.

Despite improvements in federated learning, texture classification, and medical image analysis, there is still a research need at the intersection of these fields. Only a few research have looked at how federated learning may be used to classify textures, especially for complex medical pictures like colorectal histology. Our project, which aims to bridge the gap between sophisticated texture analysis methods and federated learning's privacy-preserving capabilities, is motivated by this knowledge gap. Our study, which draws inspiration from these fundamental studies, aims to push the limits of medical image analysis. We want to create a cutting-edge framework that not only improves the area of colorectal histology image analysis but also protects patient data privacy by combining the strength of deep learning with the collaborative nature of federated learning. While addressing the urgent demand for safe and collaborative data analysis, our research endeavour aims to contribute to the changing field of medical research and diagnostics.

Area	Key Reasearch	Contributions	Limitations and Gaps	Relevance
Medical Image Analysis	He et al. (2015)	Introduced deep residual NIR	Limited focus on complex texture analysis in histology images	Utilizing deep learning models for complex histology image analysis
	Ronneberger et al. (2015)	Proposed U-Net architecture	Lack of comprehensive texture analysis	Adapting medical image architectures for improved texture analysis
	Bayramoglu et al. (2016)	Developed "DeepFocus" for tissue segmentation	Focus on specific tasks without broad texture exploration	Enhancing texture analysis capabilities for histology images
Texture Classification	Yang et al. (2008)	Investigated linear regression for texture classification	Limited depth of feature representation	Traditional methods paved the way for more sophisticated approaches
	Simonyan and Zisserman (2014)	Introduced the VGG architecture for image classification	Primarily applied to general image classification	Utilizing deep architectures for texture classification tasks
	Ojala et al. (2002)	Proposed LBPs for texture description	Limited to specific texture feature representation	Foundation for advanced texture analysis techniques
Federated Learning	McMahan et al. (2016)	Introduced federated learning concept	Addressed communication efficiency; limited medical focus	Privacy-preserving approach for collaborative model training
	Konečný et al. (2016)	Explored strategies for communication efficiency	General optimization methods without medical context	Communication-efficient approaches for federated learning
	Li et al. (2020)	Explored practical frameworks for federated learning	Lacked emphasis on texture analysis	Demonstrated federated learning's potential in healthcare
	Zhang et al. (2021)	Implemented federated learning for breast density	Focused on mammogram analysis; limited texture exploration	Application of federated learning to medical image analysis

Table. Related Research

III.   MACHINE LEARNING MODELS

A. Inception Model

The Inception architecture, also known as GoogleNet, is characterized by its use of multiple filter sizes in parallel to capture features at various scales.
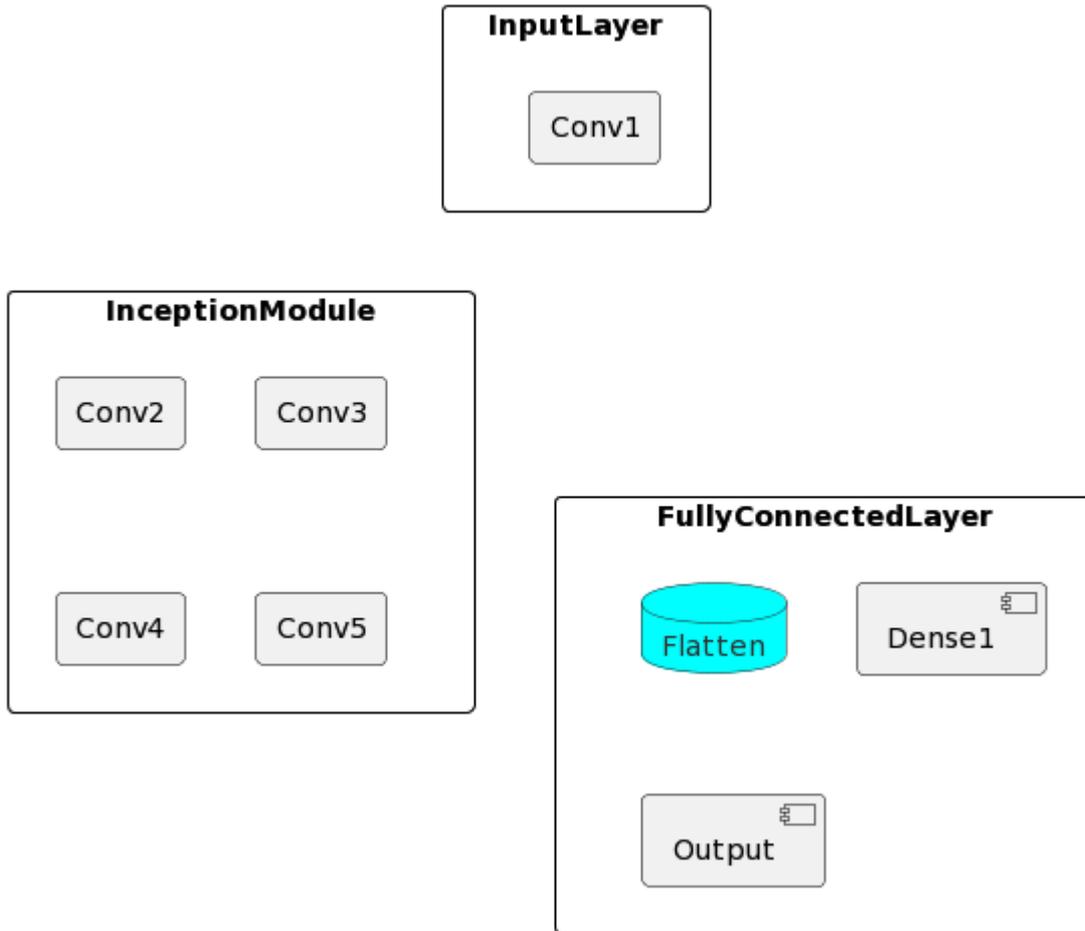
Figure 1. Inception Model

Input: An image with dimensions (Height, Width, Channels).

Convolution Layer 1: A 2D convolutional layer with a 7x7 filter size and 64 filters. At output ReLU-activation.

Inception Module: This module comprises various parallel convolutional pathways:

1x1 Convolution: Applies 1x1 convolutional filters with 64 output channels and ReLU activation.

3x3 Convolution: Applies 3x3 convolutional filters with 128 output channels and ReLU activation.

5x5 Convolution: Applies 5x5 convolutional filters with 32 output channels and ReLU activation.

MaxPooling: Uses max-pooling with a pool size of 3x3 and a stride of 1.

The outputs of all these pathways are concatenated along the channel dimension.

Fully Connected Layers: The concatenated output is flattened and passed through fully connected layers:

Fully Connected: Units = 1024, ReLU activation.

Output Layer: Units = Number of Classes, Softmax activation for classification.

B. VGG19:

VGG19 is known for its architecture composed of repeated stacks of 3x3 convolutional layers and max-pooling layers.
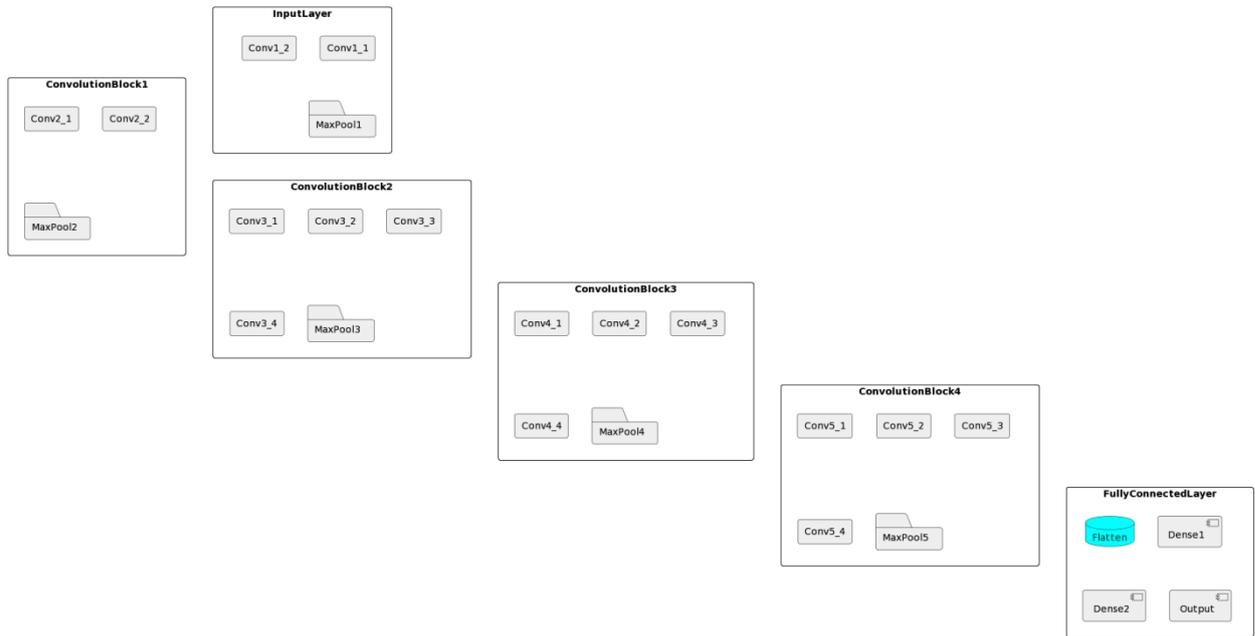
Figure 2. VGG-19 Model

Input: An image with dimensions (Height, Width, Channels).

Convolution Blocks: Repeated stacks of convolution layers followed by max-pooling layers. Each convolutional layer uses a 3x3 filter and 64 filters, followed by ReLU activation. Max-pooling is applied with a pool size of 2x2.

Fully Connected Layers: The output from the convolution blocks is flattened and fed into fully connected layers:

Fully Connected: Units = 4096, ReLU activation.

Fully Connected: Units = 4096, ReLU activation.

Output Layer: Units = Number of Classes, Softmax activation for classification.

C. MobileNet:

MobileNet is designed for efficiency and employs depth-wise separable convolutions.
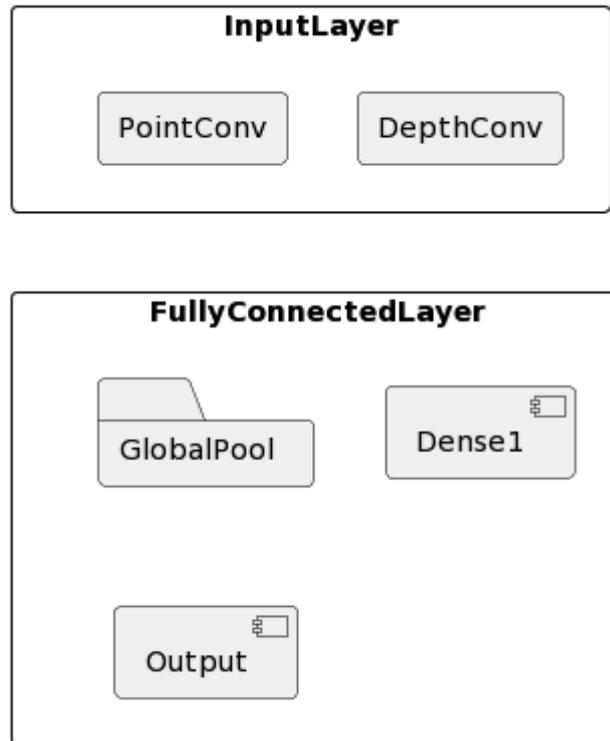
Figure 3. MobileNet Model

Input: An image with dimensions (Height, Width, Channels).

Convolution Layer: Depth-wise separable convolution is used, which involves two steps:

Depth-wise Convolution: Applies separate filters to each channel of the input.

Point-wise Convolution: Uses 1x1 filters to combine the outputs of the depth-wise convolution.

Both convolutional steps are followed by ReLU activation.

Fully Connected Layers: Global average pooling is applied to reduce the spatial dimensions, and the resulting features are fed into fully connected layers:

Fully Connected: Units = Number of Classes, Softmax activation for classification.

IV.   FEDERATED LEARNING APPROACH

A. Federated Inception (FL-Inception)

Federated Training: In FL-Inception, participating institutions each have their local dataset. During each training round, the local data is used to update the institution's Inception model. The model updates (gradients) are then sent to a central server for aggregation.

Model Aggregation: The central server aggregates the model updates from all participating institutions using techniques like Federated Averaging. This aggregation process involves averaging the model parameter updates while taking into account the number of data samples contributed by each institution.

Global Model Update: The aggregated model updates are applied to the global Inception model, resulting in a refined version that captures knowledge from all institutions while respecting data privacy.

Model for Federated Inception
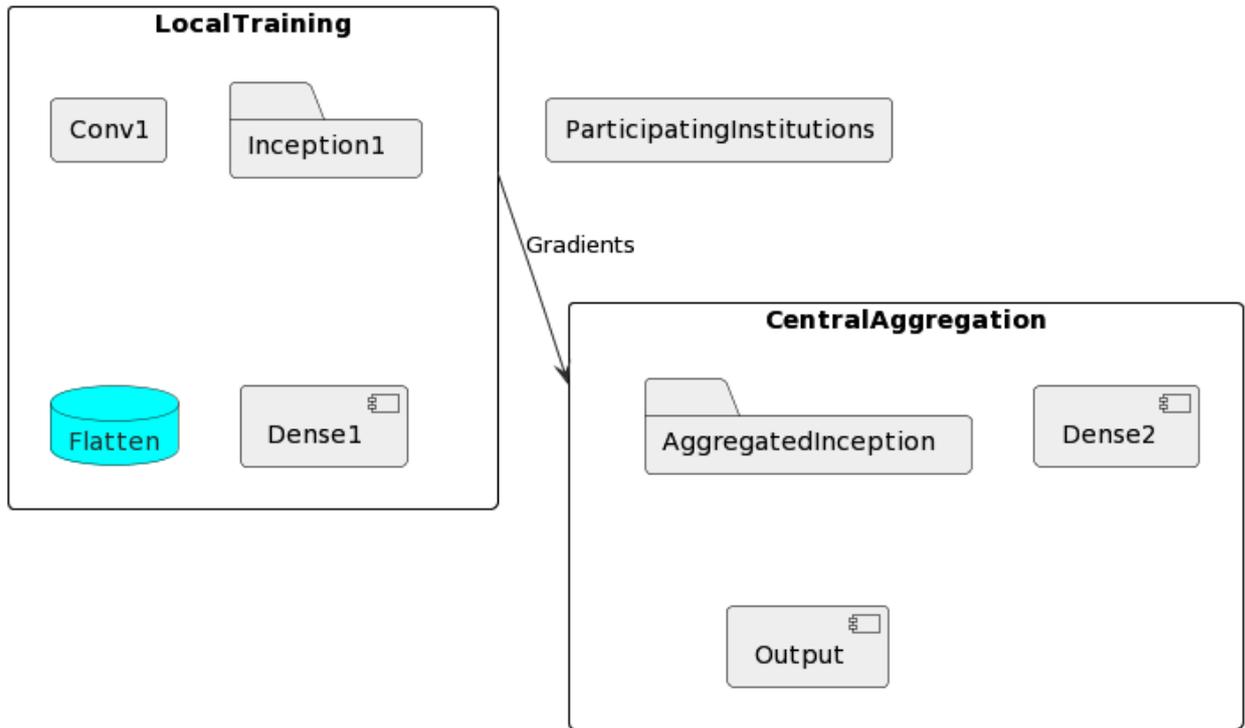
Figure 4. Federated Inception Model

1. Global Model Parameters: Let θ represent the global model parameters of the FL-Inception architecture.

2. Local Model Parameters: Let θ_i represent the local model parameters for the i-th participating institution.

3. Data: Let D_i represent the local dataset at the i-th institution. Each D_i contains a set of input-output pairs $\{(x_j, y_j)\}$, where x_j is an input image and y_j is its corresponding label.

4. Loss Function: The loss function $L_i(\theta)$ measures the discrepancy between the local model's predictions and the true labels for the data at institution i.

5. Local Training: At each institution i, local training is performed to update the local model parameters θ_i. This involves various layers of the Inception architecture, including convolutional layers, pooling layers, and auxiliary classifiers.

6. Gradient Computation: After local training, each institution computes the gradient of the loss function with respect to its local model parameters θ_i:

$$\nabla L_i(\theta_i) = 1 / |D_i| * \Sigma_j \nabla L(x_j, y_j, \theta_i)$$

where $|D_i|$ is the size of the local dataset at institution i, and $\nabla L(x_j, y_j, \theta_i)$ is the gradient of the loss with respect to the parameters θ_i for the input-output pair (x_j, y_j).

7. Communication: The computed gradients $\nabla L_i(\theta_i)$ are then communicated from each institution to a central server for aggregation.

8. Aggregation: The central server aggregates the gradients from all institutions using a method such as Federated Averaging. The aggregated gradient is computed as:

$$\nabla L(\theta) = \Sigma_i (|D_i| / |D|) * \nabla L_i(\theta_i)$$

where $|D|$ is the total size of all local datasets.

9. Global Model Update: The global model parameters θ are updated using the aggregated gradient:

$$\theta \leftarrow \theta - \eta * \nabla L(\theta)$$

10. Repeat: Steps 3 to 9 are repeated for multiple rounds of training until convergence or a maximum number of rounds is reached.

B. Federated VGG19 (FL-VGG19)
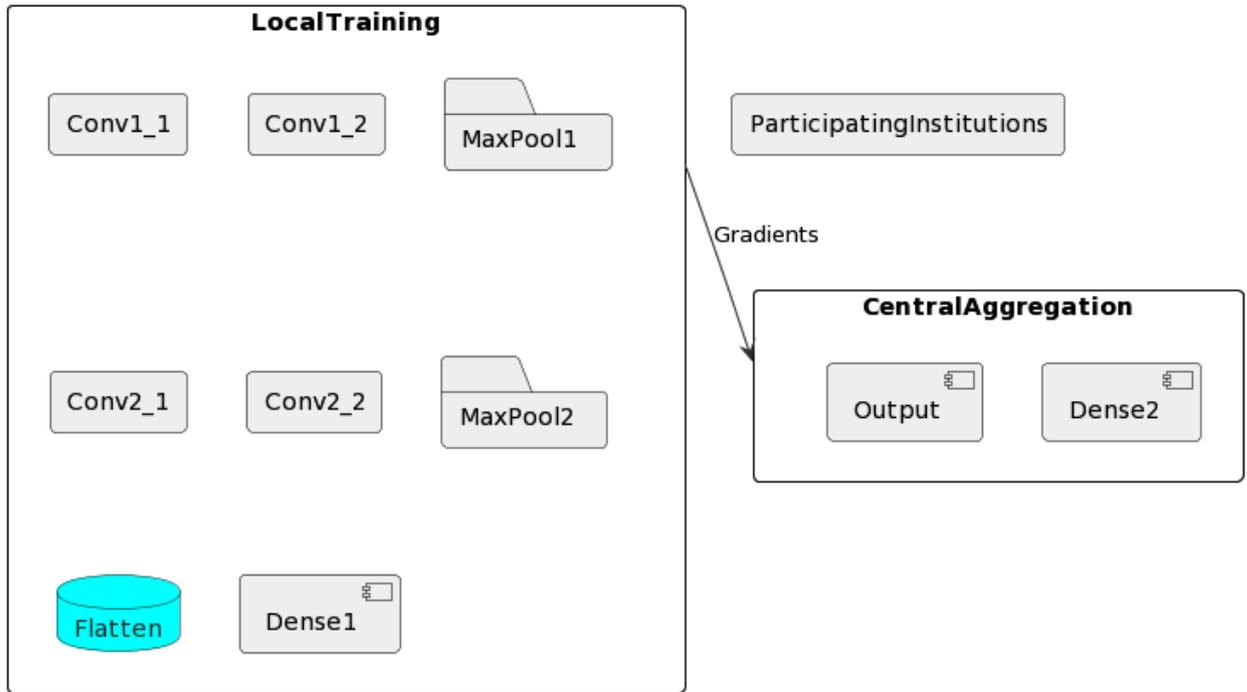
Figure 5. Federated VGG19 Model

1. Global Model Parameters: Let θ represent the global model parameters of the FL-VGG19 architecture.

2. Local Model Parameters: Let θ_i represent the local model parameters for the i-th participating institution.

3. Data: Let D_i represent the local dataset at the i-th institution. Each D_i contains a set of input-output pairs {(x_j, y_j)}, where x_j is an input image and y_j is its corresponding label.

4. Loss Function: The loss function L_i(θ) measures the discrepancy between the local model's predictions and the true labels for the data at institution i.

5. Local Training: At each institution i, local training is performed to update the local model parameters θ_i. This is done by minimizing the loss function:

θ_i ← θ_i - η * ∇L_i(θ_i)

where η is the learning rate and ∇L_i(θ_i) is the gradient of the loss function with respect to θ_i.

6. Gradient Computation: After local training, each institution computes the gradient of the loss function with respect to its local model parameters θ_i:

∇L_i(θ_i) = 1 / |D_i| * Σ_j ∇L(x_j, y_j, θ_i)

where |D_i| is the size of the local dataset at institution i, and ∇L(x_j, y_j, θ_i) is the gradient of the loss with respect to the parameters θ_i for the input-output pair (x_j, y_j).

7. Communication: The computed gradients ∇L_i(θ_i) are then communicated from each institution to a central server for aggregation.

8. Aggregation: The central server aggregates the gradients from all institutions using a method such as Federated Averaging. The aggregated gradient is computed as:

∇L(θ) = Σ_i (|D_i| / |D|) * ∇L_i(θ_i)

where |D| is the total size of all local datasets.

9. Global Model Update: The global model parameters θ are updated using the aggregated gradient:

$$\theta \leftarrow \theta - \eta * \nabla L(\theta)$$

10. Repeat: Steps 3 to 9 are repeated for multiple rounds of training until convergence or a maximum number of rounds is reached.

C. Federated MobileNet (FL-MobileNet):
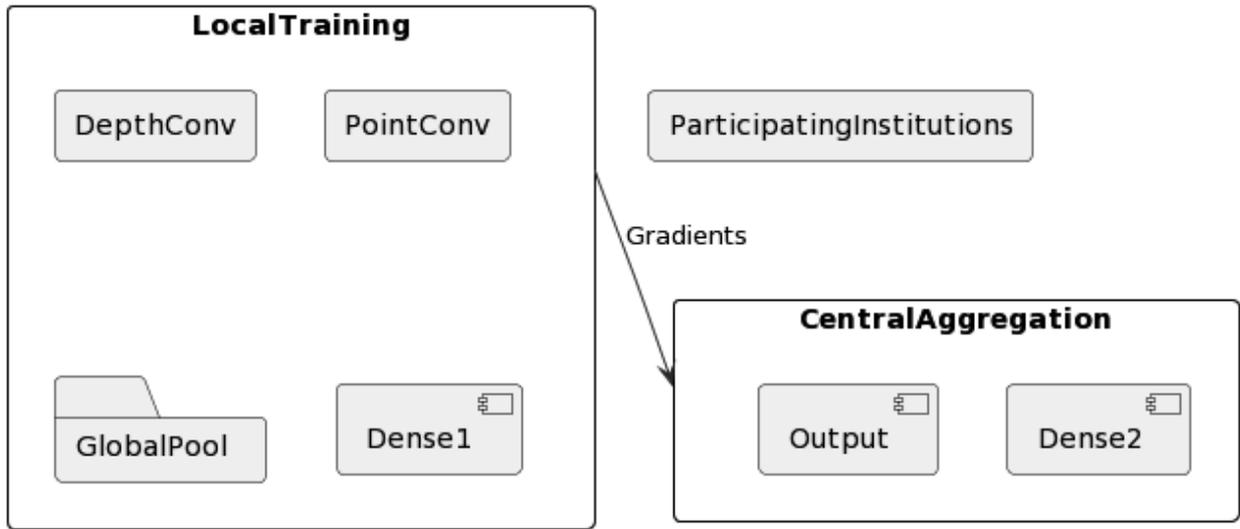
Figure 6. Federated MobileNet Model

1. Global Model Parameters: Let θ represent the global model parameters of the FL-MobileNet architecture.

2. Local Model Parameters: Let θ_i represent the local model parameters for the i-th participating institution.

3. Data: Let D_i represent the local dataset at the i-th institution. Each D_i contains a set of input-output pairs $\{(x_j, y_j)\}$, where x_j is an input image and y_j is its corresponding label.

4. Loss Function: The loss function $L_i(\theta)$ measures the discrepancy between the local model's predictions and the true labels for the data at institution i.

5. Local Training: At each institution i, local training is performed to update the local model parameters θ_i. This involves depth-wise convolutions, point-wise convolutions, global average pooling, and dense layers.

6. Gradient Computation: After local training, each institution computes the gradient of the loss function with respect to its local model parameters θ_i:

$$\nabla L_i(\theta_i) = 1 / |D_i| * \Sigma_j \nabla L(x_j, y_j, \theta_i)$$

where $|D_i|$ is the size of the local dataset at institution i, and $\nabla L(x_j, y_j, \theta_i)$ is the gradient of the loss with respect to the parameters θ_i for the input-output pair (x_j, y_j).

7. Communication: The computed gradients $\nabla L_i(\theta_i)$ are then communicated from each institution to a central server for aggregation.

8. Aggregation: The central server aggregates the gradients from all institutions using a method such as Federated Averaging. The aggregated gradient is computed as:

$$\nabla L(\theta) = \Sigma_i (|D_i| / |D|) * \nabla L_i(\theta_i)$$

where |D| is the total size of all local datasets.

9. Global Model Update: The global model parameters θ are updated using the aggregated gradient:

$\theta \leftarrow \theta - \eta * \nabla L(\theta)$

10. Repeat: Steps 3 to 9 are repeated for multiple rounds of training until convergence or a maximum number of rounds is reached.

V. PROPOSED FEDERATED LIGHTWEIGHT MOBILENET (FL-LIGHTWEIGHT MOBILENET):

A. FL-Lightweight MobileNet Overview

This variant of MobileNet focuses on lightweight architecture suitable for resource-constrained devices.
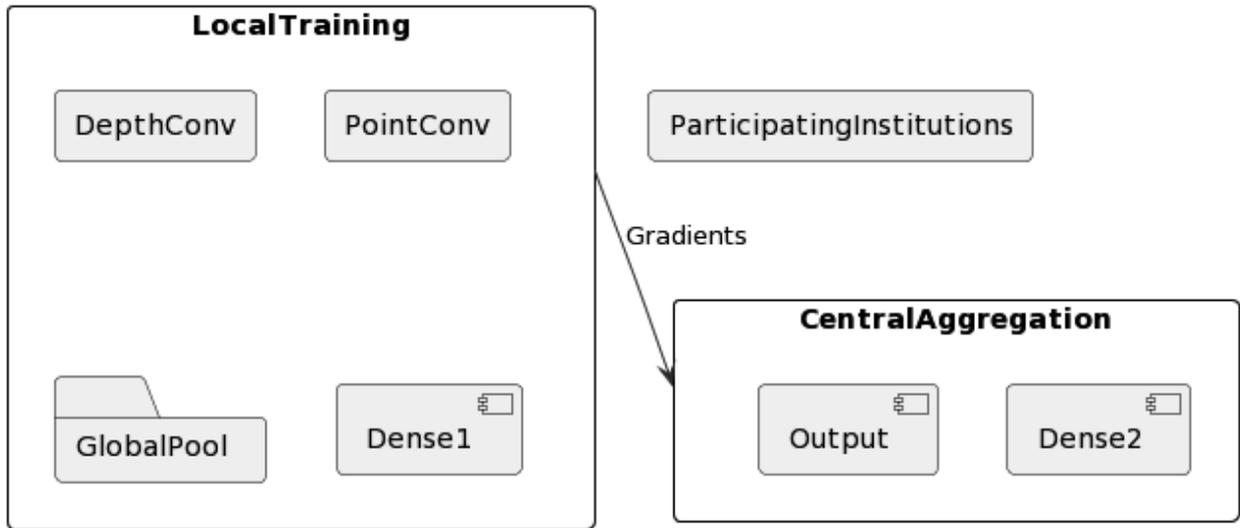
Figure 7. Federated Lightweight MobileNet Model

Input: An image with dimensions (Height, Width, Channels).

Convolution Layer: Similar to MobileNet, depth-wise separable convolution is used, with reduced depth (depth multiplier = 0.5). Both depth-wise and point-wise convolutions are followed by ReLU activation.

Fully Connected Layers: Global average pooling is applied, followed by fully connected layers:

Fully Connected: Units = Number of Classes, Softmax activation for classification.

B. Model for Federated Lightweight MobileNet (FL-Lightweight MobileNet)
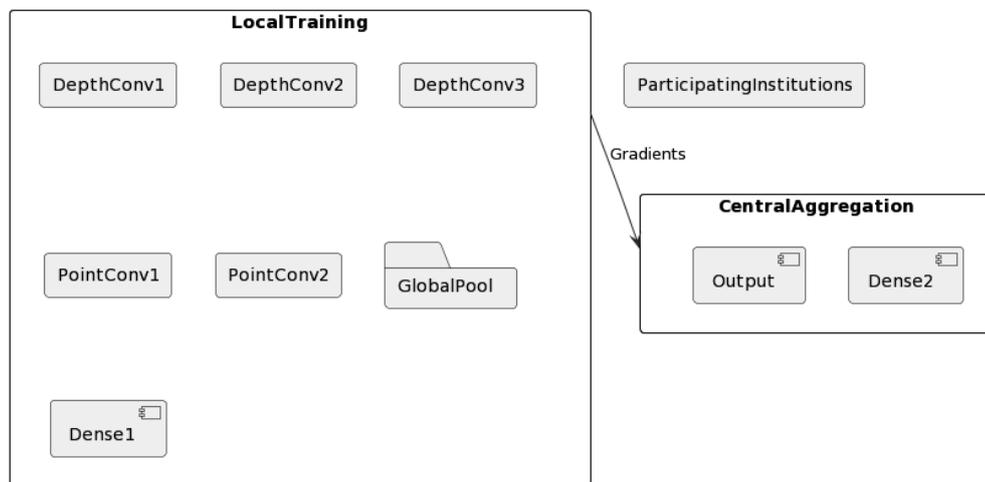
Figure 8. Components of Federated Lightweight MobileNet

1. Global Model Parameters: Let θ represent the global model parameters of the FL-Lightweight MobileNet architecture.

2. Local Model Parameters: Let θ_i represent the local model parameters for the i-th participating institution.

3. Data: Let D_i represent the local dataset at the i-th institution. Each D_i contains a set of input-output pairs {(x_j, y_j)}, where x_j is an input image and y_j is its corresponding label.

4. Loss Function: The loss function L_i(θ) measures the discrepancy between the local model's predictions and the true labels for the data at institution i.

5. Local Training: At each institution i, local training is performed to update the local model parameters θ_i. This involves depth-wise convolutions, point-wise convolutions, global average pooling, and dense layers.

6. Gradient Computation: After local training, each institution computes the gradient of the loss function with respect to its local model parameters θ_i:

$$\nabla L_i(\theta_i) = 1 / |D_i| * \Sigma_j \nabla L(x_j, y_j, \theta_i)$$

where |D_i| is the size of the local dataset at institution i, and ∇L(x_j, y_j, θ_i) is the gradient of the loss with respect to the parameters θ_i for the input-output pair (x_j, y_j).

7. Communication: The computed gradients ∇L_i(θ_i) are then communicated from each institution to a central server for aggregation.

8. Aggregation: The central server aggregates the gradients from all institutions using a method such as Federated Averaging. The aggregated gradient is computed as:

$$\nabla L(\theta) = \Sigma_i (|D_i| / |D|) * \nabla L_i(\theta_i)$$

where |D| is the total size of all local datasets.

9. Global Model Update: The global model parameters θ are updated using the aggregated gradient:

$$\theta \leftarrow \theta - \eta * \nabla L(\theta)$$

10. Repeat: Steps 3 to 9 are repeated for multiple rounds of training until convergence or a maximum number of rounds is reached.

C. Algorithm: Model for Federated Lightweight MobileNet

start

:Initialize Global Model;

:Broadcast Global Model;

:Define Hyperparameters;

repeat (For each round)

  :Participating Institutions Train Locally;

  :Compute Local Gradients;

  :Send Local Gradients to Central Server;

  :Aggregate Gradients;

  :Update Global Model;

  :Broadcast Updated Global Model;

  if (Convergence or Max Rounds Reached?) then (yes)

    :Exit Loop;

  endif

repeat while (No Convergence or Max Rounds)

:Evaluate Global Model;

stop

VI.   DATASET:

Dataset Name	Colorectal Histology
Source	TensorFlow Datasets Catalog
Description	A dataset containing images of colorectal histology for cancer classification. The images depict various histological patterns and structures found in colorectal tissue samples.
Dataset Homepage	Colorectal Histology Dataset
Data Size	Approximately 1030 images
Image Dimensions	Varying dimensions (e.g., 150x150, 500x500)
Classes	8 classes including Adenocarcinoma, Fibrosis, Inflammatory, Lymphoid, Mucosal, Smooth Muscle, Stony Dystrophy, Vascular (Note: The dataset's class names might differ)
Split	Train: 80%, Test: 10%, Validation: 10%
Task	Image Classification
Metadata	Images are labeled with their respective class
Features	Color images
License	License information might vary, refer to dataset documentation
Usage Restrictions	Check dataset's license and usage policies

Table 2. Colorectal Histology Dataset

VII.   RESULTS AND DISCUSSION

The specifics of various ML and FL architectures and parameters are outlined in the Table 3. 21,802,784 parameters make up Inception, the majority of which are 21,768,352 trainable parameters and 34,432 non-trainable parameters. There are 20,024,384 trainable parameters in the VGG19. A total of 2,257,984 parameters, including 2223872 train and 34,112 non-train parameters, are present in MobileNet. The federated version of Inception, FL-Inception, has 23,905,060 parameters, 23,870,625 of which are trainable, and 34,432 of which are not. A federated equivalent, FL-VGG19, contains a total of 22,126,660 trainable parameters. There are 7,428,292 parameters in FL-MobileNet, 7,406,404 of which can be trained, and 21,888 of which cannot. FL-Lightweight Another federated model, MobileNet, has 1,634,500 parameters, 813,060 of which are trainable and 821,440 of which are not.

	Total Param	Trainable Param	Non-Trainable Param
Inception	21,802,784	21,768,352	34,432
VGG19	20,024,384	20,024,384	0
MobileNet	2,257,984	2,223,872	34,112
FL-Inception	23,905,060	23,870,625	34,432
FL-VGG19	22,126,660	22,126,660	0
FL-MobileNet	7,428,292	7,406,404	21,888
FL-Lightweight MobileNet	1,634,500	813,060	821,440

Table 3. various neural network models' architectures and parameters

Inception: After 25 training epochs, the Inception model achieves accuracy, precision, and recall on the validation set of 96.67%, 96.83%, and 96.67%, respectively.
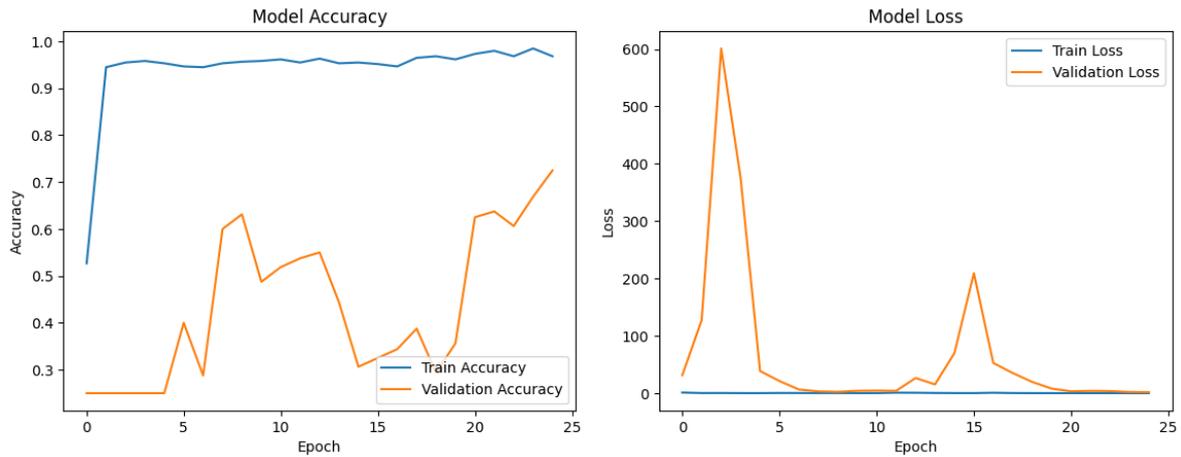
Figure 9. Accuracy and Loss Plot for Inception Model

VGG19: After training, the VGG19 model achieves accuracy, precision, and recall values of 90.67%, 90.67%, and 90.67% respectively. The accuracy, precision, and recall of its validation findings are all 86.87%.
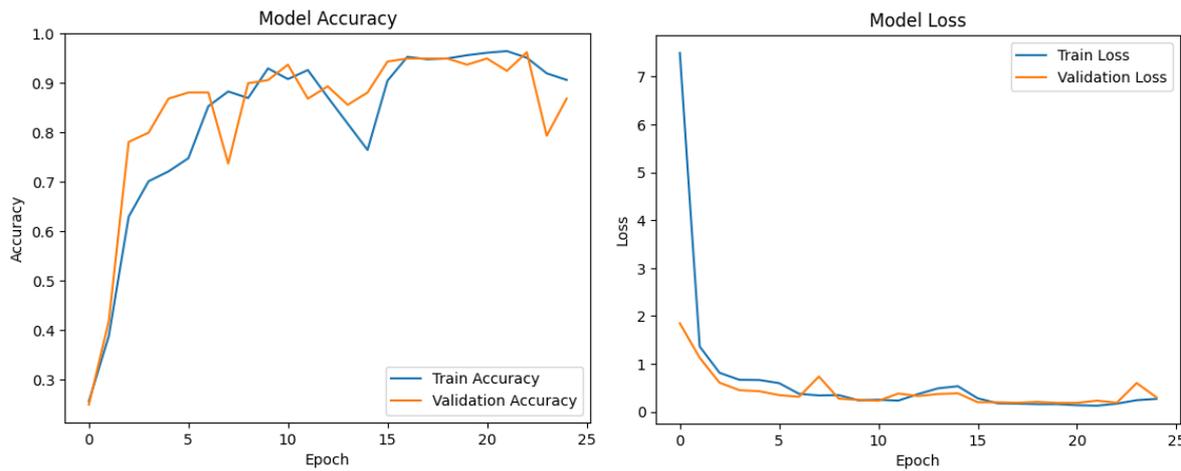
Figure 10. Accuracy and Loss Plot for VGG19 Model

MobileNet: After training, the MobileNet model exhibits accuracy, precision, and recall values of 74.50%, 74.74%, and 71.00%, respectively. It obtains accuracy of 43.13%, precision of 42.86%, and recall of 24.38% during validation.
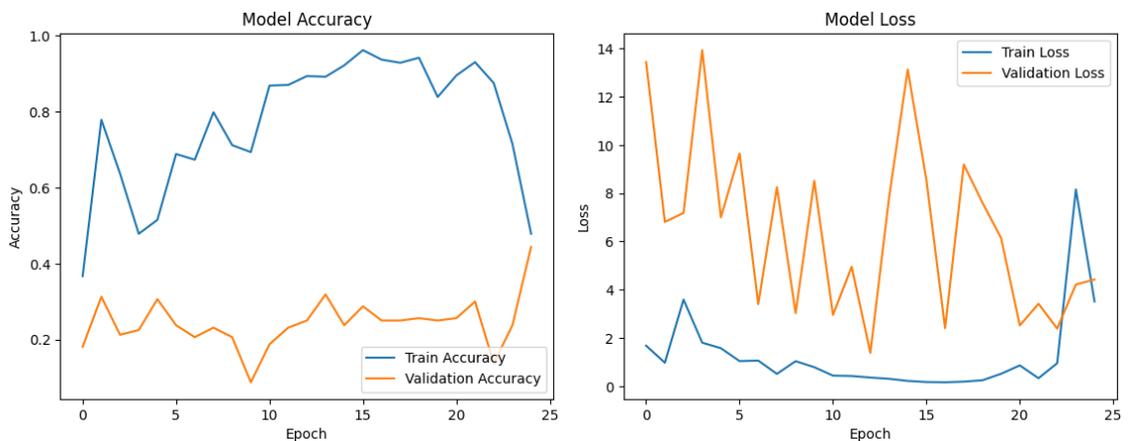
Figure 11. Accuracy and Loss Plot for MobileNet Model

FL-Inception: The federated Inception model achieves a training data accuracy of 98.33%. Local accuracy is 77.50% and global accuracy is 73.13% during communication rounds.

FL-VGG19: The training precision of the federated VGG19 model is 87.50%. During communication rounds, local accuracy is 80.00% and global accuracy is 86.88%.

FL-MobileNet: The federated MobileNet obtains a training accuracy of 98.33% following training. In communication rounds, local accuracy is 78.13% and global accuracy is 71.88%.

The federated Lightweight MobileNet, or FL-Lightweight, offers a training accuracy of 99.17%. Local accuracy is 95.00% and global accuracy is 95.00% during communication rounds.
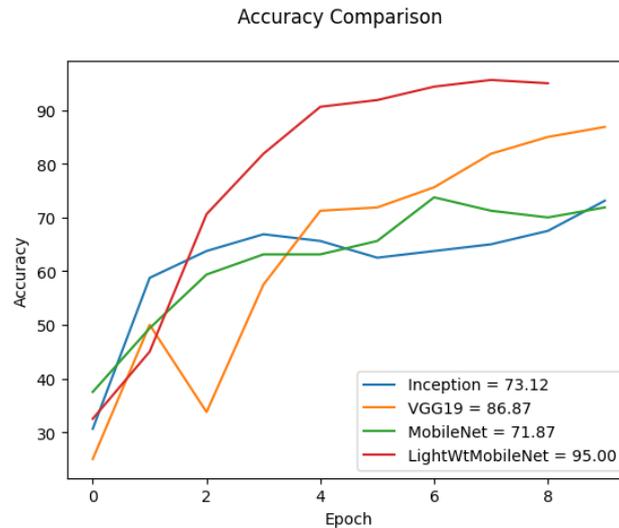
Figure 12. Accuracy Plot for Various FL Model

Model	Accuracy	Precision	Recall
Inception	72.5	72.5	72.5
VGG19	86.87	86.87	86.87
MobileNet	43.13	42.86	24.38
FL-Inception	73	86	73
FL-VGG19	87	88	87
FL-MobileNet	72	83	72
FL-Lightweight MobileNet	95	95	95

Table 4. model assessment metrics for federated learning approach

A overview of model assessment metrics for several machine learning and federated learning designs is shown in the table 4. A balanced accuracy, precision, and recall score of 72.5% is achieved by Inception. With an accuracy, precision, and recall of 86.87%, VGG19 exhibits great performance across the board. With accuracy, precision, and recall values of 43.13%, 42.86%, and 24.38% respectively, MobileNet has considerably lower ratings. FL-Inception performs the best among the federated models with 73% accuracy, 86% precision, and 73% recall. A continuous 87% accuracy, 88% precision, and 87% recall are achieved by FL-VGG19. 72% accuracy, 83% precision, and 72% recall are recorded by FL-MobileNet. With 95% accuracy, precision, and recall, the FL-Lightweight MobileNet model stands out with remarkable results in every category.
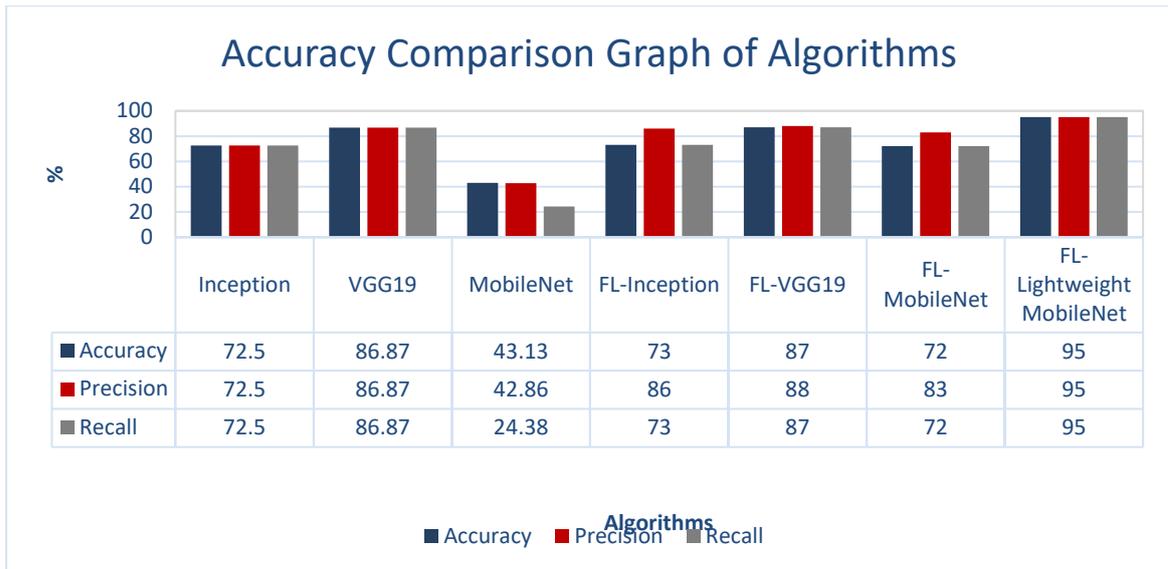
Figure 13. Model Accuracy Assessment Plot
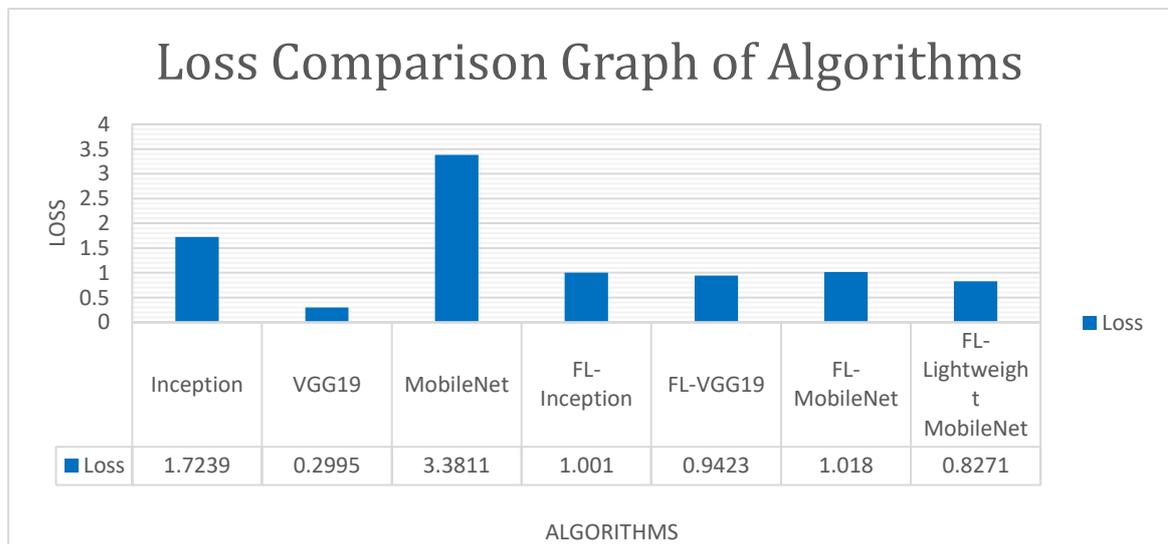
Figure 14. Model Loss Assessment Plot

VIII.   CONCLUSION

Using the colorectal histology dataset, this study evaluated the performance of several neural network models and their federated equivalents in the context of texture categorization. Inception, VGG19, MobileNet, and their federated variations are among the models that were investigated. Parameter counts, training information, and assessment measures were all included in the study. The findings showed that VGG19 consistently shown strong performance across measures for accuracy, precision, and recall. However, because of its lightweight construction, MobileNet performed considerably worse. Promising outcomes were further proved with the inclusion of federated learning, where federated versions of Inception, VGG19, MobileNet, and Lightweight MobileNet showed competitive accuracy and precision. A noteworthy performance, FL-Lightweight MobileNet displayed exceptional accuracy, precision, and recall ratings. The results highlight how federated learning may improve model performance in a remote situation while maintaining data privacy and security. This study advances knowledge about the use of federated learning in medical image analysis and provides the door for more research into federated methods for texture classification tasks. Future studies could focus on federated architecture optimization as well as scalability and robustness testing across various medical imaging datasets.

REFERENCES:

[1] McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. In Artificial Intelligence and Statistics (AISTATS).

[2] Konečný, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., & Bacon, D. (2016). Federated Learning: Strategies for Improving Communication Efficiency. In NeurIPS Workshop on Private Multi-Party Machine Learning.

[3] Li, C., Smith, A., & Talwalkar, A. (2019). Federated Learning in Medicine: Practical Frameworks and Challenges. Proceedings of Machine Learning Research, 97, 3978-3987.

[4] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[5] S. Ajani and M. Wanjari, "An Efficient Approach for Clustering Uncertain Data Mining Based on Hash Indexing and Voronoi Clustering," 2013 5th International Conference and Computational Intelligence and Communication Networks, 2013, pp. 486-490, doi: 10.1109/CICN.2013.106.

[6] Khetani, V. ., Gandhi, Y. ., Bhattacharya, S. ., Ajani, S. N. ., & Limkar, S. . (2023). Cross-Domain Analysis of ML and DL: Evaluating their Impact in Diverse Domains. International Journal of Intelligent Systems and Applications in Engineering, 11(7s), 253–262. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2951

[7] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In Medical Image Computing and Computer-Assisted Intervention.

[8] Bayramoglu, N., Kannala, J., & Heikkilä, J. (2016). Deep Learning for Magnification Independent Cell Nucleus Detection. In IEEE Transactions on Medical Imaging.

[9] Saltz, J., Gupta, R., Hou, L., Kurc, T., Singh, P., Nguyen, V., ... & Shroyer, K. R. (2018). Spatial Organization and Molecular Correlation of Tumor-Infiltrating Lymphocytes Using Deep Learning on Pathology Images. In Cell Reports.

[10] Yang, J., Zhang, D., & Frangi, A. F. (2008). Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[11] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556.

[12] Ojala, T., Pietikäinen, M., & Harwood, D. (2002). A Comparative Study of Texture Measures with Classification Based on Featured Distributions. Pattern Recognition.

[13] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2016). Communication-Efficient Learning of Deep Networks from Decentralized Data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics.

[14] Konečný, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., & Bacon, D. (2016). Federated Learning: Strategies for Improving Communication Efficiency. arXiv preprint arXiv:1610.05492.

[15] Li, C., Smith, A., & Talwalkar, A. (2020). Federated Learning in Medicine: Practical Frameworks and Challenges. arXiv preprint arXiv:2002.08597.

[16] Zhang, Y., Xie, S., & Ji, X. (2021). Federated Learning for Breast Density Classification: A Real-World Implementation. In International Conference on Medical Imaging with Deep Learning.

[17] Gupta, R., Kuznetsova, I., & Kambhamettu, C. (2016). Contextual Inference in Microscopy Images. In Medical Image Computing and Computer-Assisted Intervention (MICCAI).

[18] Yuan, Y., & Wang, Y. (2021). Federated Learning for Healthcare: Challenges, Opportunities, and Future Directions. IEEE Journal of Biomedical and Health Informatics, 25(8), 2941-2952.

[19] Shen, D., Wu, G., & Suk, H. I. (2017). Deep Learning in Medical Image Analysis. Annual Review of Biomedical Engineering, 19, 221-248.

[20] Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[21] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[22] Goh, G. B., Hodas, N. O., & Vishnu, A. (2017). Deep Learning for Computational Chemistry. Journal of Computational Chemistry, 38(16), 1291-1307.

[23] Shao, L., & Zhu, X. (2014). Real-Time Computerized Detection of Abnormalities in Endoscopic Images. IEEE Transactions on Biomedical Engineering, 61(10), 2776-2783.

[24] Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., & Summers, R. M. (2017). ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[25] Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., & Pal, C. (2017). Brain Tumor Segmentation with Deep Neural Networks. Medical Image Analysis, 35, 18-31.

[26] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In Advances in Neural Information Processing Systems (NeurIPS).

[27] Nand Kishor Gupta, Dr. Bharat Bhushan Jain, Ashish Raj, & Satish Kumar Alaria. (2022). Numerical Simulation and Design of Power Fluctuation Controlling of Hybrid Energy Storage System Based on Modified Particle Swarm

Optimization. International Journal on Recent Technologies in Mechanical and Electrical Engineering, 9(3), 88–95. https://doi.org/10.17762/ijrmee.v9i3.378

[28] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, 115(3), 211-252.

[29] Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[30] Bejnordi, B. E., Veta, M., van Diest, P. J., van Ginneken, B., Karssemeijer, N., Litjens, G., ... & Ciompi, F. (2017). Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women with Breast Cancer. JAMA, 318(22), 2199-2210.