Federated GAN Based Biomedical Image Augmentation and Classification for Alzheimer's Disease

by

Aditya Roy
19101414
Md. Mahbubur Rahman
19101069
Shafi Ahmed
19101424

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
September 2022

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Aditya Roy
19101414

Md. Mahbubur Rahman
19101069

Shafi Ahmed
19101424

Approval

The thesis titled "Federated GAN Based Biomedical Image Augmentation and Classification for Alzheimer's Disease" submitted by

1. Aditya Roy(19101414)
2. Md. Mahbubur Rahman(19101069)
3. Shafi Ahmed(19101424)

Of Summer, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on September, 2022.

Examining Committee:

Supervisor:
(Member)

Md. Golam Rabiul Alam, PhD
Professor
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Saadat Hasan Khan
Lecturer
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Federated Learning (FL) is a distributed machine learning approach that can develop a global or customized model from scattered datasets on edge devices using federated datasets. 'Federated GAN Based Biomedical Image Augmentation and Classification for Alzheimer's Disease' will focus on augmenting the medical images using Federated Generative Adversarial Network. Due to patient-doctor confidentiality, the scarcity of data in the medical sector is a massive hindrance to the advancement of machine learning models in this sector. Our study aims to augment the existing medical data, in this case, magnetic resonance imaging(MRI) images of the brain, and test that augmented dataset on existing classification models to evaluate our generated MRI images' quality. Generative Adversarial Networks (GANs) have been utilized in order to synthesize realistic and varied Alzheimer's disease affected MRI images in order to cover the data shortage in the actual medical image distribution and identify Alzheimer's disease using Federated Learning. We expect our proposed model to successfully augment the medical images and be over 90% accurate at detecting the medical condition.

Keywords: Federated Learning; Generative Adversarial Network (GAN); Augmentation; Classification; Alzheimer's Disease

Acknowledgement

First and foremost, all praise to the Almighty for whom our thesis has been completed fluidly without any interruptions.

We would like to express our profound gratitude to our Supervisor, Dr. Md. Golam Rabiul Alam, for his kind supervision and endless support throughout our work.

We thank our Co-Supervisor, Mr. Saadat Hasan Khan, for his commitment and dedication in our work.

And finally, we are grateful to our parents, whose encouragement and prayers has been crucial towards our work.

Table of Contents

Declaration i

Approval ii

Abstract iv

Acknowledgment v

Table of Contents vi

List of Figures viii

List of Tables 1

1 Introduction 2
- 1.1 Research Problem . . . 3
- 1.2 Research Objectives . . . 5

2 Literature Review 6
- 2.1 Federated Learning . . . 6
- 2.2 Generative Adversarial Networks (GANs) . . . 6
- 2.3 Related Works . . . 6

3 Methodology 11
- 3.1 Dataset Details . . . 13
- 3.2 Data Pre-processing . . . 15

4 Model Implementation 16
- 4.1 GAN Model . . . 16
  - 4.1.1 Generator Model . . . 17
  - 4.1.2 Discriminator Model . . . 18
- 4.2 Disesase Detection Models . . . 19
  - 4.2.1 VGG16 . . . 19
  - 4.2.2 EfficientNetB6 . . . 20
  - 4.2.3 Xception . . . 21
- 4.3 Federated Averaging Algorithm (FedAvg) . . . 22
- 4.4 Optimizer . . . 23
  - 4.4.1 Adaptive Moment Estimation (Adam) . . . 23
- 4.5 Activation Functions . . . 24
  - 4.5.1 Leaky Rectified Linear Unit (LeakyReLU) . . . 24

4.5.2 Softmax . . . 25
4.5.3 Sigmoid . . . 25
4.6 Loss Functions . . . 26
4.6.1 Binary Cross-Entropy . . . 26
4.6.2 Categorical Cross-Entropy . . . 26

5 Results & Analysis **27**
5.1 GANs . . . 32
5.1.1 Non Demented Class . . . 32
5.1.2 Very Mild Demented Class . . . 33
5.1.3 Mild Demented Class . . . 34
5.1.4 Moderate Demented Class . . . 35
5.2 Pre-FED Disease Detection . . . 36
5.2.1 VGG16 Outcome . . . 36
5.2.2 EfficientNetB6 Outcome . . . 40
5.2.3 Xception Outcome . . . 43
5.3 Post-FED Disease Detection . . . 47
5.3.1 VGG16 Outcome . . . 47
5.3.2 Xception Outcome . . . 51
5.4 Grad-CAM . . . 55
5.5 Classification Reports & Comparisons . . . 56

6 Conclusion **57**

Bibliography **61**

List of Figures

3.1 Flow chart of the proposed Federated GAN Based Biomedical Image Augmentation and Detection Model for Alzheimer's Disease 11
3.2 Pie Chart of Train Dataset 13
3.3 Pie Chart of Test Dataset 14
3.4 Data Pre-processing 15

4.1 Core GAN Model Architecture 16
4.2 Generator Model Architecture 17
4.3 Discriminator Model Architecture 18
4.4 VGG16 Model Architecture 19
4.5 EfficientNetB6 Model Architecture 20
4.6 Xception Model Architecture 21
4.7 FedAvg Algorithm 22
4.8 ReLU and LeakyReLU Activation Function 24

5.1 Epoch 46 (Non Demented) 28
5.2 Epoch 50 (Very Mild Demented) 29
5.3 Epoch 100 (Mild Demented) 30
5.4 Epoch 2000 (Moderate Demented) 31
5.5 Bar chart of real vs. fake detection rate progression in Non Demented Class 32
5.6 Bar chart of real vs. fake detection rate progression in Very Mild Demented Class 33
5.7 Bar chart of real vs. fake detection rate progression in Mild Demented Class 34
5.8 Bar chart of real vs. fake detection rate progression in Moderate Demented Class 35
5.9 Line graph of VGG16 accuracy (Pre-FED) 36
5.10 Line graph of VGG16 loss (Pre-FED) 37
5.11 Confusion matrix of VGG16 (Pre-FED) - Testing on real images' dataset 38
5.12 Confusion matrix of VGG16 (Pre-FED) - Testing on mixed images' dataset 39
5.13 Line graph of EfficientNetB6 accuracy (Pre-FED) 40
5.14 Line graph of EfficientNetB6 loss (Pre-FED) 41
5.15 Confusion matrix of EfficientNetB6 (Pre-FED) 42
5.16 Line graph of Xception accuracy (Pre-FED) 43
5.17 Line graph of Xception loss (Pre-FED) 44

5.18 Confusion matrix of Xception (Pre-FED) - Testing on real images' dataset 45
5.19 Confusion matrix of Xception (Pre-FED) - Testing on mixed images' dataset 46
5.20 Line graph of VGG16 accuracy (Post-FED) 47
5.21 Line graph of VGG16 accuracy (Post-FED) 48
5.22 Confusion matrix of VGG16 (Post-FED) - Testing on real images' dataset 49
5.23 Confusion matrix of VGG16 (Post-FED) - Testing on mixed images' dataset 50
5.24 Line graph of Xception accuracy (Post-FED) 51
5.25 Line graph of Xception loss (Post-FED) 52
5.26 Confusion matrix of Xception (Post-FED) - Testing on real images' dataset 53
5.27 Confusion matrix of Xception (Post-FED) - Testing on mixed images' dataset 54
5.28 Grad-CAM Visualization of VGG16 Model 55

List of Tables

5.1 Classification Report of VGG16, EfficientNetB6 & Xception (Pre-FED) 56
5.2 Classification Report of VGG16 & Xception (Post-FED) 56

Chapter 1

Introduction

In the medical imaging domain, dealing with smaller datasets and fewer annotated samples [1]–[5] is arduous since supervised Machine Learning techniques need large labeled datasets. Annotations are made by radiologists with a specialized understanding of the data and tasks in medical imaging tasks. Most medical image annotations take a significant amount of time. Public medical datasets are accessible online, although the volume of databases are still inadequate and only relevant to a limited number of medical conditions. Patients, physicians, researchers, and radiologists are all involved in the process of collecting medical data, which is both time-consuming and costly[3]. As a result, deficiency of sufficient data towards further advanced machine learning models arise.

We believe data augmentation is the ideal way to address the deficiency of medical data. Networks may be trained more effectively with the use of conventional data augmentation techniques[6]. The most common data augmentation methods are translation, rotation, flipping, and scaling of dataset pictures. Using high-quality examples, synthetic data augmentation is an advanced kind of data augmentation. It is possible to learn more variability and enrich the dataset by utilizing synthetic data samples generated by a generative model [7].

GANs (Generative Adversarial Networks) are a deep-learning-based generative model. They are employed in a method of unsupervised learning. There are unsupervised models that may be utilized for creating new examples in the distribution of inputs. While there is no output variable in this model, there are input variables (X) and samples in the data (Y). Generative models employ just training data to identify patterns in the input variables and then produce an unknown output from the training data. In GANs, two neural networks compete with each other to construct or generate data variations. The Generator Model and the Discriminator Model are two sub-models of the Generative Adversarial Networks' architecture.

The entire procedure can be encapsulated in the following mathematical formula,

$$V(D,G) = E_{x\sim Pdata(x)}[\log D(x)] + E_{z\sim Pz(z)}[\log 1 - D(G(z))] \tag{1.1}$$

Discriminator Network D(x) and Generator Network G(z) indicate the networks of generators and discriminators, respectively. The distribution of actual data is represented by Pdata(x), the distribution of generator data by Pdata(z), samples from

real data are represented by x and samples from generator data are represented by y.

The GAN framework has recently been used in a number of medical imaging applications[8]–[13]. The GAN paradigm for image to image translation has been used for translations from label to segmentation, segmentation to image translation, and translation across medical modalities. To learn retinal vascular segmentation images, Costa et al. [8] developed a fully convolutional network. Then they figured out how to translate a binary vessel tree into a new retinal image. A GAN was trained by Dai et al. [9] to build lung field and heart segmentation pictures from chest X-ray images. Xue et al. [10] termed the two GAN networks as Segmentor and Critic, and they learned the translation between brain MRI pictures and a binary segmentation map of the brain tumor. Nie et al. [11] trained a patch-based GAN to convert between brain CT images and the correlating MRI images. They also proposed an image refinement approach based on an auto-context model. Ben-Cohen et al. [14] also used GAN to generate cross-modality images, from an abdominal CT scan to a PET scan image highlighting liver abnormality. The GAN approach for image inpainting has sparked some research. Schlegl et al. [12] used healthy patches of retinal tissue to train GAN to understand the data distribution of healthy tissue. The GAN was then evaluated for anomaly detection in retinal pictures on patches of both undetected healthy and anomalous data.

1.1 Research Problem

Due to rationale like Patient-Doctor confidentiality, which prevents the doctors and institutions from sharing medical data obtained from the patient, creates a huge scarcity of publicly available medical data. We investigated strategies for synthetic data augmentation to grow our medical dataset due to the challenge of limited data in the medical imaging area.

Data security and privacy are always an issue with classical machine learning techniques. Data confidentiality is required for data privacy and security, as privacy cannot be ensured if data are vulnerable to unauthorized access. Existing machine learning algorithms solutions cannot afford to be secure.

Traditional machine learning algorithms are carried out in a centralized data center, where data owners upload their data; as a consequence, data is private and owners are unwilling to share [15].

Furthermore, data collecting is a time-consuming and difficult process that is essential for machine learning advancement. Individuals are increasingly using machine learning as a commodity service. The sensitive information contained in the training set will be revealed if machine learning techniques provided by untrustworthy actors are used blindly [16].

In the field of biomedicine, for example, knowing that a patient's clinical record was used to train a disease-related model can imply the patient has the disease [17]. However, in ML, a centralized data center can expose clients' data to attackers, pos-

ing a significant danger of data privacy. An attacker can also use the collaborative learning method to recreate sensitive data from the client's device. Furthermore, the attacker has the ability to influence the learning process and retrieve information from the client's gradients. As machine learning trains the model to connect through a central server, attackers find it easier to penetrate and abuse the data, since their goals and tactics have broadened and continue to do so.

To overcome the aforementioned problem, we propose the Federated Learning approach. Federated Learning has shown to be a promising paradigm for keeping client data private and secure. Federated Learning is a fundamental concept that allows machine learning models to be developed utilizing data sets distributed across several devices while ensuring a secure environment for the data. FL allows many people to work together on training a machine learning model without having to share local data. One of the essential characteristics of federated learning is data security. Smarter models, reduced latency and less battery usage are all possible thanks to Federated Learning.

This technology makes it possible for portable devices, such as smartphones, to work together on developing a common prediction model while still retaining all of the training data locally. Moves model teaching to the edge, incorporating devices such as cellphones, laptops, IoT, and even organizations - hospitals, clinics, diagnostic centers, which must adhere to strict privacy requirements. Keeping personal data local has a huge security benefit. Real-time prediction is possible since prediction takes place on the system itself. FL reduces the time lag generated by transmitting raw data to a central server and then returning the results to the system. Since the models are saved on the device, the prediction approach works even if there is no internet connection. FL reduces the amount of hardware equipment available. The hardware requirements for FL versions are minimal, and what is available on mobile devices is more than adequate.

We are using Generative Adversarial Networks to create high-quality photographs (GANs). GANs are known as adversarial where these architectures set two neural networks against one another to generate new, fictitious data that may be used to simulate actual data. In the GAN model design, there are two sub-models: a generator model that brings new instances and a discriminator model that evaluates whether the created examples are genuine or fraudulent. Thus, we can generate an accurate representation of test data and solve the data scarcity problem, enabling us to train our model more efficiently.

As a result, the research is attempting to address the following questions:
How accurately can we augment MRI of the brain using GANs and how effectively can we detect Alzheimer's disease while ensuring patients' privacy through Federated Learning?

1.2 Research Objectives

This research aims to develop a GAN based image augmentation system, and a Federated Learning model to detect Alzheimer's disease, preserving the privacy of the patients, which can be detected from a dataset where the dataset can be distributed and train those distributed datasets in a decentralized way, in multiple separate client servers.

The Objectives of this research are:

1. To deeply understand GANs and how these work.
2. To extensively explore Federated Learning and how it works.
3. To vastly increase the confidentiality of critical clinical data given by clients.
4. To test train datasets on distinct devices remotely in order to improve efficiency.
5. To evaluate the model.
6. To detect Alzheimer's disease with an accuracy of over 90%.

Chapter 2

Literature Review

2.1 Federated Learning

Federated learning trains a Neural Network (NN) (other ML) model on edge devices, which is subsequently delivered back to the Machine Learning Model Owner (MLMO), such as a server. The server compiles these models into a single global model and delivers it back. This approach is repeated until the global NN model reaches a certain level of accuracy. Any transaction and model data in FL is stored on the MLMO server.

2.2 Generative Adversarial Networks (GANs)

GANs, or Generative Adversarial Networks, are generative models based on deep learning. GANs are a model architecture for training a generative model in general, and deep learning models are most commonly used in this architecture. Since GANs are unsupervised, they do not require labeled data to be trained. GANs are currently the sharpest picture generators. This is made possible by adversarial training.

2.3 Related Works

In the field of image processing, image augmentation is frequently used to minimize overfitting on the training dataset and to increase prediction accuracy on the testing dataset.

In a prior study named "The Effectiveness of Image Augmentation in Deep Learning Networks for Detecting COVID-19: A Geometric Transformation Perspective", geometric augmentations were used to compare the performance of 17 deep learning algorithms with and without geometric optimizations on COVID-19 identification.

Data augmentation should be employed to strengthen the model's robustness and generalizability, according to the authors [18]. Furthermore, they compared DarkNet-19 which was also empirical rather than therapeutically justified. Many datasets

were used to assess the 17 deep learning algorithms' efficiency (MCC).

Using the same geometrical augmentations in all situations is not efficient, however. The authors used an example of the comparison between a dog in a photograph and an X-ray image was used to identify COVID using segmentation which did not provide identical accurate results. It is important to note that the augmentation stage is not a random technique that can be applied to all study areas, but rather a domain-dependent procedure.

According to [19], In contrast to normal picture augmentation, biomedical image augmentation has a unique set of properties. Time-series-based or z-stack/layered picture data are common, as are data in intricate forms like txt or csv. A lack of consideration for biomedical use cases makes it difficult to employ non-biomedical software in these unusual cases. The Augmentor package, a stochastic pipeline-based technique, was utilized to solve this issue. An initial lesion scan was supplemented with numerous masks, such as the segmentation mask and the pigment mask during this augmentation.

However, image augmentation can be time-consuming. Fast and flexible picture alterations for computer vision applications, such as classification, segmentation, or detection, were suggested in the study [20]. The authors explained how GPU processing exceeds CPU processing with parallel processing. The augmentation library for biomedical images has complex target support such as image and mask target support, bounding box support, keypoint support, and multiple targets. With Albumentations they processed 1.2x to 52x more images per second compared to other image augmentation libraries.

In the case of Federated Learning, according to [14], each client device contains a local training dataset. For privacy reasons, the dataset is never uploaded to the main server. The client device computes its own data using a global model and sends updates to the server. The researchers' main contributions are identification of decentralized data received from mobile devices, selecting a straightforward and practical algorithm, and evaluation. In their approach, termed 'FederatedAveraging,' they suggest combining local SGD on each client with a server that conducts model averaging. For federated optimization, they emphasized the non-IID and unbalanced properties. The researchers trained in a controlled environment with K clients each with their own dataset and selected a random fraction of them. The data collection was done when a client was charged, on unmetered connections. They set a clock time for synchronous SGD optimization. With FedAvg they achieved 85% accuracy on image classification and 10.5% accuracy for the LSTM model.

The paper [21] highlighted the importance of FL using mobile devices. The authors set out to find a solution to the situation of not optimizing for a global model on a central server and instead disregarding each client. The goal of this paper's training was to provide a mobile keyboard suggestion and a word-level language modeling challenge. For the main paper [14], they presented an attention method for model aggregation. When it came to learning a differentially private client model, the researchers used both the standard GRU model and a randomized learning process.

To achieve attentive federated aggregation (FedAtt), they combined the layer-wise contributions of chosen clients' neural language models to the central server and they improved the trained models' accuracy by 77-88 percent using their method and Treebank [21] and WikiText-2 [22] datasets on the client GRU model.

"Group Knowledge Transfer: Federated Learning of Large CNNs at the Edge" authors Chaoyang He, Murali Annavaram, and Salman Aveshe believe that scaling up a convolutional neural network is beneficial to model accuracy, but it also introduces difficulties such as high model size, which makes training on resource-constrained edge devices challenging. In this study, the authors reformulate FL to FedGKT, a group knowledge transfer training approach. FedGKT is meant to quickly train tiny CNNs on the edge and regularly transmit their information to a server-side CNN with a large capacity via knowledge distillation. Their findings indicate that FedGKT can achieve an accuracy level comparable to, if not slightly greater than, FedAvg.

"FedCV: A Federated Learning Framework for Various Computer Vision Tasks" by Chaoyang He and colleagues shown that federated learning has the ability to rescue a number of intriguing computer vision applications that centralized training can't handle owing to a variety of difficulties, including privacy concerns, data transport and maintenance expenses [23], [24].

Additionally, the research gap between computer vision (CV) and federated learning (FL) is enormous, which results in model performance in FL being significantly lower than that of centralized training.

FedCV, a federated learning library they developed as part of their study, integrates numerous FL algorithms to various essential CV tasks, such as picture segmentation and object recognition, and enables the framework to be flexible in exploring techniques utilizing new distributed computing protocols, such as customizing information transmission between clients and establishing specific training processes. In spite of this, they employed CIFAR-100, GLD-23K PASCAL VOC and COCO datasets, as well as models such as EfficientNet and MobileNet, as well as Deeplab V3+, UNet, YOLOv5, FedAvg.

However, debugging an FL system remains a significant challenge as there is no access to a centralized dataset that we can check for which input results are giving unexpected results. In the paper [25], the authors identified challenges such as sanity checking and model debugging, data labeling, and detecting bias in training data that are not possible in a decentralized dataset. Non-inspectable data might be used in federated, privacy-preserving modeling approaches. They recommended testing for 'out of vocabulary (OOV) spikes as a way to troubleshoot the language model. Two DP Federated GANs for Generating Image Data were suggested by the authors to debug the image classifier. The federated GANs generate privacy-preserving samples that detect the nature of a defect in on-device picture preprocessing after they are trained. With these approaches, the authors presented debug solutions for FL systems without violating client privacy rights.

Despite the ease of accessing some medical picture datasets, the list is still too small to support contemporary deep learning and machine learning models. The most major issue in medical imaging is dealing with little datasets and annotated samples [1]–[5]. Concerns about privacy, security, lack of appropriate equipment, lack of engagement with medical specialists, etc., all impede the availability of data and the evolution of models. Obtaining such data in the medical field is tough, says (Frid-Adar et al., 2018)[21]. As a result of their study, they were able to demonstrate that the use of deep learning Generative Adversarial Networks (GANs) to create artificial medical images may complement real-world data and boost CNN performance for medical image classification by ten to twelve. The authors' approach also illuminated the field of employing traditional data augmentation to assist the training process of neural networks, which is a well-established method in computer vision issues. Synthetic data examples learned by generative models expand and diversify the dataset, therefore boosting system training. There are two types of augmentation illustrated in the paper: conventional augmentation (image modifications) and generative models (synthesis of new instances from data examples). Besides, some of the more papers related to the segmentation of images including Costa et al. [8], Dai et al. [9], Nie et al. [10], Ben-Cohen et arXiv:1803.01229v1 [cs.CV] 3 Mar 2018 2 al. [12], Schlegl et al. [11] show that GAN is one of the most famous and widely used networks for image generation and further advances the existing models.

One of the limitations of this paper was there were no 2D to 3D conversions in input volumes, but overall, their objective of improving the training results via augmentation was quite successful.

Looking forward to disease detection, for diabetes-based eye disease detection the paper [26] discussed the need for a fully automated system. Using the FRCNN algorithm with fuzzy k-means, they employed an automated disease localization and segmentation strategy based on the FRCNN algorithm with fuzzy k-means (FKM). They used handmade characteristics to distinguish between diseased and healthy areas of photographs in order to automate the identification of eye disorders. However, because of color, size, and increased intra-class differences, these characteristics could not accurately depict the DR, DME, and glaucoma areas. Their deep learning approach detects bounding boxes of disease regions. They sub-categorized the localization phase into locating the DR region, DME region, and Glaucoma Region. Finally, the authors had around 94.5% accuracy on ORIGA< HRF and DR HAGIS datasets.

To detect Covid-19, the paper [27] extracted features from CT Scan images and used these to train them against Pneumonia and other pulmonary diseases, and finally used Grad-CAM technique having plotted class activation maps. While doing detection or prediction tasks, the Grad-CAM technique provides a visual description for any deeply linked neural network and aids in understanding more about the model.

Han et al. (2018)[28] claim that in order to calculate the millions of parameters in a deep CNN, a large number of annotated samples are required, which presently prevents many improved deep CNNs from being used in circumstances with sparse training data.

The researchers used a two-step technique, combining CNN transfer learning with online data augmentation. CNN has many major limitations, including the need for a large number of labeled training samples for weight parameter learning and the need for a good PC with adequate CPU power to expedite the learning process.

Nowadays, there is a problem in adapting deep CNNs to small datasets while maintaining comparable performance to large-scale datasets. Additionally, the web data augmentation component of this study was derived from classic data augmentation techniques such as rotation, translation, zoom, flips and shears as well as color disruption.

The demand for a huge amount of training data is eliminated, and a robust and strong classifier is developed. Through web data augmentation, significant variety is given to the training set, enhancing the fine-tuned network's generalization capacity and further reducing overfitting. One downside of their technology was the lengthy augmentation process and moreover, GANs are an extremely successful way for creating small-size, high-quality fake data, not efficient for large-sized data.

However, as most algorithms for prostate ultrasound image segmentation do not provide pixel-level separation, today's segmentation method is not accurate enough to accurately position a biopsy needle, imposing additional stress and time demands on the health care professional.(Liu et al., 2021) As a consequence, the authors of [29] devised a system for prostate ultrasound picture segmentation that includes three modules: feature extraction, detection, and segmentation based on the enhanced Mask R-CNN. Classification is another critical component of the prostate ultrasound picture system. Imaging techniques such as segmentation and classification have been used to classify the ultrasound image of a prostate to identify whether or not cancer is present. A feature extractor and classifier's performance is heavily reliant on the shape, texture, color, and underlying visual qualities of a picture to be classified traditionally. The results showed that the data set on producing ultrasound images of prostate disease increased the rate of detection.

Moreover, the research [30] discussed the harmful effects and disadvantages of mammography in breast cancer detection. The authors also discussed the limitations of using CT and MRI due to low sensitivity for subcentimeter lesions for their low spatial resolution. In contrast, they used thermal imaging to predict breast tumor location. In the paper natural heat transfer equation and near-infrared fluorescent and agent for imaging. Furthermore, they identified BRCA1 and BRCA2 genes present in most breast cancer patients. For the images, the authors used 20 sequential thermal images with 15s intervals and cropped them. The main model in this paper, CAD, is a deep neural network (pre-trained Inception V3 model) with the SVM model as a classifier.

Chapter 3

Methodology

The main purpose of Federated GAN-based biomedical image augmentation and classification for Alzheimer's disease is to detect medical conditions by addressing the data scarcity problem and ensuring the safety of client data. Multiple users can train a machine learning model using FL without having to share local data. Federated learning addresses data security. The utility of data can be kept even when it is stored locally thanks to federated learning.

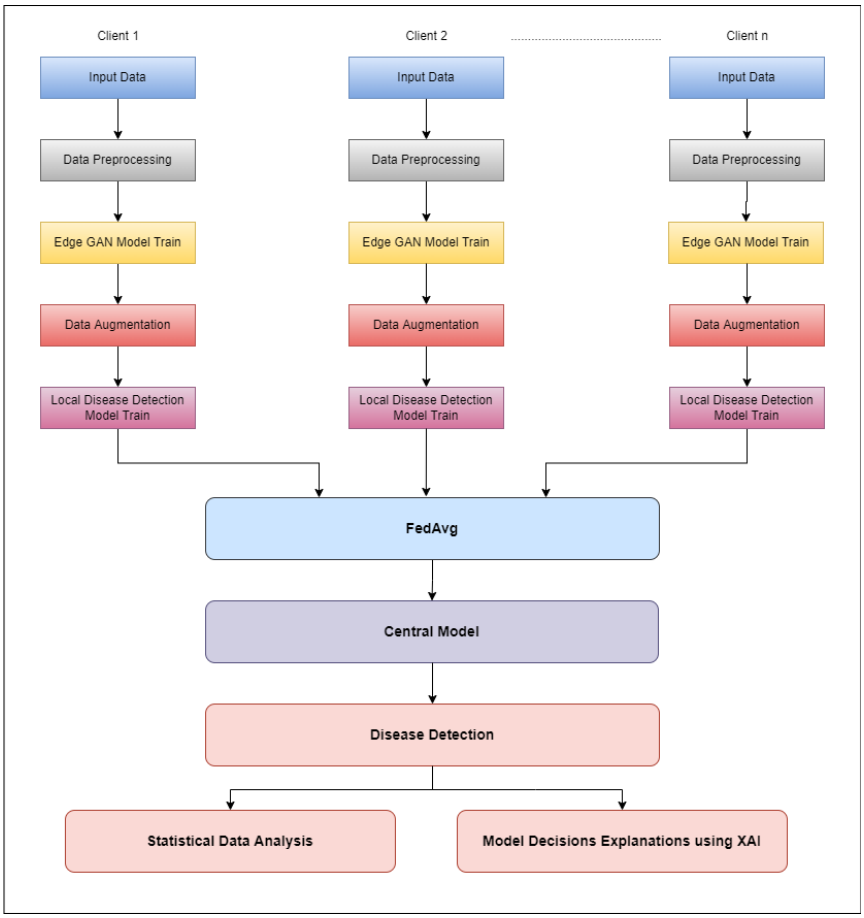

Figure 3.1: Flow chart of the proposed Federated GAN Based Biomedical Image Augmentation and Detection Model for Alzheimer's Disease

Our overall workflow significantly depends on multiple major stages including taking input data from clients, data preprocessing, and training the GAN model on edge devices for augmenting images using the parameters received from the edge devices to the central server implying FedAvg on aggregating global parameters.

1. Preprocessing of input data: This stage involves validating and preparing the input data in order to facilitate the processing, testing, and training models.

2. GAN Model training on Edge Devices: This step is responsible for the processing, generating, and differentiating input medical images using the GAN model and forwarding the parameters of this stage to the global server aggregating the parameters using FedAvg.

3. Fed Averaging: This stage serves as the central server for updating the parameters obtained from the edge devices using the Fed Avg method.

3.1 Dataset Details

The dataset we have chosen for this has been collected from [31] and contains a total ∼6400 images across the training and testing portion.

The training and validation part combined contains a total ∼5760 images, consisting of four classes of brain Magnetic Resonance Imaging (MRI) images. The classes are, (i) Very Mild Demented, (ii) Mild Demented, (iii) Moderate Demented, and (iv) Non Demented. MRI of the brain, in this case produces high-quality two-dimensional pictures of the brain and brainstem stating demented status of the brain, without using ionized radiation or radioactive tracers.

The train and validation portion of the dataset contains 2016 image files for Very Mild Demented, 806 image files for Mild Demented, 52 image files for Moderate Demented, and 2880 image files for Non-Demented classes.

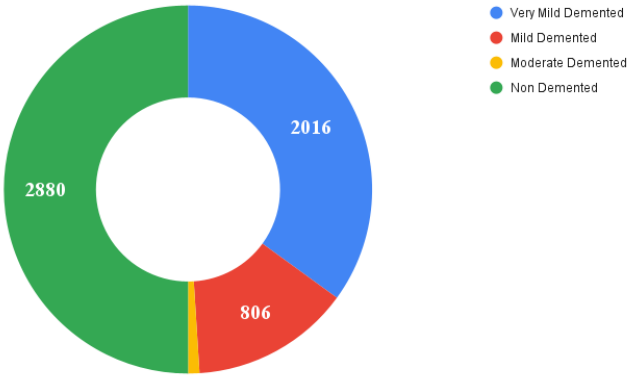

Figure 3.2: Pie Chart of Train Dataset

On the other hand, the testing part contains a total ∼640 images, containing 224 image files for Very Mild Demented, 90 image files for Mild Demented, 6 image files for Moderate Demented, and 320 image files Non-Demented and classes.

Furthermore, the dimensions of both train and test image data are 176x208, and the sizes of these images are 5 kilobytes.

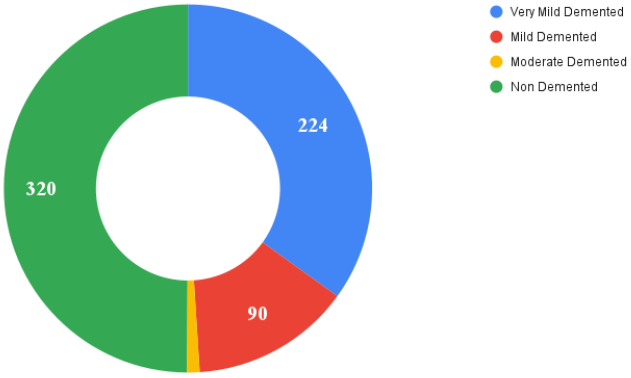

Figure 3.3: Pie Chart of Test Dataset

Here, we observe that the Moderate Demented class of our dataset has significantly less amount of data compared to the rest of the class data in our dataset.

3.2 Data Pre-processing

For machine learning models to work effectively, we must first convert raw data into a format that the models can process which is called data preprocessing. Real-world data, such as text, images, and video, is inherently nuanced. Machine Learning models may not be able to analyze it accurately because of its inconsistencies and inaccuracies, as well as the fact that it is often incomplete and has an established design. Since it needs the necessary properties to be used by Machine Learning models, this is a difficult crucial step.

It is clearly noticeable that the Moderate Demented class does not have sufficient amount of data. To work with this, we have applied 15 degree rotation for augmentation using keras ImageDataGenerator which is an in-place or on the fly image data augmenter. The process of augmenting begins with accepting a batch of training images, after which each image in the batch is subjected to a series of random transformations (such as random rotation, resizing, shearing, etc.), in this case random 15-degree rotations. Additionally, it replaces the initial batch with the new batch that has been randomly altered, allowing us to continue training our GANs model. We trained it for 1965 epochs and managed to overcome the lack of data situation for Moderate Demented class.

Initially, we used python's OpenCV, an open-source computer vision library. Considering our dataset consists of images of various different dimensions and colors, we have changed the color grading in our images into grayscale, to bring a similarity to our dataset. Additionally, we have changed the image dimension to 128x128 pixels, from 176x208, to make the training more efficient.

Furthermore, since all the labels of our dataset were in string values, we have used One-Hot encoding to represent all the labels numerically, and to make the training data more convenient and eloquent. Moreover, it will help us to rescale the data smoother.

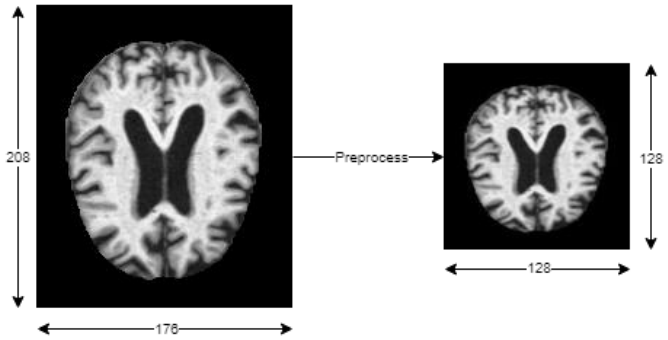

Figure 3.4: Data Pre-processing

Chapter 4

Model Implementation

4.1 GAN Model

Generative Adversarial Networks (GAN) is a generative framework for deep learning which is frequently used in image, video, and voice generation following an adversarial process. It is essentially an architecture that uses the assistance of two neural networks to produce new and false data instances that may pass for actual data. GAN was first introduced in 2014 by Ian Goodfellow and others at the University of Montreal [32].

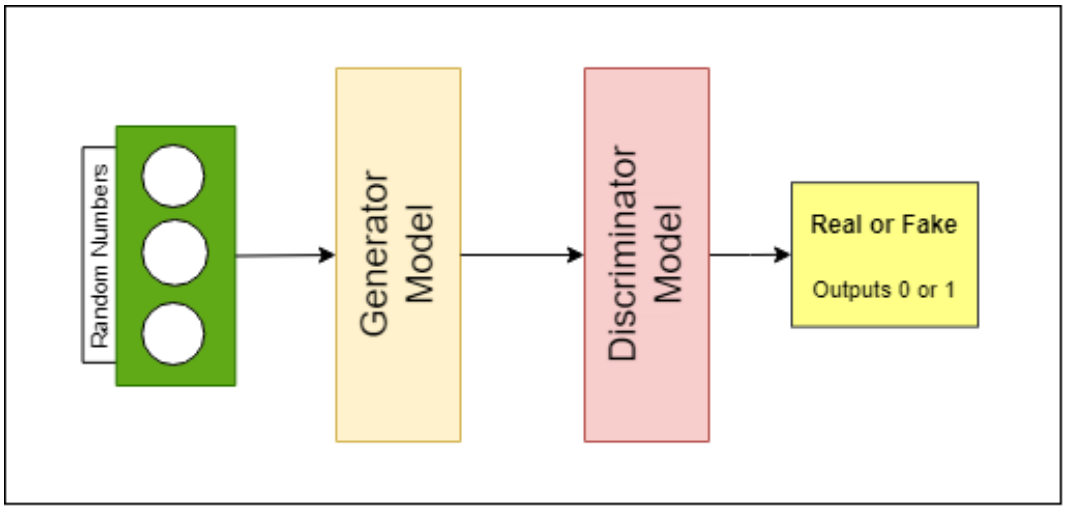

Figure 4.1: Core GAN Model Architecture

Initially, both our Generative Model and Discriminator Model contain precisely one input tensor and one output tensor, thus we begin with a sequential model. Adam, an optimizer, is also being used with a learning rate of 0.0002 and an initial decay rate of 0.5. The model is then compiled with the assistance of a binary cross-entropy loss function. This model takes random numbers as the input tensor and outputs 0/1 representing whether the image generated by the generator model is real or fake.

4.1.1 Generator Model

The generator model is the most important component of the core GAN model since it produces fictitious data using discriminator feedback. This section of the main GAN model seeks to convince the discriminator to categorize its output as real.

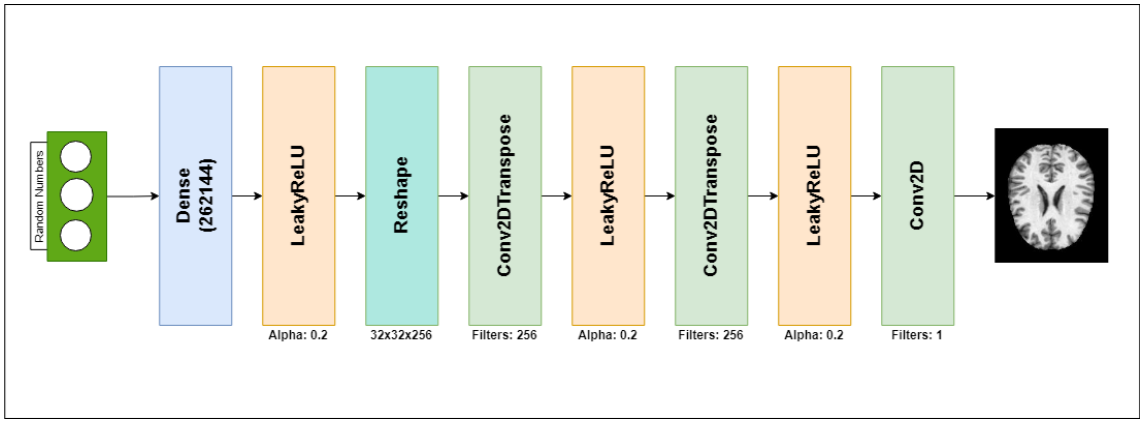

Figure 4.2: Generator Model Architecture

In our situation, we used a Dense layer with $32 * 32 * 256 = 262144$ units as the generator model's input layer. This layer receives inputs from the random numbers and passes a vector to the Reshape layer for reshaping. Furthermore, the output from the Reshape layer is upsampled twice in the Conv2DTranspose layer with 256 filters in order to get more dense, detailed, and precise information in the output picture. Finally, LeakyReLU activation layers with a negative slope coefficient (α) value of 0.2 are used between the layers to accelerate training by assisting the neural network to learn faster.

4.1.2 Discriminator Model

This portion of the core GAN model is nothing more than a classifier. Fundamentally, the objective of the discriminator model is to differentiate real images from the images generated by the generator model.

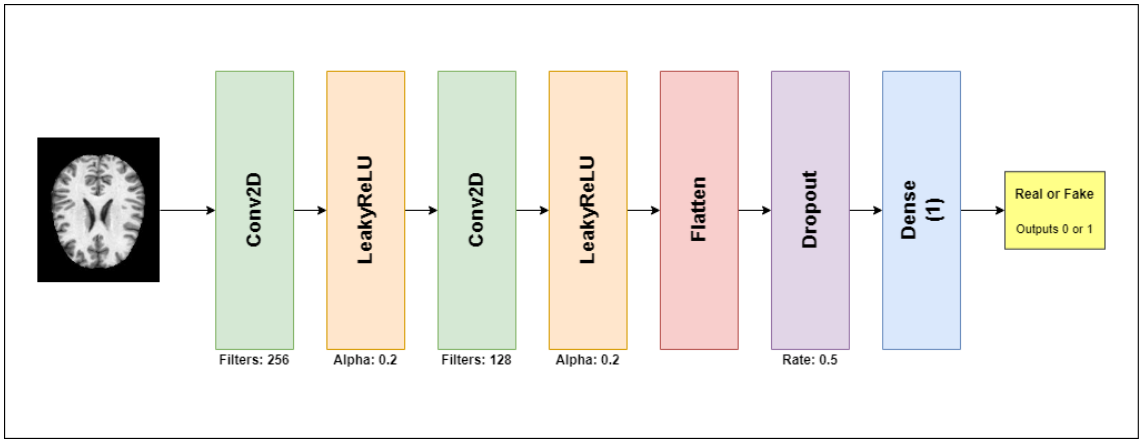

Figure 4.3: Discriminator Model Architecture

Initially, it receives the input in the Conv2D layer which adds a kernel over the 2D image input performing an element-wise multiplication. It will downsample the image to a single pixel and forward it to the next layer. After the Conv2D layers, we have utilized a flatten layer with the default argument to flatten the input. Subsequently, we are regularizing the input using the Dropout layer with a rate of 0.5 to eliminate the overfitting problem on the training data. In closing, we are using a fully connected layer, Dense with 1 unit which helps the model classify based on the outputs from the convolutional layers.

4.2 Disesase Detection Models

4.2.1 VGG16

VGG16 where VGG stands for Visual Geometry Group is a Convolutional Neural Network (CNN) model that opened the way for a number of key advances in the field of computer vision. The model was initially suggested in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2013 by Karen Simonyan and Andrew Zisserman [33]. In addition, VGG16 consists of 16 convolutional layers, some of which are followed by a pooling layer that decreases the height and width of pictures.

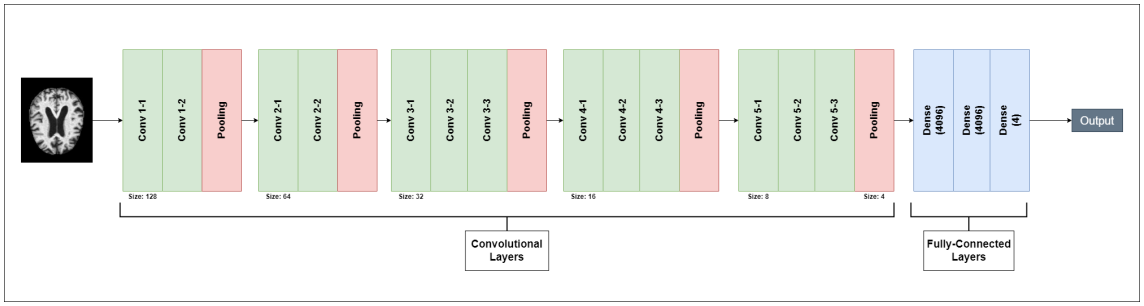

Figure 4.4: VGG16 Model Architecture

One of our disease detection models was implemented using a slightly different version of the VGG16 model architecture. To be precise, we have defined the image dimensions to 128x128x1 and are utilizing the weights of the pre-trained ImageNet model. Furthermore, the last Dense layer of the primary architecture has been replaced with another Dense layer with softmax activation and three-dimensional outer space.

4.2.2 EfficientNetB6

EfficientNet, which was initially shown in Tan and Le's 2019 paper, is one of the most efficient models that achieves staggering accuracy on ImageNet and popular image classification transfer learning tasks. It has 8 variants i.e. B0, B1, B2, B3, B4, B5, B6, and B7 where the input shapes vary from one variant to another.

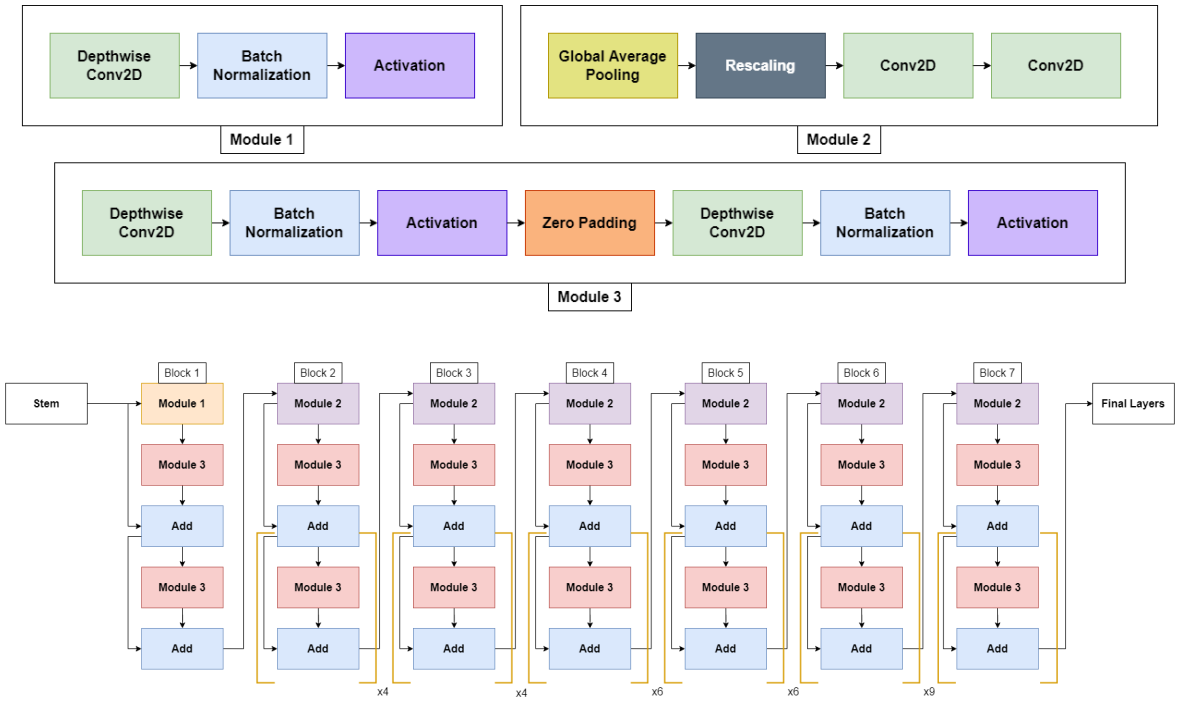

Figure 4.5: EfficientNetB6 Model Architecture

To detect the disease, we used a slightly modified version of the EfficientNetB6 model architecture. Similar to the VGG16 modification, the last Dense layer of the main architecture has been replaced with a new Dense layer that has softmax activation and three-dimensional outer space.

4.2.3 Xception

Extensive research has shown that Xception is a considerably better version than Inception-V3's. Researchers at Google came up with this Deepwise Separable Convolution neural network design. The Inception module with the most towers is Depthwise Separable Convolutions (DSC). The input first passes via the entrance flow, then the middle flow, and finally the exit flow.

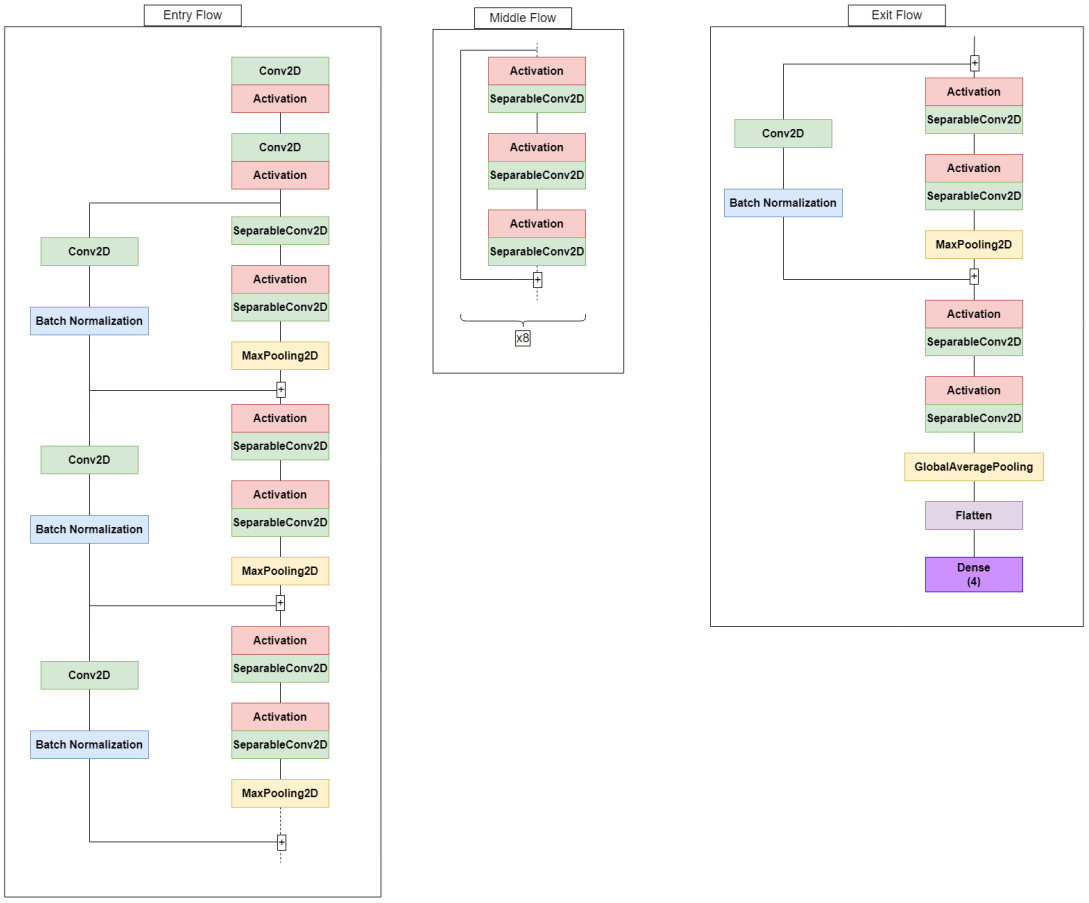

Figure 4.6: Xception Model Architecture

Nevertheless, the last fully connected layer of the main architecture of the Xception model has been replaced with a new Dense layer that has softmax activation and three-dimensional outer space to detect the disease in our case.

4.3 Federated Averaging Algorithm (FedAvg)

For distributed training systems with a large number of client-side devices, federated averaging (FedAvg) is one of the most used and communication-efficient approaches. One of the main inspirations objectives for our experiment was ensuring the privacy of the medical data. FedAvg helps us in this case with its privacy-protecting feature, which eventually requires clients to save their data locally.

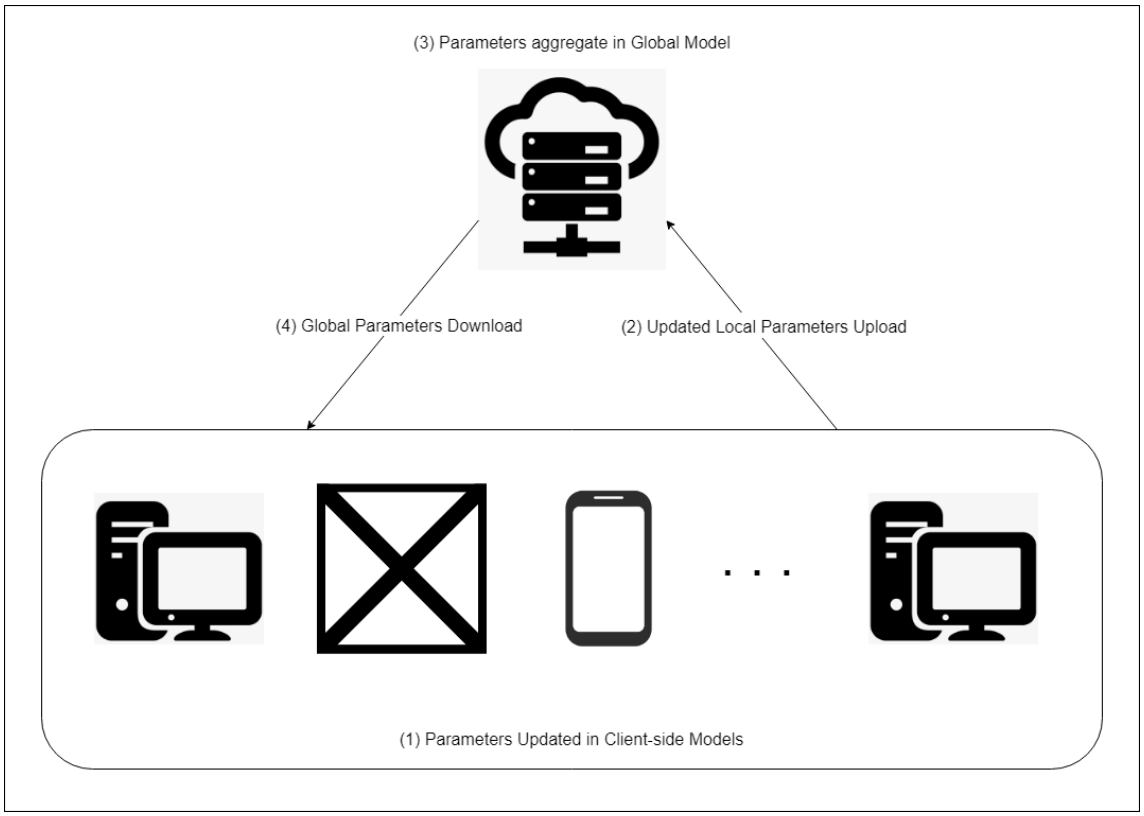

Figure 4.7: FedAvg Algorithm

To facilitate communication amongst client-side devices, a central parameter server is utilized from where each client receives the parameters from the central global server, which also aggregates the updated parameters from clients. However, it requires massive communication between the client-side devices and the global model in the central server. Also, any kind of attack on the central server can result in a breach in the databridge.

4.4 Optimizer

When it comes to obtaining a decent efficient performance out of a model, optimizers can be useful. Traditionally, they have been used to reduce the error generated by loss functions while also increasing the model's efficiency. These allow for the tweaking of a neural network's weights and learning rate in order to minimize losses and enhance efficiency. There are a number of optimizers that can help a model to increase its performance. Such as Gradient Descent, Stochastic Gradient Descent, Mini-Batch Gradient Descent, Adaptive Gradient Descent (AdaGrad), Root Mean Square Propagation (RMS-Prop), Adaptive Moment Estimation (Adam), etc.

4.4.1 Adaptive Moment Estimation (Adam)

Adam, or adaptive moment estimation, is a well-known, simple-to-implement optimizer that is based on stochastic gradient descent. It is also popular in the fields of computer vision and natural language processing. To be more specific, adam is an algorithm that allows for the computation of adaptive learning rates for each parameter, and it will be more feasible to apply in our scenario than RMS-Prop and AdaDelta since it is more computationally efficient, requires less memory, well-suited to problems with vast amounts of data and parameters, and able to achieve good results in a short time period. Moreover, the learning rate of the optimizer varies according to the initial decay rates given in β_1 and β_2. Hence, we have used Adam in the detections of the Alzheimer's disease model to optimize pre-trained VGG16 with a learning rate of $1e^{-}5$ and in the primary GAN model with a learning rate of 0.0002 in our experiment and with an initial decay rate (β_1) of 0.5.

4.5 Activation Functions

Activation functions are a type of function that aids the neural network in learning complicated patterns in input and determining the neural network's output. Explicitly, these functions are frequently found near the end of a layer and assist in determining which parameters' values should be forwarded to the next tier of layers. Furthermore, these functions improve a model's output, accuracy, and computational efficiency.

4.5.1 Leaky Rectified Linear Unit (LeakyReLU)

The ReLU, or rectified linear activation function, is a piecewise linear function that outputs the input directly if it is positive and zeros otherwise. ReLU and LeakyReLU can easily be differentiated and explained via a simple graph which are demonstrated below.

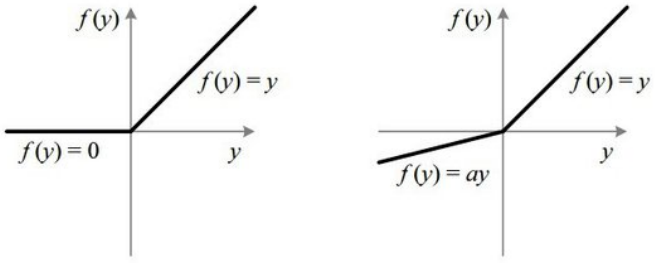

Figure 4.8: ReLU and LeakyReLU Activation Function

The Leaky Rectified Linear Unit, or LeakyReLU, is based entirely on ReLU, except instead of a flat slope for negative values, it has a tiny slope. While writing the core GAN model, we employed LeakyReLU activation layers numerous times in our discriminator and generator models with a negative slope coefficient (α) of 0.2 because they speed up training and alleviate the dead ReLU problem that may occur if we used ReLU instead of LeakyReLU.

4.5.2 Softmax

When dealing with multi-class classification, Softmax activation trumps sigmoid activation in terms of performance. It is not recommended to use the sigmoid function for multi-class classification issues since the probabilities do not take the probability of the other classes into account. The Softmax activation function follows the following equation:

$$softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}} \tag{4.1}$$

The z represents the data from the output layer's neurons, while the exponent acts as a nonlinear function. These numbers are then normalized and translated into probabilities. We have used the Softmax activation function in the dense layer of the disease detection model based on VGG16 architecture as there are more than two classes (non-demented, very mild demented, and mild demented) in our case.

4.5.3 Sigmoid

A mathematical function with a distinctive S-shaped curve is known as a Sigmoid function. All sigmoid functions including logistic function, hyperbolic function, and arctangent map the entire number line into a small range, of 0s and 1s. Sigmoid function has the following formula:

$$sigmoid(z) = \frac{1}{1 + e^{-z}} \tag{4.2}$$

To be precise, sigmoid function's applications of converting a real value into a number that can be interpreted as a probability is being used in our dense layer of discriminator model and in the Conv2D layer in the generator model of the core GAN model. We are using sigmoid as our activation function to normalize the medical images and keep the intensity of the images between 0$\sim$1.

4.6 Loss Functions

Loss functions are essentially the quantifier of the difference between the expected outcome and the outcome generated by any given model, from which gradients can be obtained to update the weights for the next layer. There are a lot of loss functions with different functionalities, i.e. mean squared error loss, mean absolute error loss, binary cross-entropy loss, hinge loss, squared hinge loss, multi-class cross-entropy loss, categorical cross-entropy loss, etc.

4.6.1 Binary Cross-Entropy

Binary Cross-Entropy, also known as Logloss is commonly utilized in the cases of binary classification. It compares each of the expected probabilities to the actual class outputs, which might be 0 or 1. Based on how far off the projected value each probability is, the score is then determined. In a nutshell, this is a negative average of the corrected predicted probabilities' logarithm. Binary Cross-Entropy follows the following formula:

$$loss = -\frac{1}{N}\sum_{i}^{N}\sum_{j}^{M} y_{ij}\log p_{ij} \tag{4.3}$$

Here, N is the number of rows, M is the number of classes in the classification problem, and p is the probability. We employed Binary Cross-Entropy in our discriminator model of the core GAN model as well as the core GAN model itself because both models categorize in binary.

4.6.2 Categorical Cross-Entropy

Unlike binary cross-entropy in binary classification, in multi-class classification problems, categorical cross-entropy is a well-known loss function. Besides, categorical cross-entropy is most useful when the labels are encoded in one-hot encoding. We have briefly discussed in the Data Pre-processing part that we are encoding our string labels using one-hot encoding to represent them numerically. Hence, we have used categorical cross-entropy as a loss function in the detection of Alzheimer's disease model on pre-trained VGG16.

Chapter 5

Results & Analysis

Upon processing our dataset, we have been successful to generate fictitious data using GANs. Once our data has been augmented, we fed a blend of our augmented data and real data to the VGG16 model and successfully achieved an accuracy of 98.42%, which is substantially higher than our initial accuracy objective. Our GAN model receives a random number in the input layer and generates new images of the MRI of the brain. Consequently, we input this image into our Discriminator model in order to convince it to categorize the generator's output as real. Through this discriminator model, we are identifying if our model can distinguish between real and generated MRI images.

The accuracy rate of the GAN model depends on the classification rate of the fake images as real and fake images as fake and it varies from epoch to epoch. Epochs containing a higher accuracy percentage of real and lower accuracy percentage of fake are presumed to be a good augmentation of our actual dataset, which demonstrates our generated images are accurate enough to be compared, or even confused with real images.

Examples of generated images with an acceptable rate of accuracy from the discriminator model for classifying generated images as real and fake for each class are given below:

1. Non Demented:
   The 46th epoch of the Non-Demented class has an accuracy rate of 94% for real and 19% for fake.

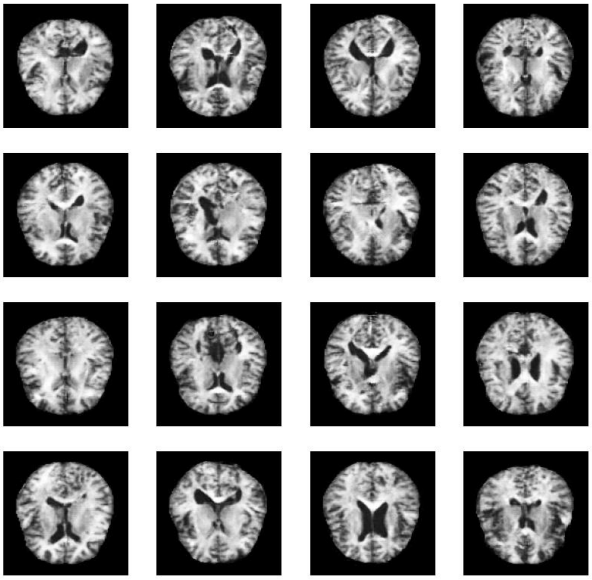

Figure 5.1: Epoch 46 (Non Demented)

2. Very Mild Demented:
   The 50th epoch of the Very Mild Demented class has an accuracy rate of 100% for real and 0% for fake.

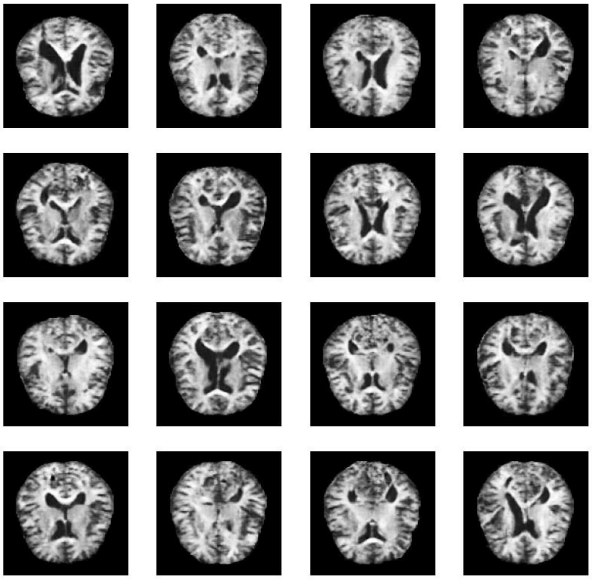

Figure 5.2: Epoch 50 (Very Mild Demented)

3. Mild Demented:
   100th epoch of the Mild Demented class have an accuracy rate of 94% for real and 31% for fake.

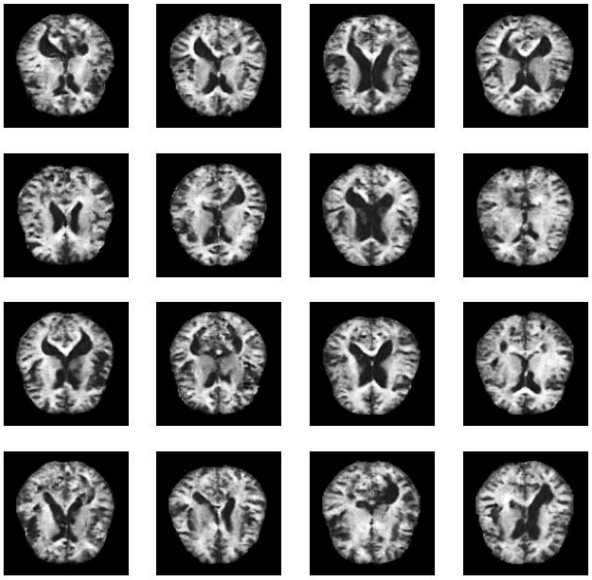

Figure 5.3: Epoch 100 (Mild Demented)

4. Moderate Demented:
   2000th epoch of the Moderate Demented class have an accuracy rate of 94% for real and 44% for fake.

Figure 5.4: Epoch 2000 (Moderate Demented)

5.1 GANs

5.1.1 Non Demented Class

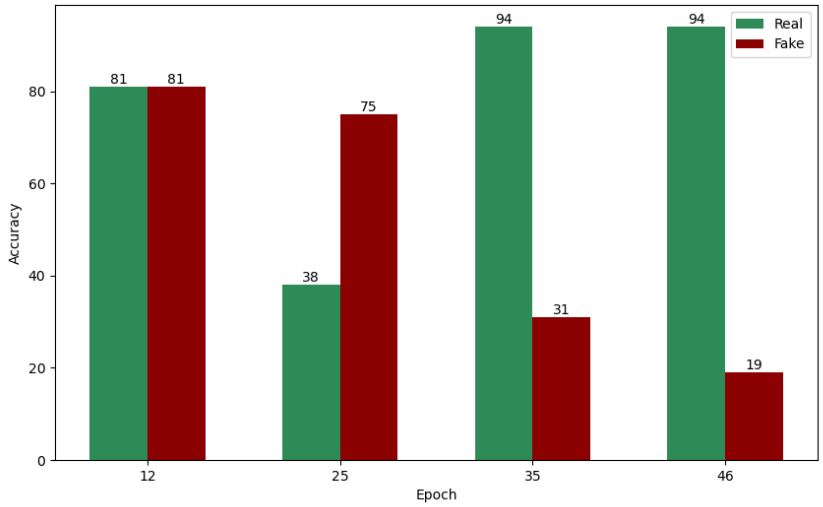

Figure 5.5: Bar chart of real vs. fake detection rate progression in Non Demented Class

Upon running the epoch 50 times in the Non-Demented class, we have generated 50 sets of images. In the first epoch, we get an accuracy rate of 100% for real and 56% for fake. Gradually, our accuracy increases, and we finally get to 94% real accuracy and 19% fake accuracy in the 46^{th} epoch. From the bar chart, we can clearly observe that in the latter epochs the accuracy of real is much higher than the previous epochs, similarly, the accuracy of fake is gradually decreasing compared to the previous epochs.

5.1.2 Very Mild Demented Class

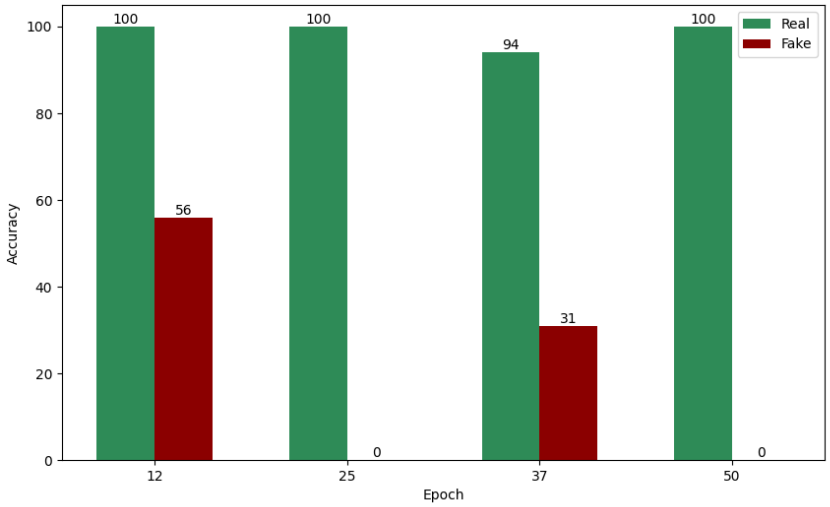

Figure 5.6: Bar chart of real vs. fake detection rate progression in Very Mild Demented Class

After running the epoch 50 times in the Very Mild Demented class, we have generated 50 sets of images. In the first epoch, we get an accuracy rate of 100% for real and 69% for fake. With each epoch, our accuracy increases, and we finally get to 100% real accuracy and 0% fake accuracy in the very last epoch. From the bar chart, we see that in the latter epochs the accuracy of real is much higher than the previous epochs, parallelly, the accuracy of fake is gradually decreasing compared to the previous epochs.

5.1.3 Mild Demented Class

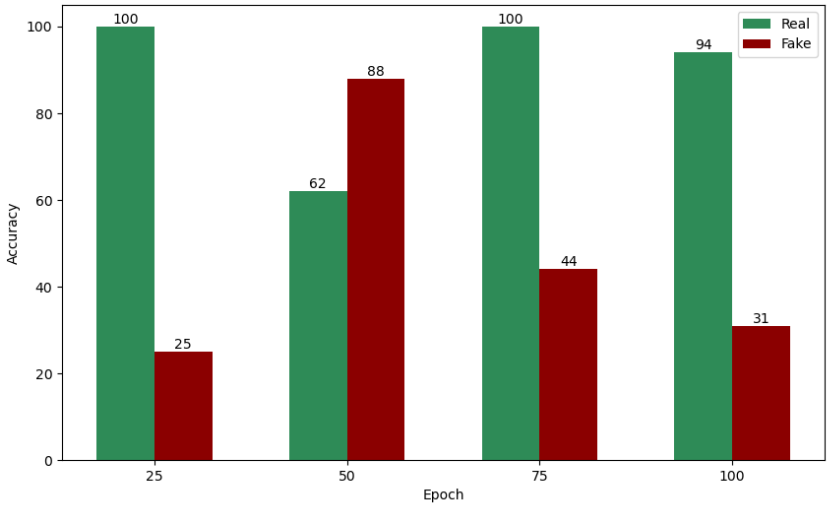

Figure 5.7: Bar chart of real vs. fake detection rate progression in Mild Demented Class

Once we have completed running the epoch 100 times in the Mild Demented class, we get 50 sets of generated images. In the first epoch, we get an accuracy rate of 100% for real and 88% for fake. Gradually, our accuracy increases, and we finally get to 94% real accuracy and 31% fake accuracy in the very last epoch. Much like the previous classes, here we can also note that in the latter epochs the accuracy of real is much higher than the previous epochs, similarly, the accuracy of fake is gradually decreasing compared to the previous epochs.

5.1.4 Moderate Demented Class

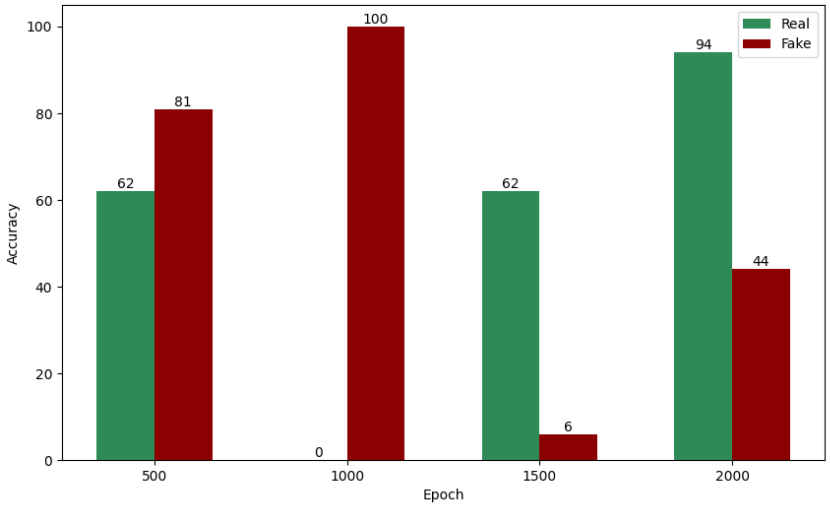

Figure 5.8: Bar chart of real vs. fake detection rate progression in Moderate Demented Class

We have generated 2000 sets of images running the epoch 2000 times in the Moderate Demented class. The initial epoch gives us an accuracy rate of 100% both for real and fake. In the 1000th epoch, we get 0% accuracy for real and 100% accuracy for fake. We get 62% real accuracy and 6% fake accuracy in the 1500th epoch. In the final epoch, the accuracy is 94% for real and 44% for fake. In the bar chart it is visible that the latter epochs show more accurate real than the previous ones, while the accuracy of fake has decreased compared to the previous epochs.

5.2 Pre-FED Disease Detection

5.2.1 VGG16 Outcome

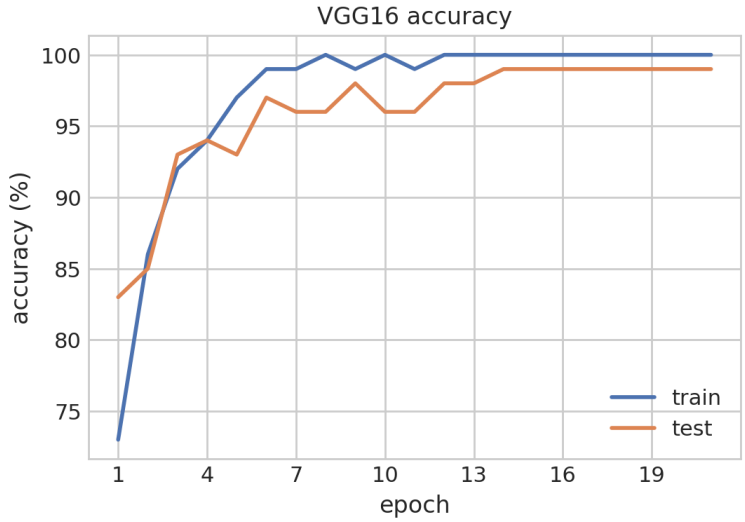

Figure 5.9: Line graph of VGG16 accuracy (Pre-FED)

For VGG16, the accuracy percentage indicates how accurately the model can distinguish between the classes. Higher accuracy means the model accurately identifies which class the given data belongs to. We have initiated VGG16 with 100 epochs.

At the initial epoch, the accuracy of the training dataset is 75.5%, and it is gradually increasing until it reaches 100% at around the 12^{th} epoch.

On the other hand, for the test dataset, the accuracy is between 85% - 88% at the initial epoch, and it gradually increases with each epoch, finally reaching 98.44%.

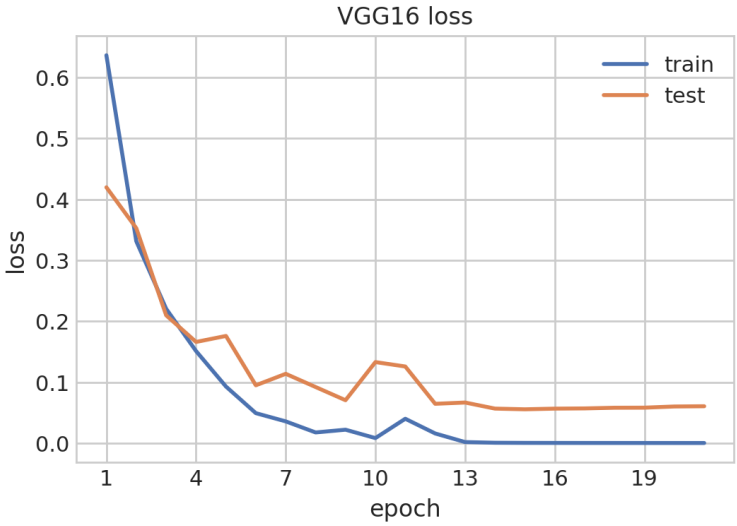

Figure 5.10: Line graph of VGG16 loss (Pre-FED)

For VGG16, the lower loss is better for classification. From the above line graph, we can see that, at the initial epoch, loss for the training dataset is at 0.5945, and it reaches less than 0.0013 by gradually decreasing at around the 12^{th} epoch.

For the test dataset, the loss function's output is between 0.37 to 0.40 at the initial epoch, and it reaches almost 0.05 at the end by gradually decreasing.

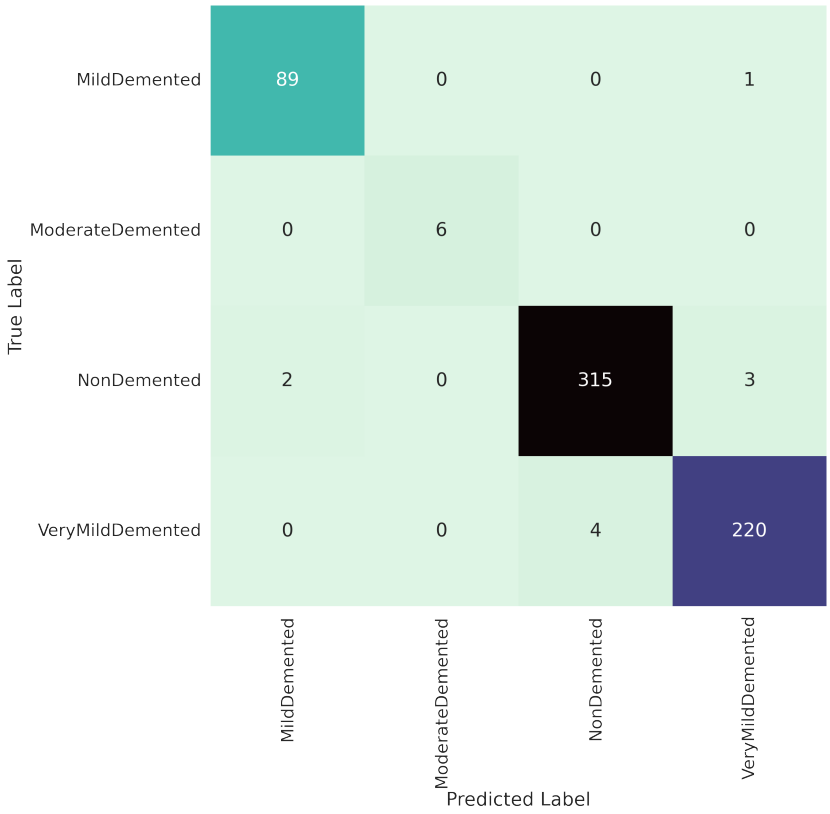

Figure 5.11: Confusion matrix of VGG16 (Pre-FED) - Testing on real images' dataset

When it comes to classifying data, a classification method's performance is shown in the form of a confusion matrix and it indicates how well a categorization system works in real life. The confusion matrix illustrates and summarizes the performance of a classification method.

From our confusion matrix given above, we can see that the Y-axis indicates the real labels, and the X-axis indicates the predicted labels by VGG16.

For the Mild Demented class, the VGG16 model correctly predicts 89 Mild Demented classes, while failing to predict one of Very Mild Demented class.

Furthermore, for the Moderate Demented class, the VGG16 model correctly predicts 6 Moderate Demented classes.

Next, for the Non-Demented class, VGG16 correctly predicts 315 while failing to predict 2 from Mild Demented and 3 from Very Mild Demented.

Finally, for the Very Mild Demented, VGG16 correctly predicts 220 while wrongly predicting 4 from Non-Demented.

However, the dataset containing real MRI images of the brain does not have enough entries for which the previous confusion matrix contains less images prediction for mild demented and moderate demented class. In solution to this, we tested our model against a mixed dataset containing real and generated MRI images of the brain and obtained the following confusion matrix.

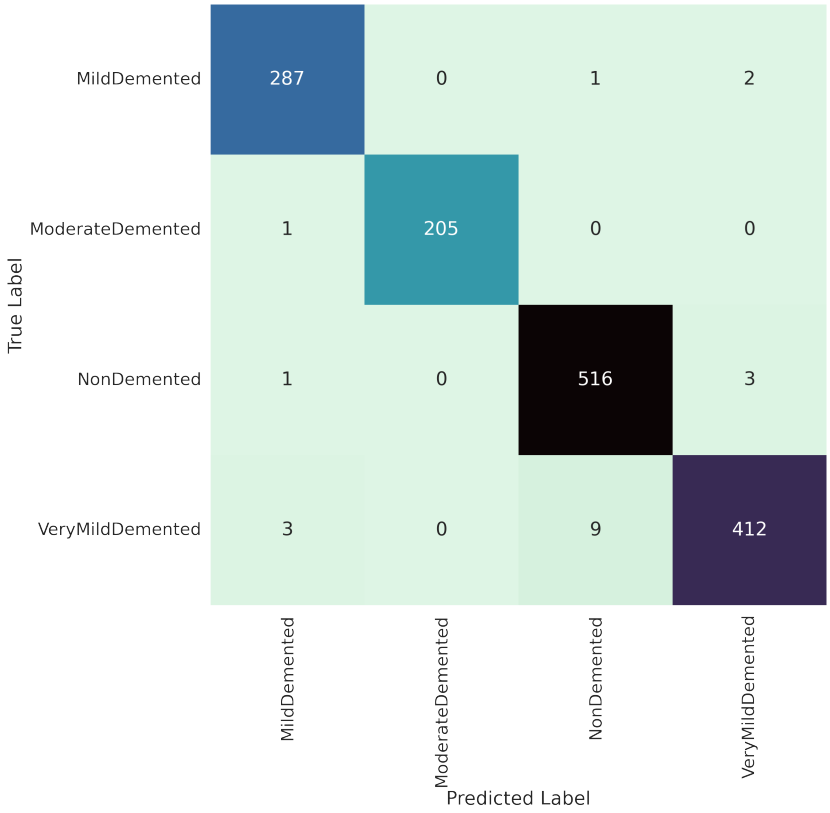

Figure 5.12: Confusion matrix of VGG16 (Pre-FED) - Testing on mixed images' dataset

From our confusion matrix given above, we can see that the Y-axis indicates the real labels, and the X-axis indicates the predicted labels by VGG16.

For the Mild Demented class, the VGG16 model correctly predicts 287 Mild Demented classes, while failing to predict two of Very Mild Demented class and one of Non Demented class.

Furthermore, for the Moderate Demented class, the VGG16 model correctly predicts 205 Moderate Demented classes while failing to predict 1 from Mild Demented class.

Next, for the Non-Demented class, VGG16 correctly predicts 516 while failing to predict 3 from Very Mild Demented and 1 from Mild Demented.

Finally, for the Very Mild Demented, VGG16 correctly predicts 412 while wrongly predicting 9 from Non-Demented and 3 from Mild Demented class.

5.2.2 EfficientNetB6 Outcome

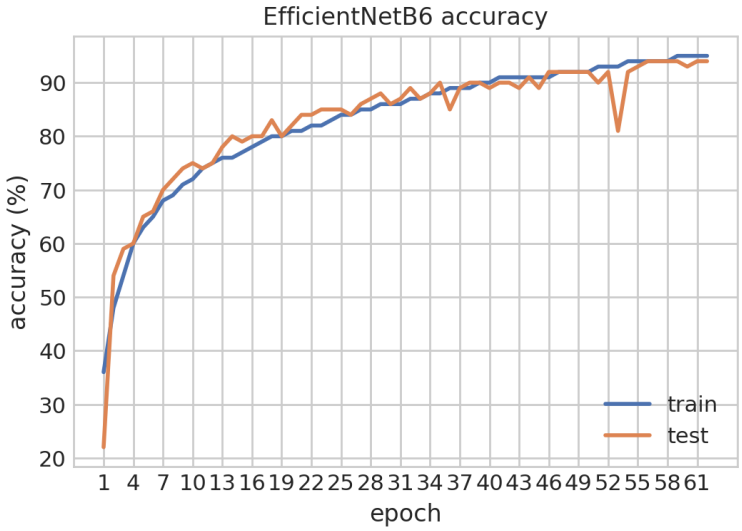

Figure 5.13: Line graph of EfficientNetB6 accuracy (Pre-FED)

We have initiated EfficientNetB6 with 100 epochs. At the initial epoch, the accuracy of the training dataset is 35.57%, and it is gradually increasing until it reaches 95.09% at around the 61st epoch.

On the flip side, for the test dataset, the accuracy is between 20% to 25% at the initial epoch, and it gradually increases with each epoch, finally reaching 87.50%.

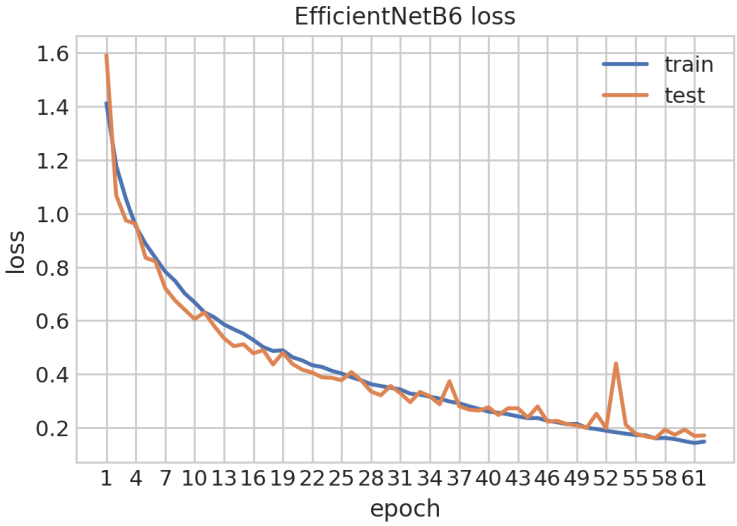

Figure 5.14: Line graph of EfficientNetB6 loss (Pre-FED)

From the above line graph we can see that, at the initial epoch, loss for the training dataset is at 1.4108, and it reaches 0.1438 by gradually decreasing at around the 61^{st} epoch.

For the test dataset, the loss function's output is almost 1.6 at the initial epoch, and it reaches 0.3413 at the end by gradually decreasing.

Figure 5.15: Confusion matrix of EfficientNetB6 (Pre-FED)

From the confusion matrix given above, we observe that the Y-axis indicates the real labels, and the X-axis indicates the predicted labels by EfficientNetB6.

For the Mild Demented class, the EfficientNetB6 model correctly predicts 77 Mild Demented classes, while failing to predict three of Non-Demented and 10 from Very Mild Demented class.

Moreover, for the Moderate Demented class, the EfficientNetB6 model correctly predicts 3 Moderate Demented classes, while predicting 2 of the Mild Demented and 1 of the Very Mild Demented class falsely.

Next, for the Non-Demented class, EfficientNetB6 correctly predicts 288 while failing to predict 6 from Mild Demented and 26 from Very Mild Demented.

Finally, for the Very Mild Demented, EfficientNetB6 correctly predicts 192 while wrongly predicting 7 from Mild Demented and 25 from Non-Demented.

5.2.3 Xception Outcome

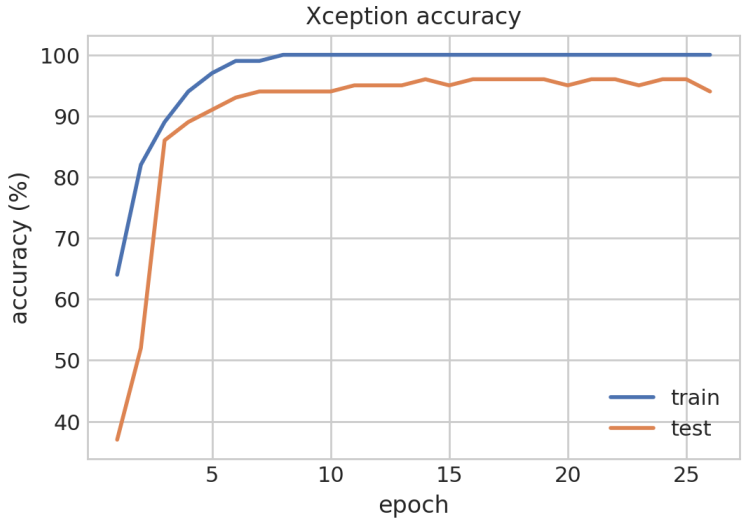

Figure 5.16: Line graph of Xception accuracy (Pre-FED)

We have initiated Xception with 100 epochs. At the initial epoch, the accuracy of the training dataset is 62.59%, and it is gradually increasing until it reaches almost 100% at around the 19^{th} epoch.

On the flip side, for the test dataset, the accuracy is between 32% to 35% at the initial epoch, and it gradually increases with each epoch, finally reaching 90.47%.

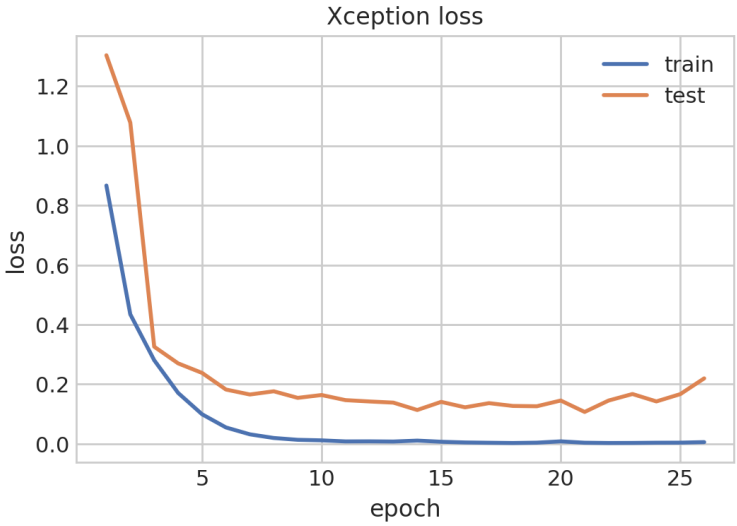

Figure 5.17: Line graph of Xception loss (Pre-FED)

From the above line graph we can see that, at the initial epoch, loss for the training dataset is at 0.8775, and it reaches 0.0041 by gradually decreasing at around the 19^{th} epoch.

For the test dataset, the loss function's output is greater than 1.2 at the initial epoch, and it reaches 0.2881 at the end by gradually decreasing.

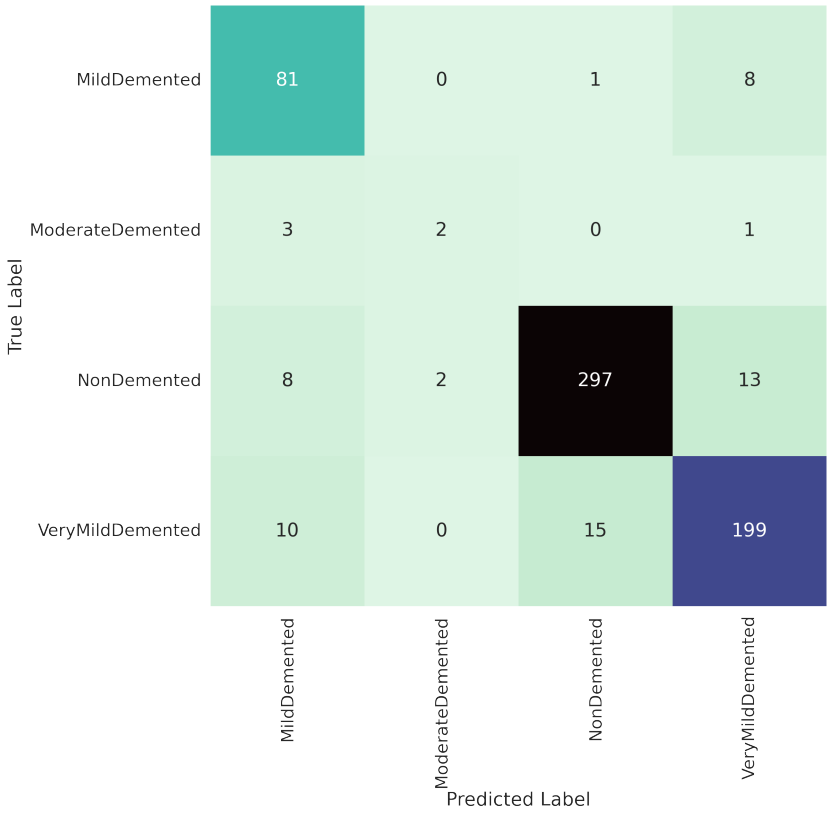

Figure 5.18: Confusion matrix of Xception (Pre-FED) - Testing on real images' dataset

From the confusion matrix given above, we observe that the Y-axis indicates the real labels, and the X-axis indicates the predicted labels by Xception.

For the Mild Demented class, the Xception model correctly predicts 81 Mild Demented classes, while failing to predict 1 from the Non Demented class and 8 from the Very Mild Demented class.

Nevertheless, for the Moderate Demented class, the Xception model correctly predicts 2 Moderate Demented classes, while wrongly predicting 3 from the Mild Demented class and 1 from the Very Mild Demented Class.

Next, for the Non-Demented class, Xception correctly predicts 297 while failing to predict 8 from Mild Demented and 13 from Very Mild Demented.

Finally, for the Very Mild Demented, Xception correctly predicts 199 while wrongly predicting 10 from Mild Demented and 15 from Non-Demented.

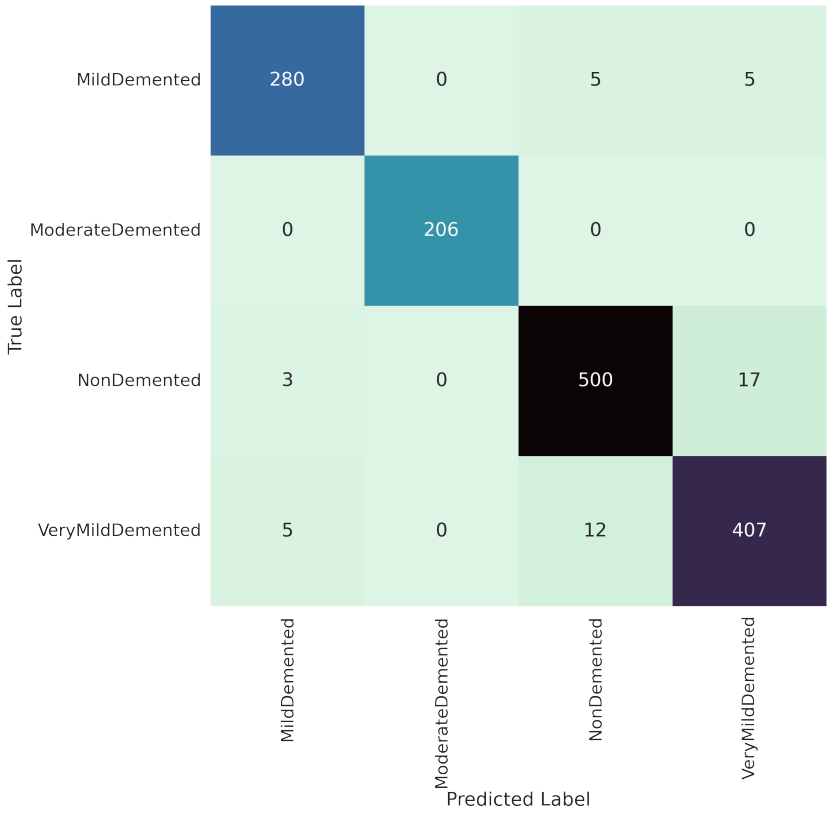

Figure 5.19: Confusion matrix of Xception (Pre-FED) - Testing on mixed images' dataset

From the confusion matrix given above, we observe that the Y-axis indicates the real labels, and the X-axis indicates the predicted labels by Xception.

For the Mild Demented class, the Xception model correctly predicts 280 Mild Demented classes, while failing to predict 5 from the Non Demented class and 5 from the Very Mild Demented class.

Nevertheless, for the Moderate Demented class, the Xception model correctly predicts 206 Moderate Demented classes.

Next, for the Non-Demented class, Xception correctly predicts 500 while failing to predict 3 from Mild Demented and 17 from Very Mild Demented.

Finally, for the Very Mild Demented, Xception correctly predicts 407 while wrongly predicting 5 from Mild Demented and 12 from Non-Demented.

5.3 Post-FED Disease Detection

5.3.1 VGG16 Outcome

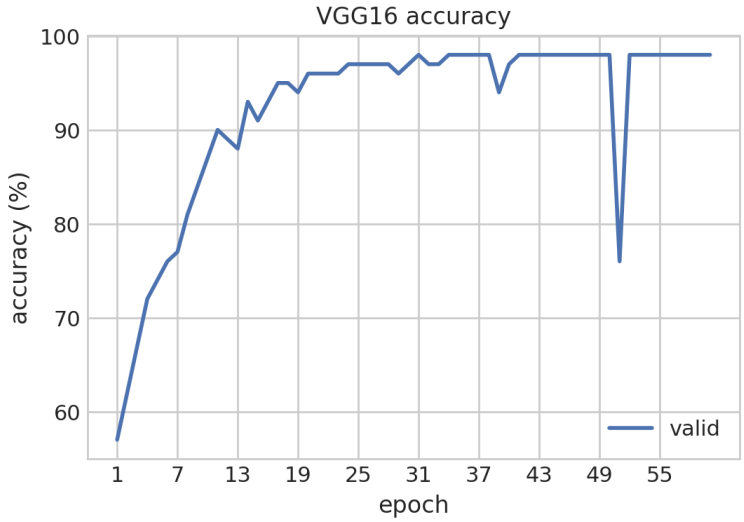

Figure 5.20: Line graph of VGG16 accuracy (Post-FED)

For VGG16, the accuracy percentage indicates how accurately the model can distinguish between the classes. Higher accuracy means the model accurately identifies which class the given data belongs to.

For the valid dataset, the accuracy is between 55% - 60% initially, and it gradually increases with some spikes in the line graph along with each epoch, finally reaching 97.81%.

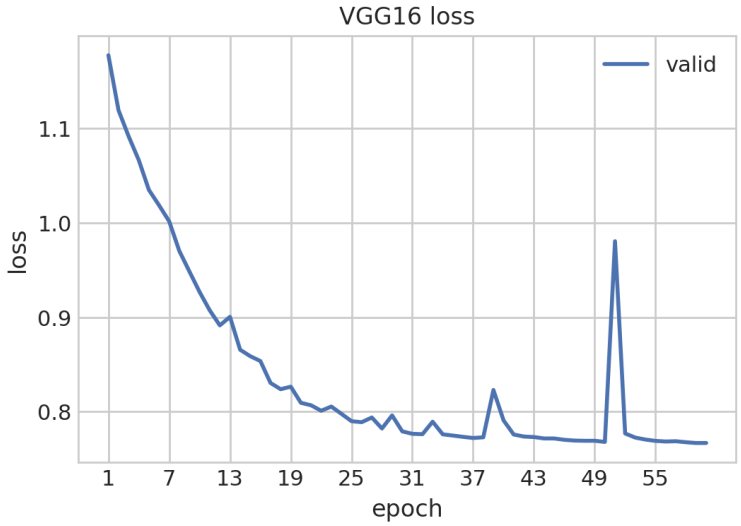

Figure 5.21: Line graph of VGG16 accuracy (Post-FED)

As we already discussed, for classification models like VGG16, the lower loss is better for classification.

From the above line graph, we can see that, at the initial epoch, loss for the valid dataset is more than 1.1, and it reaches less than 0.7 by gradually decreasing with some spikes at around the 55^{th} epoch.

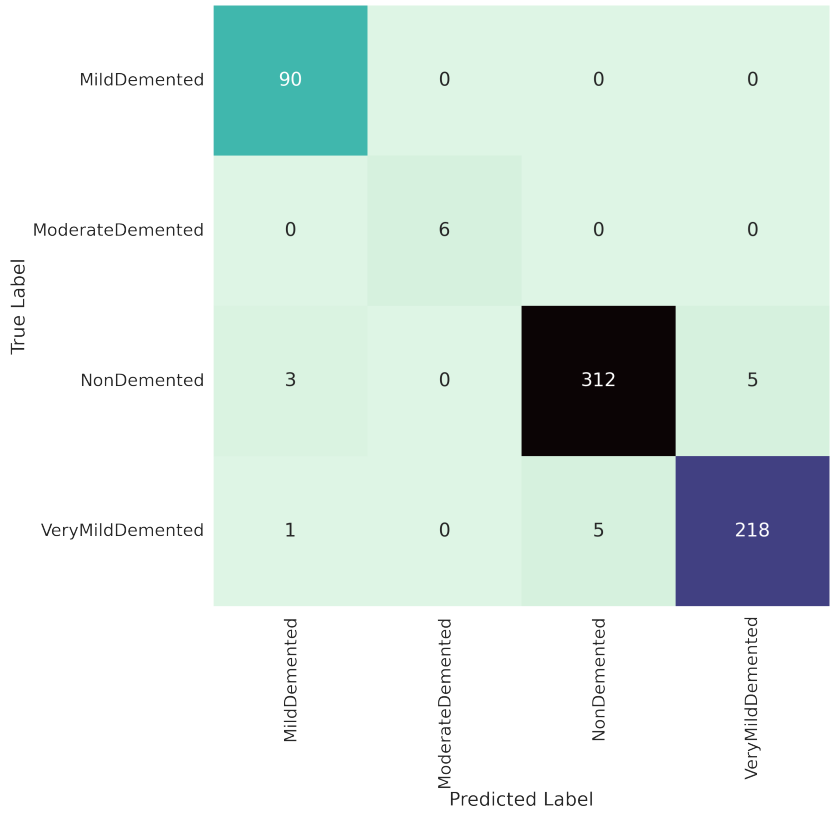

Figure 5.22: Confusion matrix of VGG16 (Post-FED) - Testing on real images' dataset

From our confusion matrix given above, we can see that the Y-axis indicates the real labels, and the X-axis indicates the predicted labels by VGG16.

For the Mild Demented class, the VGG16 model correctly predicts 90 Mild Demented classes.

Furthermore, for the Moderate Demented class, the VGG16 model correctly predicts 6 Moderate Demented classes.

Next, for the Non-Demented class, VGG16 correctly predicts 312 while failing to predict 3 from Mild Demented and 5 from Very Mild Demented.

Finally, for the Very Mild Demented, VGG16 correctly predicts 218 while wrongly predicting 5 from Non-Demented and 1 from Mild Demented class.

However, in this scenario also, the dataset containing real MRI images of the brain does not have enough entries for which the previous post-FED confusion matrix contains less images prediction for mild demented and moderate demented class. In solution to this, we tested our model against a mixed dataset containing real and generated MRI images of the brain and obtained the following confusion matrix.

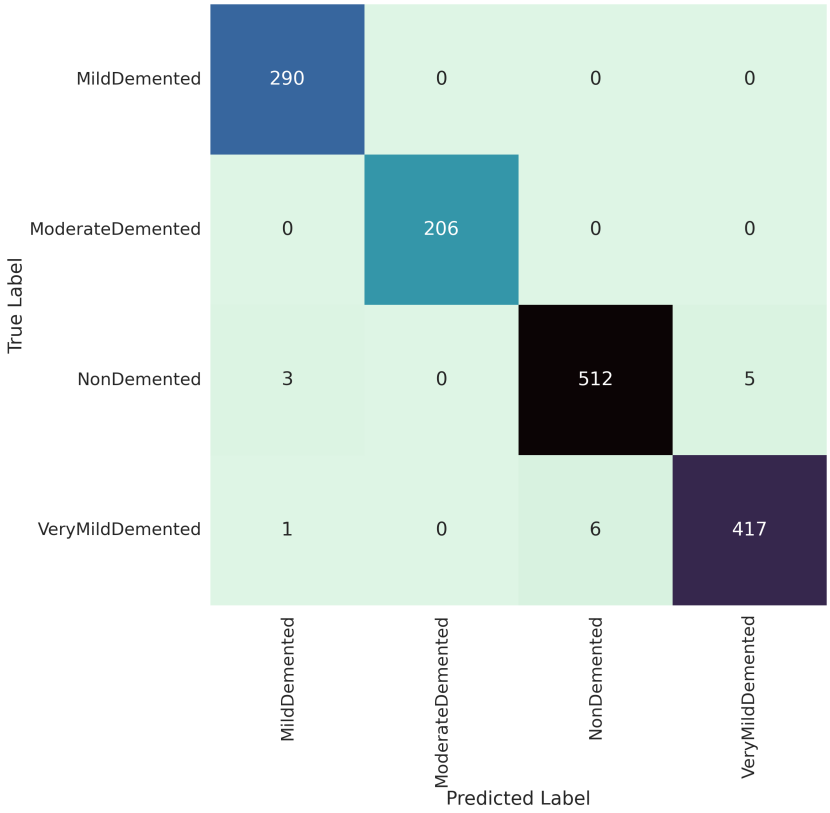

Figure 5.23: Confusion matrix of VGG16 (Post-FED) - Testing on mixed images' dataset

From our confusion matrix given above, we can see that the Y-axis indicates the real labels, and the X-axis indicates the predicted labels by VGG16.

For the Mild Demented class, the VGG16 model correctly predicts 290 Mild Demented classes, while predicting nothing wrongly.

Furthermore, for the Moderate Demented class, the VGG16 model correctly predicts 206 Moderate Demented classes.

Next, for the Non-Demented class, VGG16 correctly predicts 512 while failing to predict 3 from Mild Demented and 5 from Very Mild Demented.

Finally, for the Very Mild Demented, VGG16 correctly predicts 417 while wrongly predicting 6 from Non-Demented and 1 from Mild Demented class.

5.3.2 Xception Outcome

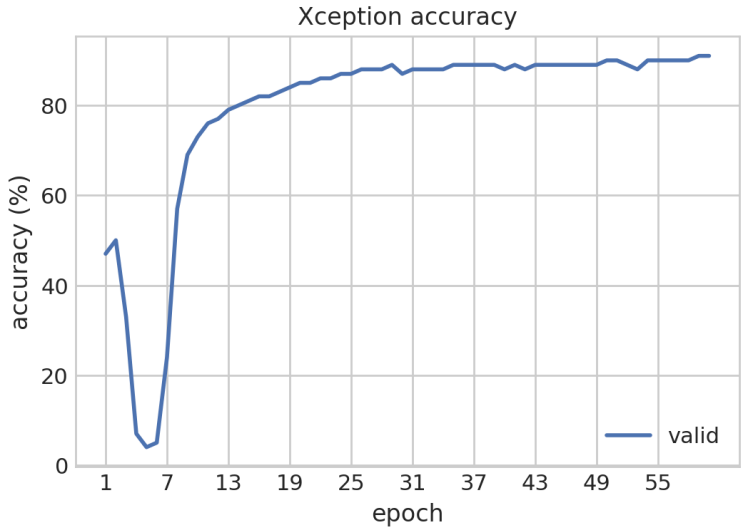

Figure 5.24: Line graph of Xception accuracy (Post-FED)

For the valid dataset, the line graph starts around 50%, decreases sharply till 7 epochs and then gradually rises with each epoch, finally reaching 93.13%.

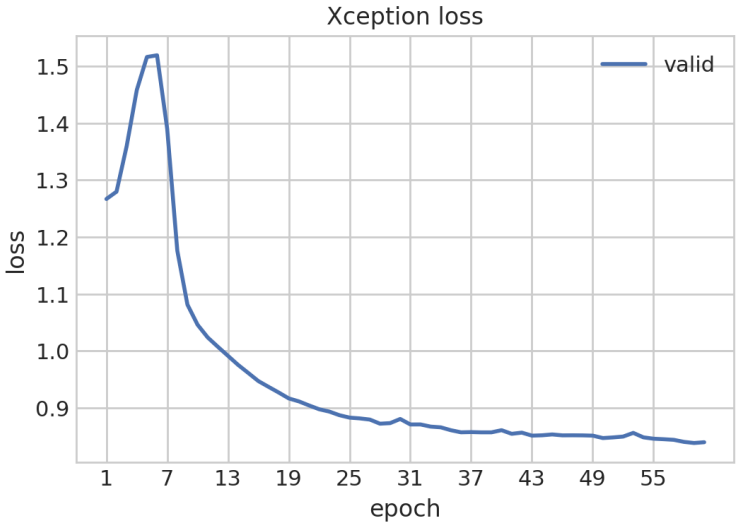

Figure 5.25: Line graph of Xception loss (Post-FED)

Similar to the accuracy line graph, for the valid dataset, the line graph starts around 1.3, increases sharply till 7 epochs and then gradually falls with each epoch, finally reaching less than 0.9.

Figure 5.26: Confusion matrix of Xception (Post-FED) - Testing on real images' dataset

From the confusion matrix given above, we observe that the Y-axis indicates the real labels, and the X-axis indicates the predicted labels by Xception.

For the Mild Demented class, the Xception model correctly predicts 83 Mild Demented classes, while failing to predict 2 from the Non Demented class and 5 from the Very Mild Demented class.

Nevertheless, for the Moderate Demented class, the Xception model correctly predicts 6 Moderate Demented classes, while not predicting any classes wrong.

Next, for the Non-Demented class, Xception correctly predicts 301 while failing to predict 2 from Mild Demented and 17 from Very Mild Demented.

Finally, for the Very Mild Demented, Xception correctly predicts 206 while wrongly predicting 6 from Mild Demented and 12 from Non-Demented.

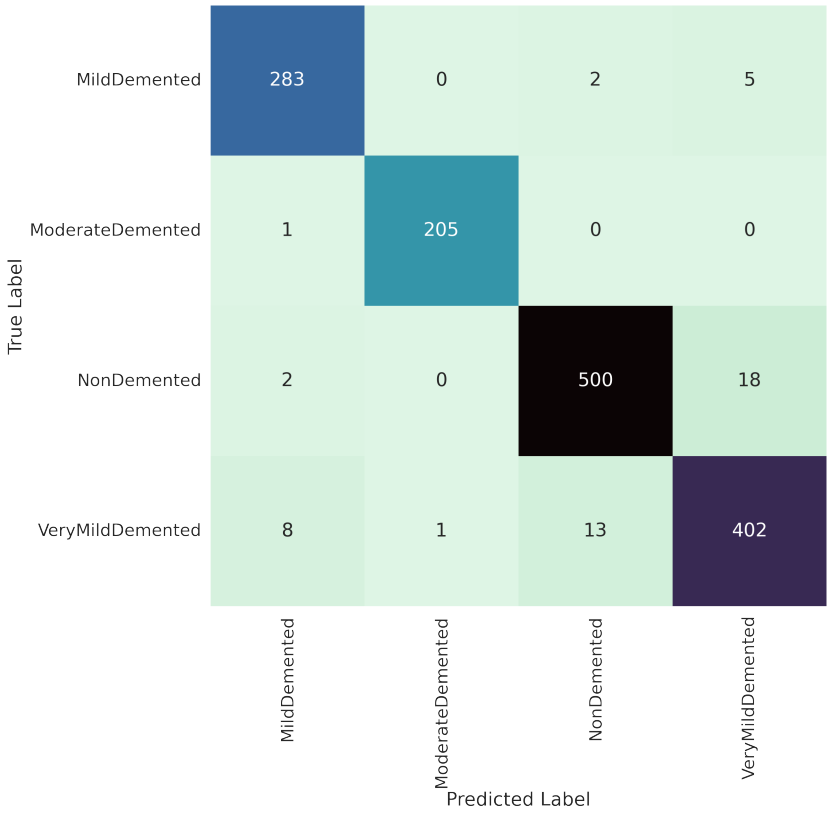

Figure 5.27: Confusion matrix of Xception (Post-FED) - Testing on mixed images' dataset

From the confusion matrix given above, we observe that the Y-axis indicates the real labels, and the X-axis indicates the predicted labels by Xception.

For the Mild Demented class, the Xception model correctly predicts 83 Mild Demented classes, while failing to predict 2 from the Non Demented class and 5 from the Very Mild Demented class.

Nevertheless, for the Moderate Demented class, the Xception model correctly predicts 6 Moderate Demented classes, while not predicting any classes wrong.

Next, for the Non-Demented class, Xception correctly predicts 301 while failing to predict 2 from Mild Demented and 17 from Very Mild Demented.

Finally, for the Very Mild Demented, Xception correctly predicts 206 while wrongly predicting 6 from Mild Demented and 12 from Non-Demented.

5.4 Grad-CAM

Grad-CAM, a method of Explainable Artificial Intelligence (XAI) that uses gradients of any target concepts flowing into the final layer to produce a coarse localization map that highlights key areas in the image for predicting the target, excels in producing visual explanations for decisions from a large class of Convolutional Neural Networks, according to [34]. Advantageously, Grad-CAM can be used with a broad range of CNN model families, including CNNs with fully connected layers like VGG16.

We have applied Grad-CAM to get a visual explanation on our fully connected layers model VGG16 and achieved the following output.

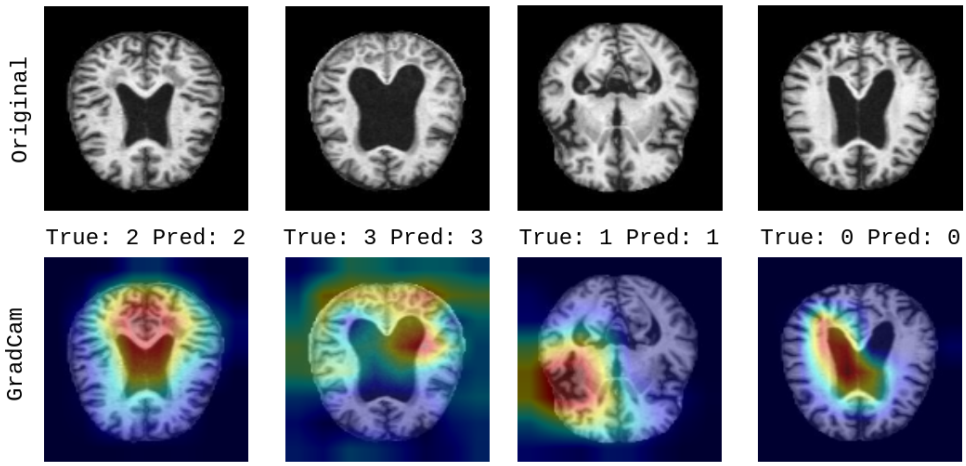

Figure 5.28: Grad-CAM Visualization of VGG16 Model

When we look at the Grad-CAM output in the figure, we can see the VGG16 activating in the top middle portion of the brain in the non demented class (2), the upper part of the brain in the very mild demented class (3), the bottom left part of the MRI image in the moderate demented class (1), and the center of the MRI image in the mildly demented class (0), showing a significant feature of the MRI picture utilized by the VGG16 network to categorize the image.

5.5 Classification Reports & Comparisons

Overall, after augmenting our original dataset, training disease detection models with that data, and testing those models with fictitious data, we successfully achieved the following successful outcome before implementing federated learning:

Model	Accuracy	Precision	Recall	f1-Score
VGG16	0.9844	0.9844	0.9844	0.9844
EfficientNetB6	0.8750	0.8762	0.8750	0.8748
Xception	0.9047	0.9060	0.9047	0.9046

Table 5.1: Classification Report of VGG16, EfficientNetB6 & Xception (Pre-FED)

However, after implementing federated learning, we were updating the global model parameters with the up to date values received from the local training models from the client side using FedAvg and as a result, our previous values of accuracy, precision, recall and f1-score have been refurbished a bit.

Model	Accuracy	Precision	Recall	f1-Score
VGG16	0.9781	0.9783	0.9781	0.9781
Xception	0.9313	0.9316	0.9313	0.9314

Table 5.2: Classification Report of VGG16 & Xception (Post-FED)

From the above pre-Fed and post-Fed classification data it is observable that, there is a consequential difference between accuracy and precision achieved through the VGG16 model, while pre-Fed dataset gives slightly higher accuracy & precision than post-Fed dataset, recall and f1-scores tend to follow the same. However, the Xception model gives us a higher accuracy, precision, recall and fa-scores through the post-Fed dataset compared to pre-Fed dataset. It is important to note that the EfficientNetB6 model has only been used in pre-Fed dataset and has produced sub-optimal accuracy and precision scores.

Chapter 6

Conclusion

Image augmentation is very effective in artificially expanding the dataset. The medical sector is a domain where data is scarce, even more so when it comes to MRI images of the brain for Alzheimer's detection. Here, image augmentation has proven to be a viable solution to the scarcity of data. However, the security of patients' data has always been a question mark in the medical sector.

Identifying the symptoms of Alzheimer's disease early can tremendously help the patients. While MRI scans may detect Alzheimer's disease-related brain shrinkage, it also rules out other illnesses. However, a system must be able to assess, detect, and classify patient data, in this case, MRI scans, with high detail while respecting data privacy for the disease detection to be successful.

As a result, developing a Deep-Learning based detection system that can effectively read, decipher, and provide necessary classifications based on patient data while ensuring ultimate confidentiality and safety is a massive challenge that can only be completed with cutting-edge technologies. We believe that by implementing Federated learning we have met the demands of privacy and through GAN-based image augmentation we have achieved accuracy.

Conclusively, this research proposes a framework for utilizing distributed patient data to augment and further enrich the dataset and detect medical conditions with the implementation of a Federated GAN model. We believe this research will significantly enhance augmenting biomedical images and detect medical conditions accurately.

Bibliography

[1] H. R. Roth, L. Lu, J. Liu, *et al.*, "Improving computer-aided detection using convolutional neural networks and random view aggregation," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1170–1181, 2016. DOI: 10.1109/TMI.2015.2482920.

[2] G. Litjens, T. Kooi, B. E. Bejnordi, *et al.*, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, 2017, ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2017.07.005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361841517301135.

[3] H. Greenspan, B. van Ginneken, and R. M. Summers, "Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1153–1159, 2016. DOI: 10.1109/TMI.2016.2553401.

[4] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, *et al.*, "Convolutional neural networks for medical image analysis: Full training or fine tuning?" *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1299–1312, May 2016. DOI: 10.1109/tmi.2016.2535302. [Online]. Available: https://doi.org/10.1109%2Ftmi.2016.2535302.

[5] J. Shi, S. Zhou, X. Liu, Q. Zhang, M. Lu, and T. Wang, "Stacked deep polynomial network based representation learning for tumor classification with small ultrasound image dataset," *Neurocomputing*, vol. 194, pp. 87–94, 2016, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2016.01.074. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231216002344.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25, Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[7] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification," *Neurocomputing*, vol. 321, pp. 321–331, 2018, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2018.09.013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231218310749.

[8] P. Costa, A. Galdran, M. I. Meyer, *et al.*, *Towards adversarial retinal image synthesis*, 2017. DOI: 10.48550/ARXIV.1701.08974. [Online]. Available: https://arxiv.org/abs/1701.08974.

[9] W. Dai, J. Doyle, X. Liang, *et al.*, *Scan: Structure correcting adversarial network for organ segmentation in chest x-rays*, 2017. DOI: 10.48550/ARXIV.1703.08770. [Online]. Available: https://arxiv.org/abs/1703.08770.

[10] Y. Xue, T. Xu, H. Zhang, L. R. Long, and X. Huang, "Segan: Adversarial network with multi-scale l1 loss for medical image segmentation," *Neuroinformatics*, vol. 16, no. 3-4, pp. 383–392, 2018. DOI: 10.1007/s12021-018-9377-x.

[11] D. Nie, R. Trullo, C. Petitjean, S. Ruan, and D. Shen, *Medical image synthesis with context-aware generative adversarial networks*, 2016. DOI: 10.48550/ARXIV.1612.05362. [Online]. Available: https://arxiv.org/abs/1612.05362.

[12] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, *Unsupervised anomaly detection with generative adversarial networks to guide marker discovery*, 2017. DOI: 10.48550/ARXIV.1703.05921. [Online]. Available: https://arxiv.org/abs/1703.05921.

[13] A. Ben-Cohen, E. Klang, S. P. Raskin, M. M. Amitai, and H. Greenspan, "Virtual PET images from CT data using deep convolutional networks: Initial results," in *Simulation and Synthesis in Medical Imaging*, Springer International Publishing, 2017, pp. 49–57. DOI: 10.1007/978-3-319-68127-6_6. [Online]. Available: https://doi.org/10.1007%2F978-3-319-68127-6_6.

[14] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, PMLR, 2017, pp. 1273–1282.

[15] Y. Zhao, J. Zhao, L. Jiang, *et al.*, "Privacy-preserving blockchain-based federated learning for iot devices," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1817–1829, 2021. DOI: 10.1109/JIOT.2020.3017377.

[16] C. Song, T. Ristenpart, and V. Shmatikov, *Machine learning models that remember too much*, 2017. DOI: 10.48550/ARXIV.1709.07886. [Online]. Available: https://arxiv.org/abs/1709.07886.

[17] M. Elgendi, M. U. Nasir, Q. Tang, *et al.*, "The effectiveness of image augmentation in deep learning networks for detecting covid-19: A geometric transformation perspective," *Frontiers in Medicine*, vol. 8, p. 629 134, 2021.

[18] W. Liang, J. Yao, A. Chen, *et al.*, "Early triage of critically ill covid-19 patients using deep learning," *Nature Communications*, vol. 11, no. 1, 2020. DOI: 10.1038/s41467-020-17280-8.

[19] M. D. Bloice, P. M. Roth, and A. Holzinger, "Biomedical image augmentation using augmentor," *Bioinformatics*, vol. 35, no. 21, pp. 4522–4524, 2019.

[20] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, p. 125, 2020.

[21] M. A. Marcinkiewicz, "Building a large annotated corpus of english: The penn treebank," *Using Large Corpora*, vol. 273, 1994.

[22] C. He, M. Annavaram, and S. Avestimehr, "Group knowledge transfer: Federated learning of large cnns at the edge," *Advances in Neural Information Processing Systems*, vol. 33, pp. 14 068–14 080, 2020.

[23] T. Zhang, L. Gao, C. He, M. Zhang, B. Krishnamachari, and A. S. Avestimehr, "Federated learning for the internet of things: Applications, challenges, and opportunities," *IEEE Internet of Things Magazine*, vol. 5, no. 1, pp. 24–29, 2022.

[24] T. Zhang, C. He, T. Ma, L. Gao, M. Ma, and S. Avestimehr, "Federated learning for internet of things," in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, 2021, pp. 413–419.

[25] S. Augenstein, H. B. McMahan, D. Ramage, *et al.*, "Generative models for effective ml on private, decentralized datasets," *arXiv preprint arXiv:1911.06679*, 2019.

[26] T. Nazir, A. Irtaza, A. Javed, H. Malik, D. Hussain, and R. A. Naqvi, "Retinal image analysis for diabetes-based eye disease detection using deep learning," *Applied Sciences*, vol. 10, no. 18, p. 6185, 2020.

[27] H. Panwar, P. Gupta, M. K. Siddiqui, R. Morales-Menendez, P. Bhardwaj, and V. Singh, "A deep learning and grad-cam based color visualization approach for fast detection of covid-19 cases using chest x-ray and ct-scan images," *Chaos, Solitons & Fractals*, vol. 140, p. 110 190, 2020, ISSN: 0960-0779. DOI: https://doi.org/10.1016/j.chaos.2020.110190. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960077920305865.

[28] D. Han, Q. Liu, and W. Fan, "A new image classification method using cnn transfer learning and web data augmentation," *Expert Systems with Applications*, vol. 95, pp. 43–56, 2018.

[29] Z. Liu, C. Yang, J. Huang, S. Liu, Y. Zhuo, and X. Lu, "Deep learning framework based on integration of s-mask r-cnn and inception-v3 for ultrasound image-aided diagnosis of prostate cancer," *Future Generation Computer Systems*, vol. 114, pp. 358–367, 2021.

[30] S. J. Mambou, P. Maresova, O. Krejcar, A. Selamat, and K. Kuca, "Breast cancer detection using infrared thermal imaging and a deep learning model," *Sensors*, vol. 18, no. 9, p. 2799, 2018.

[31] S. Dubey, "Alzheimer's dataset (4 class of images)," *Kaggle*, 2019. [Online]. Available: https://www.kaggle.com/datasets/tourist55/alzheimers-dataset-4-class-of-images.

[32] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, *Generative adversarial networks*, 2014. DOI: 10.48550/ARXIV.1406.2661. [Online]. Available: https://arxiv.org/abs/1406.2661.

[33] S. Tammina, "Transfer learning using vgg-16 with deep convolutional neural network for classifying images," *International Journal of Scientific and Research Publications (IJSRP)*, vol. 9, no. 10, pp. 143–150, 2019.

[34] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 336–359, Oct. 2019. DOI: 10.1007/s11263-019-01228-7. [Online]. Available: https://doi.org/10.1007%2Fs11263-019-01228-7.

[35] C. He, A. D. Shah, Z. Tang, *et al.*, "Fedcv: A federated learning framework for diverse computer vision tasks," *arXiv preprint arXiv:2111.11066*, 2021.

[36] N. Dardagan, A. Brđanin, D. Džigal, and A. Akagic, "Multiple object trackers in opencv: A benchmark," in *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, IEEE, 2021, pp. 1–6.

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

[38] 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.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[40] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

[41] J. Bridle, "Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters," *Advances in neural information processing systems*, vol. 2, 1989.

[42] J. Feng and S. Lu, "Performance analysis of various activation functions in artificial neural networks," in *Journal of physics: conference series*, IOP Publishing, vol. 1237, 2019, p. 022 030.

[43] T. Sun, D. Li, and B. Wang, "Decentralized federated averaging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.