Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 3, March : 2025

OPTIMIZING FEDERATED SHUFFLING WITH PRIVACY FOR BRAIN TUMOR MRI CLASSIFICATION

Potturi Reshma Assistant Professor, Dept of CSE Seshadri Rao Gudlavalleru Engineering College, Gudlavalleru, Andhra Pradesh, India reshmapothuri@gmail.com
Boggarapu Teja Bala Tanmai, Bhukya Srinu Naik, Bolem Naga Bhushanam, Alladi Vineetha, UG Student, Dept of CSE, Seshadri Rao Gudlavalleru Engineering College, Gudlavalleru, Andhra Pradesh, India tejabalatanmai555@gmail.com

ABSTRACT

The increasing use of medical imaging for diagnosing brain tumors has highlighted the need for effective and privacy-preserving machine learning models. This study presents a solution for brain tumor classification from MRI scans using federated learning, which ensures patient data privacy by keeping the data decentralized and only sharing model updates. A deep learning approach, utilizing transfer learning with the VGG-16 architecture, is employed to classify images into different tumor types: pituitary, meningioma, glioma, and no tumor. Data augmentation techniques are applied to enhance the training dataset, addressing class imbalance issues. The model is trained in a federated learning environment with multiple clients, where each client trains the model on their local dataset, and the results are aggregated to update the global model. After training, the model's performance is evaluated using metrics such as classification accuracy, precision, recall, F1-score, and a confusion matrix. This approach allows for high accuracy in tumor classification while ensuring data privacy through federated learning. The proposed method aims to facilitate efficient, privacy-preserving medical image classification in a distributed setting, particularly in scenarios with sensitive data.

Keywords: **Federated Learning, VGG-16, Transfer Learning, Brain Tumor MRI Classification, Privacy, Model Aggregation, TensorFlow, Keras, Federated Shuffling, Data Augmentation, Image Preprocessing, Model Training, Image Classification**

I. INTRODUCTION

This paper explores the advancements and open problems in federated learning, focusing on its potential applications and challenges. The authors provide an extensive review of the state-of-the-art techniques, including the trade-offs between model accuracy and privacy concerns, while discussing the scalability and efficiency of federated learning systems [1]. The study highlights the importance of ensuring data privacy while still enabling robust machine learning models, and the authors propose future directions for improving federated learning techniques to address these issues.

This paper surveys the privacy-preserving techniques in federated learning, outlining a comprehensive taxonomy and identifying key challenges in maintaining privacy while achieving effective learning outcomes. The authors review existing methods, including differential privacy, and propose a detailed framework for the future development of privacy-preserving federated learning approaches [2]. The study stresses the importance of balancing privacy and utility in machine learning, advocating for new solutions that can better protect users' sensitive information during federated training processes.

This paper investigates learning rate adaptation strategies in differentially private learning environments, emphasizing their role in optimizing model performance while maintaining strong privacy guarantees. The authors propose new methodologies to adapt learning rates dynamically during training to ensure that models converge efficiently without compromising privacy [3]. The

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 3, March : 2025

study underscores the critical need for developing privacy-preserving techniques that do not degrade the effectiveness of machine learning models.

This paper discusses privacy-preserving federated data sharing, exploring the feasibility of secure data sharing mechanisms within federated learning frameworks. The authors propose a novel approach to preserve privacy while enabling effective data sharing among distributed entities [4]. The study highlights the challenges involved in achieving secure and efficient federated data sharing and emphasizes the importance of innovative solutions for maintaining both privacy and utility.

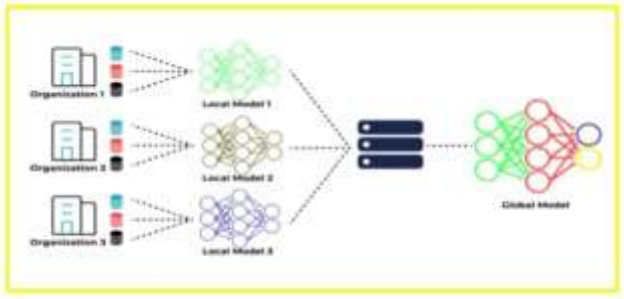

Fig 1.1: Federated Learning Environment

This paper introduces the concept of federated f-differential privacy, focusing on the application of differential privacy techniques within federated learning systems. The authors present a framework for ensuring privacy during model training, particularly when dealing with sensitive data from multiple sources [5]. The study advocates for the integration of federated learning and differential privacy to create more secure and privacy-preserving machine learning systems, paving the way for future research and real-world applications.

II. LITERATURE SURVEY

Cramer et al. explored secure multiparty computation and secret sharing techniques, providing a comprehensive overview of the foundational principles behind securely computing functions over multiple parties without exposing private inputs. Their work emphasized the significance of these methods in ensuring privacy and data security in distributed systems, thus laying the groundwork for privacy-preserving computing in the context of federated learning [6]. **Dwork** introduced the concept of differential privacy, a technique aimed at ensuring privacy when analyzing or sharing data. Her work, presented in a lecture series, is pivotal in understanding how to protect individual data while still allowing for meaningful aggregate analysis. This foundational concept has become central to modern privacy-preserving techniques, especially in machine learning models that rely on large datasets [7]. **Dwork and Roth et al.** further advanced differential privacy by outlining its algorithmic foundations, elaborating on how privacy guarantees can be integrated into machine learning algorithms. Their work provides a detailed theoretical framework for implementing differential privacy in various computational settings, helping shape privacy-preserving machine learning algorithms that are widely adopted today [8]. **Kairouz et al.** focused on extremal mechanisms for local differential privacy, addressing the challenge of maintaining privacy in decentralized environments. Their research showed how these mechanisms can be used to protect privacy without sacrificing the utility of the data, contributing significantly to the development of privacy-preserving algorithms in federated learning [9]. **Ding et al.** proposed techniques for collecting telemetry data privately in distributed systems. Their work discusses methods for gathering system data while ensuring that sensitive information remains protected, making significant strides toward the application of privacy-preserving techniques in large-scale data collection scenarios, which are critical for the development of federated learning frameworks [10]. **Han et al.** explored differentially private distributed constrained optimization, focusing on how to achieve privacy guarantees in optimization tasks that involve distributed systems. Their work is particularly relevant to federated learning, where models are trained across multiple decentralized data sources while maintaining privacy [11]. **Beaude et al.** introduced a privacy-preserving method to optimize distributed resource allocation. Their work presents a strategy to

balance the privacy of individual data sources with the need for efficient resource distribution in distributed systems, an important aspect of federated learning where data privacy and optimization need to coexist [12]. **Caldas et al.** addressed the challenge of reducing client resource requirements in federated learning. Their research explores how to optimize the efficiency of federated learning by minimizing the computational burden on individual clients, thus expanding its reach and making it more practical for real-world applications [13]. **He et al.** investigated federated learning techniques for large convolutional neural networks (CNNs) at the edge, specifically for mobile and IoT devices. Their work demonstrates how group knowledge transfer techniques can be applied to train large models on decentralized data, offering a practical solution for deploying complex models in federated settings [14]. **Hamer et al.** proposed FedBoost, a communication-efficient algorithm for federated learning. Their research focuses on improving the communication efficiency in federated learning systems, which is crucial for real-world deployments of machine learning models in distributed environments. FedBoost reduces communication overhead while maintaining model accuracy, addressing one of the key limitations of federated learning [15]. **Rieke et al.** discussed the future of digital health with federated learning, focusing on its applications in medical imaging and healthcare. Their research highlights the potential of federated learning to securely train models on sensitive health data while maintaining patient privacy, showcasing the transformative potential of federated learning in the field of digital health [16].

III. PRELIMINARIES

Federated Learning Model (federated_model.py) This Python script defines the federated learning setup for the brain tumor MRI classification task. It involves splitting the dataset across multiple clients and performing model training and aggregation on each client.

- *Client Data Preparation*: Splits the training data into multiple clients for federated learning.
- *Model Definition*: Uses a VGG-16 model pre-trained on ImageNet for feature extraction and adds custom layers for classification.
- *Federated Training Loop*: Trains the model locally on the client, aggregates the weights, and updates the global model.
- *Model Aggregation*: After local training, the model weights are averaged across clients to obtain a global model.
- *Evaluation and Testing*: After federated training, the model is evaluated on a test set to compute metrics like accuracy and confusion matrix.

Model Evaluation and Metrics (model_evaluation.py) This script is responsible for evaluating the performance of the trained federated model. It calculates various metrics like accuracy, precision, recall, and F1-score using classification_report and visualizes the model's performance using a confusion matrix.

- *Confusion Matrix Visualization*: Displays the confusion matrix to show how well the model is performing across different tumor types.
- *Accuracy and Loss Tracking*: Plots the model's accuracy and loss during training to visualize learning progress.

Federated Learning Setup and Data Preprocessing (data_processing.py) This module handles data loading, augmentation, and preprocessing for federated learning:

- *Data Shuffling*: Shuffles the dataset to ensure randomness and avoid bias in training.
- *Image Augmentation*: Applies random transformations like brightness and contrast adjustments to enhance the model's generalization capability.
- *Image Preprocessing*: Resizes images to a fixed size (128x128) and normalizes them to a range [0, 1].

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 3, March : 2025

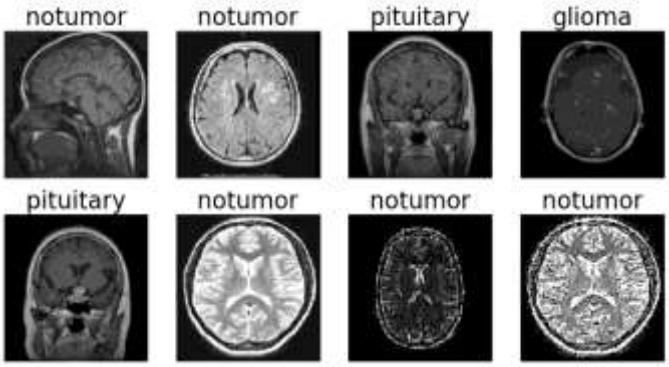

Fig 3.1: Images after Preprocessing

• *Label Encoding and Decoding*: Encodes categorical labels into numerical values for model training and decodes them back for evaluation.

Transfer Learning Setup (transfer_learning.py) This component initializes the VGG-16 base model and incorporates transfer learning by freezing layers and fine-tuning the last few layers:

• *VGG-16 Base Model*: Loads a pre-trained VGG-16 model for feature extraction.

• *Freezing Layers:* Freezes the weights of all layers except the last few convolutional blocks.

• *Custom Dense Layer*: Adds a fully connected dense layer followed by a softmax output layer for classification.

Federated Learning Client Simulation (federated_client.py) This Python script simulates the federated learning client model that handles local training:

• *Data Partitioning*: Splits the dataset into smaller batches, each representing data for a specific client.

• *Local Training*: Trains the model on each client's data and updates local weights.

• *Model Update*: After local training, the client sends back the model's updated weights to the central server for aggregation.

Execution Flow

• *Federated Learning Simulation*: Simulates federated learning where multiple clients train their models locally and send back the updated weights to the server.

•*Model Aggregation*: The server aggregates the weights received from clients and updates the global model.

• *Testing and Evaluation*: After federated training, the model is tested on a separate test dataset, and performance metrics are generated.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 3, March : 2025

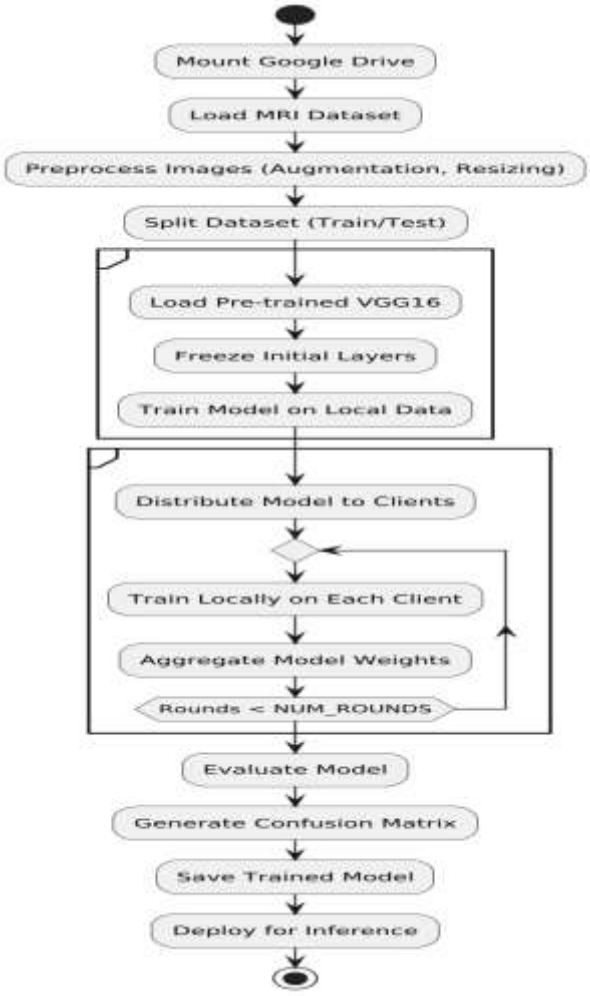

Fig 3.2: Flow diagram

Security Features

• *Data Privacy*: The federated learning setup ensures that clients train their models without sharing raw data, protecting user privacy.

• *Local Model Updates*: Each client computes model updates locally, and only model weights are exchanged, ensuring sensitive data is not exposed during training.

• *Model Fine-Tuning*: The model is fine-tuned on each client's data, ensuring it adapts to each client's local distribution.

Usage The federated learning system is designed to run on multiple clients, and the global model is iteratively updated after each round of training.

• The model can be deployed to classify brain tumor MRI images by leveraging the federated learning setup for distributed training across multiple clients.

IV. DATASET EXPLANATION

A. Dataset Overview:

1) Image Data: This dataset consists of MRI images of brain tumors categorized into various types, including Pituitary Tumor, Glioma, Meningioma, and No Tumor (Notumor).

2) Labels: Each image is associated with a label representing the tumor type or the absence of a tumor.

B. Features (Data Points):

1) Images:

• The core data consists of brain MRI images, where each image is of size 128x128 pixels, processed and augmented for training.

- Images are loaded in RGB format and go through data augmentation such as random brightness and contrast adjustments to improve model robustness.
- Each image is associated with one of the following labels: 'pituitary', 'notumor', 'meningioma', or 'glioma'.

2) Labels:

- Pituitary: MRI scans that represent pituitary tumors.
- Notumor: MRI scans with no detectable tumor.
- Meningioma: MRI scans that represent meningioma tumors.
- Glioma: MRI scans representing glioma tumors.

3) *Augmentation Techniques*:

- The images are augmented to introduce variability into the dataset and improve the model's ability to generalize. This includes brightness and contrast adjustments applied to each image randomly.

4) Categorical Labels:

- Each MRI image is associated with a label representing its class. These labels are encoded as numerical values (e.g., pituitary as 0, meningioma as 1, etc.) for model training.

C. Target Distribution:

1) *Class Distribution*: - The dataset includes images of different tumor types (Pituitary, Meningioma, Glioma) and non-tumor images (Notumor). - A pie chart visualizes the distribution of these labels within the training set. - The dataset is not balanced, as the number of images per class may vary, which can be accounted for during training (e.g., through techniques like class weighting or oversampling).

2) *Train-Test Split*: - The data is split into a training set and a testing set. A pie chart represents the split between training and testing data to show the proportion of data used for training the model versus evaluating its performance.

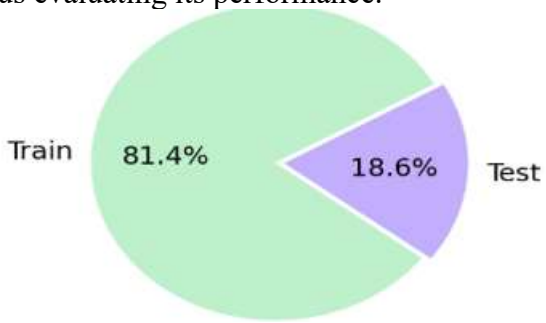

Fig 4.1: Split between the training and testing dataset

D. Data Preprocessing:

• *Image Loading:* - Images are loaded from the specified directories and resized to a consistent size (128x128 pixels). - The images are then augmented (brightness and contrast enhancement) to improve the robustness of the model during training.

• *Label Encoding*: - Categorical labels (tumor types) are converted into numerical values using label encoding (e.g., 'pituitary' becomes 0, 'glioma' becomes 1).

• *Normalization*: - The pixel values of the images are normalized to a range of [0, 1] for better model performance and stability.

• *Data Shuffling*: - The dataset is shuffled to ensure randomness and reduce bias during training.

E. Correlation Analysis:

• *Image and Label Correlation*: - An analysis of the correlation between specific tumor types and image features could be done to understand common patterns in MRI scans. This could involve examining the pixel intensities or shapes associated with certain tumor types.

• *Class Distribution*: - Understanding the distribution of the different tumor classes in the dataset helps in identifying if class imbalances exist, which could impact model performance.

• *Augmentation Impact*: - Evaluating how different augmentation techniques (brightness, contrast, etc.) impact model performance can guide further improvements in data processing.

F. Visualization:

• *Image Visualization*: - A set of images is visualized along with their corresponding labels to check the variety of images in the dataset and their associated tumor types.

• *Class Distribution Visualization*: - A pie chart shows the distribution of different tumor types in the training set, which helps in understanding the balance or imbalance of the dataset.

• *Confusion Matrix*: - A confusion matrix is used to visualize the performance of the classification model across the different tumor types. It shows the true positives, false positives, true negatives, and false negatives, providing insights into areas where the model may need improvement.

• *Model Accuracy and Loss*: - Line plots represent the training accuracy and loss over epochs, which helps in evaluating the model's learning behavior and detecting any issues such as overfitting.

G. Additional Considerations:

• *Model Evaluation*: - The model is evaluated using classification metrics such as accuracy, precision, recall, and F1-score to measure its performance on the test set. - The results are displayed using the classification_report function, which provides a comprehensive evaluation of the model's performance.

• *Confusion Matrix Insights*: - The confusion matrix provides detailed insights into how well the model is distinguishing between different tumor types, showing where misclassifications occur and helping to improve the model.

V. METHODOLOGY

A. Data Loading and Initial Exploration

1) Dataset Setup:

- The dataset consists of MRI images for brain tumor classification, stored in two directories: Training and Testing. Each image is associated with a label that represents one of four categories: pituitary tumor, meningioma, glioma, or no tumor.
- The dataset is loaded and paths to the images along with their corresponding labels are gathered.

2) Initial Data Exploration:

- A basic exploration of the dataset is performed to understand its composition, including the distribution of different tumor types. A pie chart is generated to visualize the proportions of each category in the training data.
- The training and testing datasets are shuffled to ensure randomness and avoid any inherent order bias.

B. Statistical Analysis and Visualization

1) Data Augmentation:

- Augmentation techniques such as random brightness and contrast adjustment are applied to images to improve the model's ability to generalize. This is especially important since the model may otherwise overfit to specific features of the training images.

2) Label Encoding:

- The categorical labels (e.g., 'pituitary', 'meningioma', 'glioma', and 'notumor') are encoded into numerical values, which are required for model training.

3) *Data Visualization*:

- The dataset is visualized using graphs like pie charts to show the distribution of tumor types within the training and testing datasets.
- Sample images are displayed with their respective labels to understand the variety in the data and confirm correct loading and augmentation.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 3, March : 2025

C. Data Preprocessing

1) Image Preprocessing:

- The images are resized to a consistent size (128x128 pixels) and normalized to a range between 0 and 1. This helps the model learn effectively by standardizing input image sizes, scales.
- Augmented images are processed to introduce variability in the dataset, improving model robustness and generalization.

2) *Model Training and Data Handling:*

- Data is processed in batches, which are generated dynamically using the datagen function. The function handles both loading and augmenting the images for each batch, making the training process efficient.
- The encoded labels are used in training the model, which is compiled with the Adam optimizer and sparse categorical cross-entropy loss function to handle multi-class classification.

D. Model Training and Evaluation

1) *Transfer Learning:*

- The base of the model is the pre-trained VGG-16 network, which is fine-tuned for the task of classifying brain tumors. The convolutional layers are frozen to retain the learned features, and the fully connected layers are fine-tuned.
- The VGG-16 model is adjusted by adding a custom dense layer followed by a dropout layer to prevent overfitting. The output layer uses softmax activation to classify images into one of the tumor categories.

2) Federated Learning Simulation:

- The dataset is divided into NUM_CLIENTS (10 in this case) for federated learning, simulating a scenario where each client trains a model on its local data.
- Federated learning is performed over multiple rounds (NUM_ROUNDS = 3), where the global model is updated after each round by averaging the weights of the models from selected clients.

3) Model Evaluation:

- After training, the model's performance is evaluated using a separate test dataset, with metrics such as accuracy, precision, recall, and F1-score.
- A confusion matrix is used to visualize how well the model performs in distinguishing between different types of brain tumors.

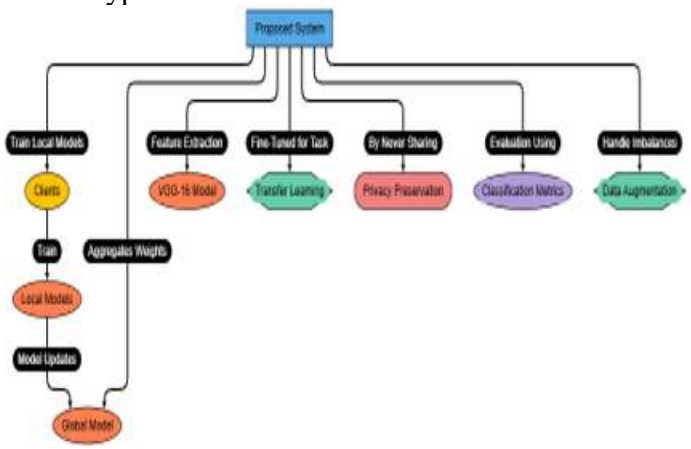

Fig 5.1: Architecture Diagram

E. Reporting and Feedback

1) *Model Evaluation Report:*

- The performance of the model is summarized using a classification report, providing detailed metrics for each tumor type (e.g., accuracy, precision, recall, and F1-score).
- A confusion matrix is plotted to understand how well the model is identifying each category and where misclassifications occur.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 3, March : 2025

2) Model Deployment:

- After evaluation, the final model is saved (my_model.h5) for deployment in real-world applications. The trained model can be used for predicting the type of brain tumor from MRI images.

F. Continuous Monitoring and Update

1) Federated Learning and Model Updates:

- The federated learning setup allows for continuous updates to the global model without requiring the clients to share raw data, maintaining privacy. As new data is collected, it can be added to the federated system for further training.

2) Model Fine-Tuning:

- Fine-tuning continues across various rounds, ensuring that the model adapts to new data and the latest advancements in medical imaging.

3) Continuous Improvement:

- Based on the results and evaluation, new strategies like class weighting, more sophisticated augmentation techniques, or deeper models (e.g., adding more layers or trying other architectures) may be explored to improve the model's accuracy.

VI. RESULTS

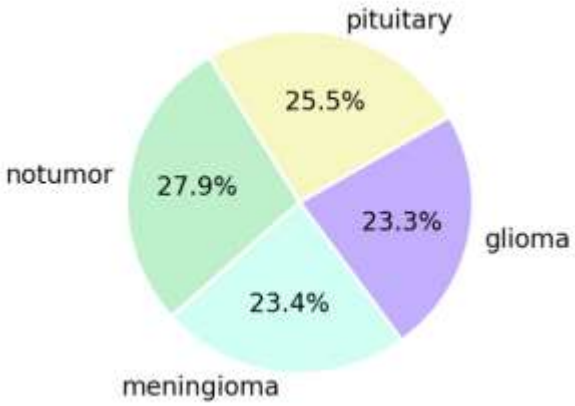

Fig 6.1: Data visualization of the MRI dataset

Fig 6.2: Image Classification UI

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 3, March : 2025

Fig 6.3: Input image for tumor prediction

Fig 6.4: Output of brain tumor prediction

	precision	recall	f1-score	support
glioma	0.97	0.94	0.95	300
meningioma	0.93	0.97	0.95	306
notumor	1.00	0.99	0.99	405
pituitary	0.99	0.99	0.99	300
accuracy			0.97	1311
macro avg	0.97	0.97	0.97	1311
weighted avg	0.97	0.97	0.97	1311

Fig 6.5: Classification Report

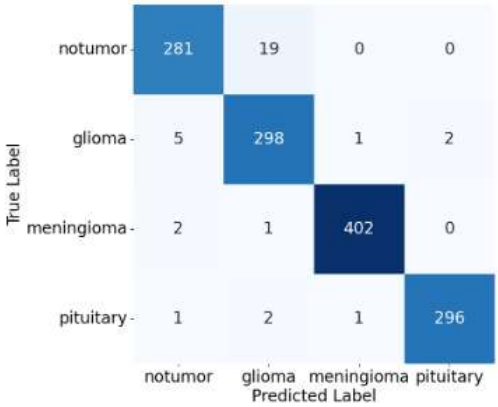

Fig 6.6: Confusion Matrix

VII. CONCLUSION

The implementation of federated learning for brain tumor MRI classification offers a robust solution that addresses both privacy concerns and the need for efficient, scalable machine learning. In this approach, a pre-trained VGG-16 model, renowned for its feature extraction capabilities, serves as the foundation for the classification task. By fine-tuning this model using a dataset of MRI images of brain tumors, it is able to classify different tumor types, including pituitary tumors, gliomas, meningiomas, and non-tumor cases. The dataset is augmented with random brightness and contrast adjustments to introduce variability, which improves the model's ability to generalize across unseen data.

A significant aspect of this approach is the use of federated learning, which allows the training process to be decentralized. In this setup, the dataset is split across multiple clients, each training a

local model without sharing their raw data. This ensures that sensitive patient data, which might be contained in the MRI images, remains private. The local models are periodically synchronized by aggregating the weights of each client's model, thus ensuring that the global model benefits from the learning of all clients while maintaining data privacy.

The model's performance is evaluated rigorously using common classification metrics, such as accuracy, precision, recall, and F1-score, which are summarized in a classification report. Additionally, the confusion matrix provides a detailed visualization of how well the model distinguishes between the various tumor types, offering valuable insights into areas for further improvement. This evaluation process confirms that the federated learning model performs effectively in classifying tumor types.

After several rounds of federated training and model aggregation, the final model is deployed, ready to classify MRI images of brain tumors with high accuracy. This deployment demonstrates the potential for federated learning to scale across healthcare institutions, enabling them to collaboratively train high-quality models without compromising patient privacy. In summary, this approach highlights the growing role of machine learning in medical applications, particularly in sensitive areas like medical imaging, where privacy and accuracy are critical.

REFERENCES

[1] P. Kairouz et al., "Advances and open problems in federated learning," Found. Trends Mach. Learn., vol. 14, nos. 1–2, pp. 1–210, 2019.
[2] X. Yin, Y. Zhu, and J. Hu, "A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions," ACM Comput. Surv., vol. 54, no. 6, pp. 1–36, Jul. 2022.
[3] A. Koskela and A. Honkela, "Learning rate adaptation for differentially private learning," in Proc. Int. Conf. Artif. Intell. Statist., vol. 108, S. Chiappa and R. Calandra, Eds., 2020, pp. 2465–2475.
[4] F. Fioretto and P. V. Hentenryck, "Privacy-preserving federated data sharing," in Proc. 18th Int. Conf. Auto. Agents MultiAgent Syst., (AAMAS), E. Elkind, M. Veloso, N. Agmon, and M. E. Taylor, Eds., 2019, pp. 638–646.
[5] Q. Zheng, S. Chen, Q. Long, and W. Su, "Federated f-differential privacy," in Proc. 24th Int. Conf. Artif. Intell. Statist., vol. 130, A. Banerjee and K. Fukumizu, Eds., 2021, pp. 2251–2259.
[6] R. Cramer, I. B. Damgrd, and J. B. Nielsen, Secure Multiparty Computation and Secret Sharing. Cambridge, U.K.: Cambridge Univ. Press, 2015.
[7] C. Dwork, "Differential privacy," in Automata, Languages and Programming (Lecture Notes in Computer Science), vol. 4052, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds. Berlin, Germany: Springer, 2006, pp. 1–12.
[8] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," Found. Trends Theor. Comput. Sci., vol. 9, nos. 3–4, pp. 211–407, 2014.
[9] P. Kairouz, S. Oh, and P. Viswanath, "Extremal mechanisms for local differential privacy," J. Mach. Learn. Res., vol. 17, pp. 492–542, Jan. 2016.
[10] B. Ding, J. Kulkarni, and S. Yekhanin, "Collecting telemetry data privately," in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 3571–3580.
[11] S. Han, U. Topcu, and G. J. Pappas, "Differentially private distributed constrained optimization," IEEE Trans. Autom. Control, vol. 62, no. 1, pp. 50–64, Jan. 2017.
[12] O. Beaude, P. Benchimol, S. Gaubert, P. Jacquot, and N. Oudjane, "A privacy-preserving method to optimize distributed resource allocation," SIAM J. Optim., vol. 30, no. 3, pp. 2303–2336, Jan. 2020.
[13] S. Caldas, J. Konecny, H. B. McMahan, and A. Talwalkar, "Expanding the reach of federated learning by reducing client resource requirements," in Proc. Workshop Federated Learn. Data Privacy Confidentiality, FLNeurIPS, 2019.
[14] C. He, M. Annavaram, and S. Avestimehr, "Group knowledge transfer: Federated learning of large CNNs at the edge," in Proc. Int. Conf. Neural Inf. Process. Syst. Red Hook, NY, USA: Curran Associates, 2020, pp. 14068–14080, Art. no. 1180.
[15] J. Hamer, M. Mohri, and A. T. Suresh, "FedBoost: A communicationefficient algorithm for federated learning," in Proc. Int. Conf. Mach. Learn., vol. 119, H. D. III and A. Singh, Eds., Jul. 2020, pp. 3973–3983.