NeuroSim: AI-Enhanced Simulation of Human Brain Activity

Shahna Shahul Hameed
School of Computing, MSc. in Artificial Intelligence
National College of Ireland
Dublin 1, Ireland
Machine Learning (MSCAIJAN24I)
x22235094@student.ncirl.ie

***Abstract*—Medical research still faces several obstacles in the early detection and accurate prediction of neurological conditions like Parkinson's, Alzheimer's and many more. The effectiveness of early interventions is limited due to improvements in neuroimaging and diagnostic procedures, current methods frequently fail to provide precise and timely predictions. The present issue is the inability to merge several data formats like MRI scans, brain activity recordings, and clinical data, for instance - into a single predictive model. In order to solve this problem, NeuroSim, an AI-enhanced simulation of human brain function is proposed. The method uses multi-modal data to forecast the start and development of neurological diseases using Deep Learning models, namely Convolutional Neural Networks (CNNs) using a VGG19 architecture. Preprocessing MRI images and using trained models to determine illness stages are part of the methodology. The findings show significant precision in differentiating between Alzheimer's and Parkinson's disease phases, providing up the possibility to more effectual early diagnosis and individualized treatment strategies.**

***Index Terms*—neuroimaging, deep learning, convolutional neural networks, VGG19, Alzheimer's**

I. Introduction

Neurological disorders, which impact millions of individuals globally, include Alzheimer's and Parkinson's disease. They are among the most challenging illnesses to identify, diagnose and treat. The most common type of dementia, Alzheimer's disease alone, affects about 50-60 million people all around the world. If considerable progress fails to be made in early identification and treatment, it is researched and concluded that this number will increase triple times by 2050.

Similar to this, the complicated symptoms and course of Parkinson's disease, which impacts over 10 million globally, provide serious risks. For many conditions to be effectively treated and managed, early diagnosis and precise initial stage prediction are essential; however, current diagnostic techniques often fail to give such accurate predictions.

The majority of present diagnostic methods depend heavily on neuroimaging methods, such as MRI scans, which provide detailed visualizations of brain structures. Although these techniques have improved the detection of neurological disorders, they often come up short of the sensitivity and specificity required for an early diagnosis. Automating the integration of neuroimaging data has shown promise in recent developments in artificial intelligence, especially in Deep Learning techniques like Convolutional Neural Networks (CNNs). This could result in diagnoses that are more precise and fast.

The research question guiding this study is: *How can Machine Learning and Artificial Intelligence can be used to develop a broad and predictive simulation of the human brain and accurately predict the start and progression of specific neurological diseases by effectively integrating multi modal data?* The goal of this study is to create and assess a CNN-based model that can use MRI data to categorize patients into various phases of these illnesses. The objective of this research is to improve the precision and dependability of early-stage neurological disease diagnosis by utilizing CNNs' capacity for pattern recognition. Convolutional Neural Networks (CNNs), in particular, are powerful and advanced Machine Learning techniques that can be used to increase the precision and dependability of early-stage neurological disease diagnosis.

The format of this document is as follows: The gaps that this study aims to fill are highlighted in the next section, which analyzes similar work in the fields of artificial intelligence and neuroimaging for disease prediction. The procedures for gathering data, preprocessing, and developing models are then covered fully in the methodology section. The model's performance results are shown in the evaluation section, along with a discussion of the consequences. The conclusion, which closes the study, provides recommendations for further research directions.

II. Literature Review

A. Related Work

Recent years have shown considerable progress in the application of deep learning techniques, specifically Convolutional Neural Networks (CNNs), in neuroimaging. Researchers have investigated multiple methods to utilize CNNs in the categorization and forecasting of neurological disorders like Parkinson's and Alzheimer's illnesses. The important works in the field are reviewed in this section, together with an analysis of the models applied, datasets used, accuracy acquired, and limits of each technique. The review serves as a basis for the

current investigation by highlighting both the advancements and the gaps that still need to be filled.

B. Alzheimer's Disease Prediction and Classification

One of the most significant studies in predicting Alzheimer's disease progression is by Tong et al. (2017) (1). who proposed a novel grading biomarker for predicting the transmutation from Mild Cognitive Impairment (MCI) to Alzheimer's Disease (AD). The study employed structural Magnetic Resonance Imaging (MRI) data from the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset. A machine learning model was developed to foresee the transformation from MCI to AD, achieving an accuracy of 78.8%. The proposed grading biomarker gave valuable information into the potential of early identification, but the study was limited by the model's dependency on a single modality (MRI) and a relatively small sample size, which could affect its applicability to wider populations.

Yagis et al. (2020) (2) took a different approach by using 3D Convolutional Neural Networks (3D CNNs) for diagnosing Alzheimer's Disease through structural MRI. The study utilized the OASIS dataset and demonstrated the supremacy of 3D CNNs in capturing spatial features compared to traditional 2D CNNs. The model achieved an accuracy of 85.6% in classifying Alzheimer's disease stages. While the study featured the advantages of using 3D CNNs in neuroimaging, it also highlighted the increased computational complexity and the need for more extensive datasets to avoid over-fitting.

Islam and Zhang (2018) (3) explored the advantages of deep learning architectures for the early diagnosis of Alzheimer's Disease through neuroimaging. Their research used the ADNI dataset and employed multiple deep learning models, including CNNs and Long Short-Term Memory (LSTM) networks, to observe MRI images. The models achieved an accuracy of 91.4%, showing the capability of deep learning in early Alzheimer's diagnosis. However, the study was limited by its focus on a single modality (MRI), and the authors suggested future work to include multi modal data for more elaborated predictions.

Basaia et al. (2020) (4) also contributed to this field by applying transfer learning techniques to the classification of Alzheimer's disease and Mild Cognitive Impairment (MCI) using MRI data from the OASIS-3 dataset. Their approach exploited pre-trained deep learning models to acquire an accuracy of 89.3% in classifying Alzheimer's disease stages. Transfer learning proved effective in reducing training time and improving model performance on relatively small datasets. On the other hand, the study noted theoretical challenges in domain adaptation, specifically when applying the model to datasets with different imaging protocols.

C. Parkinson's Disease Prediction and Classification

Prashanth et al. (2017) (5) developed a high-accuracy classification model for Parkinson's disease (PD) based on surface fitting and shape analysis using Î23I-Ioflupane Single Photon Emission Computed Tomography (SPECT) imaging. With an accuracy of 89.4%, the study used a variety of machine learning approaches on the Parkinson's Progression Markers Initiative (PPMI) dataset. A combination of surface fitting techniques and shape analysis produced an effective framework for diagnosing PD. The unpredictability in imaging data and the requirement for bigger datasets to improve model robustness, however, restricted the study.

Yang et al. (2021) (6) proposed a multi-modal technique to Parkinson's Disease classification using features extracted from various data sources, including MRI, DaTSCAN, and clinical assessments. By using a stacking ensemble learning method, the study achieved an accuracy of 92.5%, showing the effectiveness of merging multi-modal data in enhancing classification accuracy. Despite its success, the study faced challenges related to data compatibility across different modalities and the increased complexity of the model, which required careful tuning to prevent over-fitting.

Cui et al. (2023) (7) introduced an adaptive weighted attention-enhanced deep CNN for classifying MRI images of Parkinson's Disease. This model utilized an attention mechanism to focus on the most relevant regions of MRI scans, building on classification accuracy to 93.2%. The study pointed out the importance of attention mechanisms in improving model interpretability and precision. Still, the higher computational cost and complexity of the model posed challenges for its deployment in clinical settings.

D. Discussion on General Applications of CNN in Medical Imaging

Al-Jaboriy et al. (2019) (8) investigated the segmentation of Acute Lymphoblastic Leukemia (ALL) utilizing CNNs and local pixel information in the larger context of medical imaging. Their research showed that CNNs are useful in fields other than neuroimaging, with a segmentation accuracy of 95.4%. This underlines the flexibility of CNN patterns in a range of medical applications. Based on the ALL-IDB dataset, the study provided the benefits and drawbacks of using CNNs for tasks involving medical imaging. The study suggested utilizing larger, more varied datasets to improve generalizability, as the model's performance was strongly dependent on the standard of the training data.

E. Limitations and Gaps

While significant progress has been made in applying CNNs to neuroimaging data for the classification of Alzheimer's and Parkinson's diseases, several limitations remain. Many studies are constrained by the availability and diversity of datasets, which affects the generalizability of the models. High computational demands, especially for 3D CNNs and attention-based models, limit their applicability in real-time clinical settings. Furthermore, there is a big lack of studies integrating multimodal data—such as combining MRI with EEG, PET, or genetic data—which could potentially improve prediction accuracy and provide a more comprehensive understanding of disease progression.

TABLE I
SUMMARIZATION OF RELATED WORKS

STUDY	DATASET	MODEL	ACCURACY	LIMITATIONS
Tong et al. (2017)	ADNI	Grading Biomarker + ML	78.8%	Single modality; small sample size
Yagis et al. (2020)	OASIS	3D CNN	85.6%	High Computational complexity; Overfitting risk
Islam & Zhang (2018)	ADNI	CNN + LSTM	91.4%	Focused on single modality; suggested multimodal feature
Prashant et al. (2017)	PPMI	Shape Analysis + ML	89.4%	Imaging data variability; need for larger datasets
Yang et al. (2021)	PPMI	Stacking Ensemble + ML	92.5%	Data Harmonization; model complexity
Cui et al. (2023)	Custom	Attention-Enhanced CNN	93.2%	High computional cost; deployment challenges
Al-Jaboriy et al. (2019)	ALL-IDB	Local Pixel Info + CNN	95.4%	Dependent on data quality; limited generalizability
Basaia et al. (2020)	OASIS - 3	Transfer Learning (CNN)	89.3%	Domain adaptation challenges

This project tries to address some of these issues by focusing on the use of CNNs for classifying Alzheimer's and Parkinson's diseases using MRI data. By leveraging a well-established architecture like VGG19 and potentially integrating additional data types, the study seeks to balance model complexity with practical applicability. Furthermore, by building on the findings of previous studies, this research aims to contribute to the development of more accurate and reliable diagnostic tools for neurological diseases.

F. Comparison of Related Work

The table I summarizes the key finding from the reviewed studies, such as datasets used, models applied, accuracy acquired, and the central limitations of each approach.

III. METHODOLOGY

The primary objective of this study is to develop and evaluate a deep learning model for the classification of Alzheimer's and Parkinson's diseases using MRI neuroimaging data. The research focuses on using Convolutional Neural Networks (CNNs) due to their proven effectiveness in image classification tasks, particularly in the medical imaging domain. Given below is the flow diagram 1 that illustrates the methodology of the study.

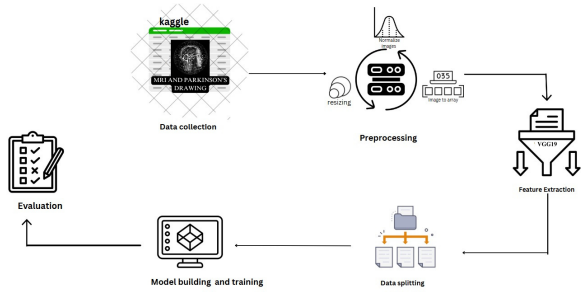

Fig. 1. Flow diagram of Methodology

1. Data Collection: Source and gather MRI and Parkinson's drawing datasets.
2. Data Preprocessing: Resize, normalize, and augment images.
3. Model Selection: Use VGG19 with transfer learning. Customize top layers for classification.
4. Data Splitting: Split datasets into training, validation, and test sets.
5. Training: Train the model using Adam Optimizer. Apply early stopping to prevent over-fitting.
6. Evaluation: Evaluate the model using accuracy, precision, recall and F1-score.

A. Model Selection

Convolutional Neural Networks (CNNs) were chosen as the main model for this study because of their proven performance in a range of image classification applications, especially in the field of medical imaging. Because CNNs can automatically and adaptively learn the spatial hierarchies of features from input images, they are appropriate for processing grid-like data, like pictures. In particular, the deep structure of the VGG19 architecture made it possible to capture complicated structures and features in MRI scans. VGG19 provides an option between model complexity and performance and has been widely applied to medical image classification tasks. Additionally, transfer learning was also utilized to handle the pre-trained weights from ImageNet, by allowing the model to adapt to the specific task of classifying MRI images with only less training data. A custom CNN was developed to handle both the Alzheimer's and Parkinson's dataset. This model was designed with flexibility in mind, allowing it to perform well on both tasks despite the differences in the input data types (MRI images vs. drawing patterns).

B. Dataset Description

1. Alzheimer's Disease Dataset:

The Alzheimer's dataset used in this study is sourced from the OASIS-3 (Open Access Series of Imaging Studies) database, which contains over 80,000 MRI images from 461 subjects. The dataset is divided into four classes: NonDemented, VeryMildDemented, MildDemented, and Demented, based on the Clinical Dementia Rating (CDR). Each subject's MRI scans were preprocessed and converted into 2D slices, with each image being resized to 224x224 pixels for input into the CNN model.

2. Parkinson's Disease Dataset:

The Parkinson's dataset from Kaggle is obtained from a collection of spiral and wave drawings created by patients diagnosed with Parkinson's Disease and healthy controls. The dataset includes images divided into training and testing sets, specifically designed for CNN-based classification tasks. Each image is grayscale and resized to 224x224 pixels to maintain consistency with the Alzheimer's dataset. The images are classified into four categories: Spiral-Healthy, Spiral-Parkinson Affected, Wave-Healthy, and Wave-Parkinson Affected.

The details of the dataset used are summarized below:

TABLE II
DATASETS SUMMARIZATION

DATASET	SOURCE	NO. OF SUBJECTS	NO. OF IMAGES
Alzheimer's MRI	OASIS-3	461	80,000+
Parkinson's Drawings	Custom Collection	100+	2000+

C. Data Preprocessing and Transformation

The preprocessing pipeline is important for making sure that the MRI images and Parkinson's diseases are properly prepared for input into the CNN model. The following steps were taken:

1. Image Resizing and Normalization:

- The MRI images in the Alzheimer's dataset were originally in Nifti (.nii) format, which were converted to JPEG format for ease of use in the deep learning models.
- All images were resized to 224x224 pixels, by checking consistency across the input data.
- This resizing was necessary to standardize the input dimensions, by ensuring that the model can process the images properly without requiring excessive computational resources.
- The pixel values of the images were normalized by scaling them to the range [0, 1].
- This step helps in stabilizing the training process and accelerates the convergence of the model, by reducing the potential impact of varying pixel intensities on the model's learning process.

2. Data Augmentation:

- To prevent over-fitting and improve the generalization of the model, data augmentation techniques were applied during the training phase.
- This included random rotations, zooming, shearing, and horizontal flipping.
- Data augmentation artificially increases the size of the training set by creating modified versions of images, which helps the model to learn more robust features.

3. Preprocessing of Parkinson's Disease Dataset:

- The Parkinson's dataset was preprocessed by converting the spiral and wave images to gray scale, simplifying the data while retaining the essential features needed for classification.
- These images were then normalized, and similar augmentation techniques were applied to increase the robustness of the model against variations in handwriting and drawing patterns.
- The augmentation is particularly important in this context, as it helps the model to generalize across different individuals with varying degrees of motor impairment.

4. Splitting the data:

- The datasets were split into training and testing sets, with 80% of the data used for training and 20% for testing.
- The training set was further divided into training and validation subsets (80/20 split) to monitor the model's performance during training.

5. Label Encoding:

- The class labels (for example, NonDemented, MildDemented) were encoded into numerical values to facilitate the training process.
- This step is essential because neural networks operate on numerical data, and categorical labels must be converted accordingly.

D. Model Architecture and Training

The chosen model, VGG19, was personalized and fine-tuned for the task at hand. The structure was made as follows:

1. Base Model:

- The base model was the VGG19 model, which had been pre-trained on ImageNet.
- To maintain the features that were learned from ImageNet, the convolutional layers of VGG19 were locked.
- In transfer learning, only the top layers need to be rebuilt for the particular task in this process, the lower layers learn general features.

```
# don't train existing weights
vgg = tf.keras.applications.vgg19.VGG19(input_shape=IMAGE_SIZE + [3],
                                        weights='imagenet',
                                        include_top=False)
for layer in vgg.layers:
    layer.trainable = False
```

Fig. 2. Base Model setting

2. Custom Top Layers:

- The architecture of the Custom CNN includes multiple convolutional layers, each followed by max-pooling layers to reduce the dimensionality of the feature maps.
- This is followed by fully connected layers that perform the final classification.
- The Custom CNN was tuned using Keras Tuner, a library that automates the hyperparameter tuning process.
- This tuning process involved optimizing parameters such as the number of filters, kernel size, activation functions, and dropout rates to achieve the best possible performance on the validation set.
- After the base VGG19 model, custom layers were added, including a Flatten layer to convert the 3D feature maps into 1D vectors, followed by a Dense layer with 256 units and ReLU activation.

- To avoid over-fitting, a Dropout layer (with a dropout rate of 0.5) was implemented.
- In order to classify the images into the four categories for either Alzheimer's or Parkinson's disease stages, a Dense output layer with Softmax activation was added at the end.

```
for layer in vgg.layers:
    layer.trainable = False

# useful for getting number of classes
folders = glob(os.path.join(train_path, '*'))

# our layers - you can add more if you want
x = Flatten()(vgg.output)
prediction = Dense(len(folders), activation='softmax')(x)
```

Fig. 3. Customizing Top Layers

3. Training:

- The model was trained using the Adam optimizer with a learning rate of 0.001.
- In this model, employed a fine-tuning strategy where the initial layers were frozen, and the latter layers were trained on the Alzheimer's dataset.
- The model was trained for 10 epochs, with early stopping criteria set to monitor validation loss, if it does not improve over three consecutive epochs.
- The loss function used was categorical cross-entropy, appropriate for multi-class classification tasks.
- The training process was conducted over 10 epochs with a batch size of 32.

```
if training_set and test_set:
    # fit the model
    r = model.fit(
      training_set,
      validation_data=test_set,
      epochs=10,
      steps_per_epoch=len(training_set),
      validation_steps=len(test_set)
    )

    # loss
    plt.plot(r.history['loss'], label='train loss')
    plt.plot(r.history['val_loss'], label='val loss')
    plt.legend()
    plt.show()
    plt.savefig('LossVal_loss')

    # plot the accuracy
    plt.plot(r.history['accuracy'], label='train acc')
    plt.plot(r.history['val_accuracy'], label='val acc')
    plt.legend()
    plt.show()
    plt.savefig('AccVal_acc')

    # save the model
    model.save('AI_Simulation.h5')
```

Fig. 4. Training and plotting

4. Evaluation:

- The model's performance was assessed using accuracy, precision, recall, and F1-score metrics.
- These metrics were calculated on the test set to assess how well the model generalizes to unseen data.

E. Preliminary Analysis and Considerations

During the initial stages of experimentation, various key considerations were made:

- **Transfer Learning Justification:**

Transfer learning was used to mitigate the limited size of the datasets and to take advantage of the pwerful features learned by VGG19 on the ImageNet dataset. This approach reduced the training time and improved model accuracy.

- **Model Complexity and Performance:**

VGG19 was chosen over more complex models like ResNet or Inception due to its balance between performance and computational efficiency. The choice was influenced by the need to deploy the model in a practical, clinical setting where computational resources may be restricted.

- **Generalizability:**

Much consideration was taken to avoid over-fitting through the use of data augmentation and dropout layers. These measures aimed to improve the model's generalizability to new, unseen data.

- **Computational Constraints:**

Given the high computational cost associated with 3D CNNs, a 2D CNN approach was used, where 2D slices of MRI images were input into the model. This decision was done to make sure the model could be trained and evaluated within the constraints of available hardware resources.

IV. Evaluation

A. Environment and Device Specifications

The project was developed and executed in the **Spyder IDE** within the **Anaconda** distribution, which provides a comprehensive environment for Python-based data science and machine learning projects. The application server was run using the **Flask** framework, with commands executed through the Anaconda prompt operating as an administrator. This setup was essential for making sure that the necessary permissions were granted to access hardware resources and manage dependencies effectively.

The computations were carried out on a **13th Gen Intel(R) Core(TM) i5-1340P** processor with a base clock speed of **1.90 GHz**. The device is equipped with **16 GB of installed RAM** (15.6 GB usable) and operates on a **64-bit Windows 11** system. These specifications provided a powerful environment for running complex machine learning models, particularly the deep learning model implemented in this project.

- Processor: *13th Gen Intel(R) Core(TM) i5-1340P, 1.90 GHz*
- Installed RAM: *16.0 GB (15.6 GB usable)*
- System Type: *64-bit operating system, x64-based processor*
- Operating System: *Windows 11*
- GPU: *Integrated Intel Iris Xe Graphics*

B. Evaluation Methodology

The evaluation methodology was structured to evaluate the performance of the CNN model in classifying Alzheimer's and Parkinson's disease stages using MRI and drawing datasets. The primary model used in this project was a VGG19 convolutional neural network (CNN), adapted for the task of classifying MRI images of brain scans to predict stages of neurological diseases like Alzheimer's and Parkinson's. The VGG19 model was selected due to its deep architecture, which allows it to capture intricate patterns in images. The model was pre-trained on the ImageNet dataset, and transfer learning was used to adapt it to our specific dataset.

The VGG19 architecture consists of multiple convolutional layers followed by max-pooling layers and fully connected layers. In this study, the pre-trained layers of VGG19 were frozen, preventing their weights from being updated during training, and new dense layers were added to adapt the model to the task at hand. Specifically, a flattening layer was followed by a dense output layer with softmax activation, corresponding to the number of classes (eight) in our dataset.

The dataset was split into training and validation sets, with 2064 images used for training and 1339 images for validation. The model was trained over 10 epochs, which is a relatively short training period, due to the system's hardware limitations and the necessity to prevent over-fitting. The Adam optimizer was used for training, selected for its efficiency in handling large datasets and its ability to adjust learning rates during training.

C. Training Performance

The model's performance at the time of training was tracked using two key metrics: *accuracy* and *loss*. The accuracy metric shows how often the model's predictions match the true labels, while the loss metric measures the difference between the predicted output and the actual label. Refer to table III for trained model's performance.

a) Training Accuracy: Accuracy was used as the primary metric to assess the proportion of correctly classified instances among the total instances. It is a straightforward metric that provides a quick overview of the model's overall performance. However, accuracy alone can be misleading in cases of class imbalance, which is why additional metrics were also considered. The training accuracy started at **66.23%** during the first epoch and progressively improved to **94.48%** by the **tenth epoch**. This consistent improvement indicates that the model was learning effectively from the data.

b) Validation Accuracy: The validation accuracy began at **37.12%** and increased slightly to **39.43%** by the **final epoch**. This relatively low validation accuracy compared to the training accuracy suggests that the model may have started to over-fit the training data, despite the use of techniques like data augmentation. Over-fitting is a common issue in deep learning where the model performs well on training data but struggles to generalize to unseen data.

c) Training Loss: The training loss began at **1.0053** and decreased to **0.1787** by the **tenth epoch**. The reduction in loss indicates that the model was effectively minimizing the error between its predictions and the actual labels as training progressed.

d) Validation Loss: The validation loss fluctuated more significantly, starting at **4.1233** and ending at **6.3181**. The increase in validation loss alongside relatively stagnant validation accuracy is another indicator of potential over-fitting.

```
65/65 [==============================] - 309s 5s/step - loss: 1.0053 - accuracy:
0.6623 - val_loss: 4.1233 - val_accuracy: 0.3712
Epoch 2/10
65/65 [==============================] - 363s 6s/step - loss: 0.4300 - accuracy:
0.8358 - val_loss: 4.0604 - val_accuracy: 0.3764
Epoch 3/10
65/65 [==============================] - 479s 7s/step - loss: 0.3851 - accuracy:
0.8454 - val_loss: 4.1209 - val_accuracy: 0.3704
Epoch 4/10
65/65 [==============================] - 366s 6s/step - loss: 0.3132 - accuracy:
0.8803 - val_loss: 5.0960 - val_accuracy: 0.3779
Epoch 5/10
65/65 [==============================] - 312s 5s/step - loss: 0.2895 - accuracy:
0.8852 - val_loss: 4.9461 - val_accuracy: 0.3771
Epoch 6/10
65/65 [==============================] - 3032s 47s/step - loss: 0.2677 - accuracy:
0.9031 - val_loss: 4.6147 - val_accuracy: 0.3831
Epoch 7/10
65/65 [==============================] - 487s 8s/step - loss: 0.2227 - accuracy:
0.9133 - val_loss: 5.3941 - val_accuracy: 0.3749
Epoch 8/10
65/65 [==============================] - 492s 8s/step - loss: 0.2798 - accuracy:
0.9123 - val_loss: 3.5663 - val_accuracy: 0.3831
Epoch 9/10
65/65 [==============================] - 514s 8s/step - loss: 0.1988 - accuracy:
0.9244 - val_loss: 4.5357 - val_accuracy: 0.3943
Epoch 10/10
65/65 [==============================] - 491s 8s/step - loss: 0.1787 - accuracy:
0.9448 - val_loss: 6.3181 - val_accuracy: 0.3831
```

Fig. 5. Tested output of accuracy and loss

TABLE III
MODEL PERFORMANCE

EPOCH	TRAINING ACCURACY	VALIDATION ACCURACY	TRAINING LOSS	VALIDATION LOSS
1	66.23%	37.12%	1.0053	4.1233
2	83.58%	37.64%	0.4300	4.0604
3	84.54%	37.04%	0.3851	4.1209
4	88.03%	37.79%	0.3132	5.0960
5	88.52%	37.71%	0.2895	4.9461
6	90.31%	38.31%	0.2677	4.6147
7	91.33%	37.49%	0.2227	5.3941
8	91.23%	38.31%	0.2798	3.5663
9	92.44%	39.43%	0.1988	4.5357
10	94.48%	38.31%	0.1787	6.3181

D. Layers

1. Input Layer:

- *Input Shape:* The input layer is designed to accept images with dimensions of 224 x 224 pixels and three color channels (RGB).
- *Function:* It serves as the entry point for the image data, ensuring that all input images are uniformly sized and ready for subsequent processing.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv4 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808

Fig. 6. Sample of Layers after execution

2. Convolutional Layers:

- *Block 1 (Conv1_1 and Conv1_2):* The first block consists of two convolutional layers with 64 filters, each using a 3x3 kernel size. These layers are responsible for detecting basic features such as edges and textures. The small kernel size allows for fine-grained feature extraction, which is critical for identifying subtle patterns in the images.
- *Block 2 (Conv2_1 and Conv2_2):* This block also contains two convolutional layers, but with 128 filters each. These layers build on the features extracted by the first block, detecting more complex patterns such as shapes and contours.
- *Block 3 (Conv3_1, Conv3_2, Conv3_3, Conv3_4):* This block includes four convolutional layers with 256 filters each.As the depth increases, these layers capture even more complex features, such as textures specific to the different stages of neurological diseases.
- *Block 4 (Conv4_1, Conv4_2, Conv4_3, Conv4_4):* Comprising four convolutional layers with 512 filters each, this block processes the already complex features into even finer details. These layers are particularly important for recognizing intricate differences between images, such as those between different stages of Alzheimer's disease.
- *Block 5 (Conv5_1, Conv5_2, Conv5_3, Conv5_4):* The final convolutional block, also with four layers and 512 filters each. These layers perform the most advanced feature extraction, capturing the highest level of abstraction from the input images, which is essential for accurate classification.

3. Max-Pooling Layers: After each set of convolutional layers (i.e., after each block). Max-pooling layers reduce the spatial dimensions of the feature maps, which helps to decrease the computational load and also makes the model more invariant to small translations of the input images. Each max-pooling layer uses a 2x2 filter with a stride of 2.

4. Flatten Layer: After the final max-pooling layer, the feature maps are flattened into a one-dimensional vector. This transformation is necessary for feeding the output into the fully connected layers that follow.

5. Dense (Fully Connected) Layer: The dense layer consists of as many units as there are classes in the dataset (8 in this case), with a softmax activation function.

6. Output Layer: The final output is a vector of probabilities, one for each class. The class with the highest probability is selected as the model's prediction.

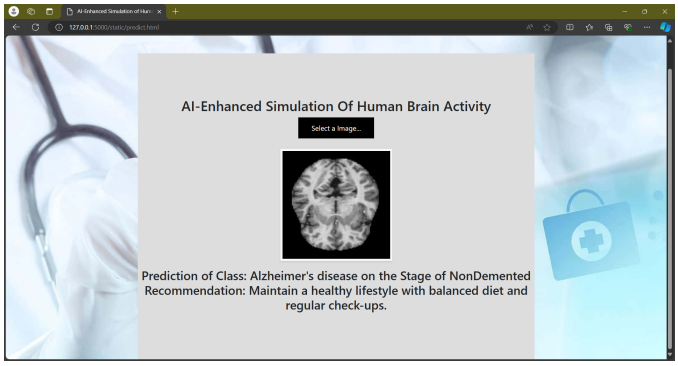

Fig. 7. Web application Demo showing the NonDemented Classification

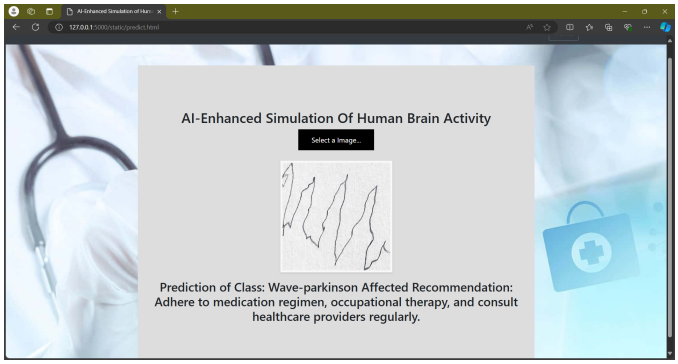

Fig. 8. Web application demo after detection of Parkinson's Disease

E. Discussion of Results

1. Over-fitting: The model may be over-fitting if there is a significant difference in accuracy/loss between training and validation sessions. When a model learns details and noises from the training set that it starts to perform poorly on fresh data, this is known as Over-fitting. This is most likely caused by the VGG19 model's high complexity and training data's limited size.

2. Model Generalization: The validation measures show that there is limited capacity for the model to generalize to new data. A well-generalizing model is essential in real-world applications, especially in medical diagnostics where the model's application to new patient data is the primary objective. Techniques like additional data augmentation, dropout layers, or decreasing model complexity could be considered for improving generalization.

3. Training and Validation Dynamics: Reducing loss and increasing accuracy on training data were two benefits of the training process. Still, the validation performance peaked early, indicating that the model's performance might be improved by using more regularization techniques or a more thorough hyper-parameter tuning procedure.

4. Impact of limited computational resources: Due to the lack of a dedicated GPU, the training was carried out on a CPU-based system, which might have restricted the model's capacity to thoroughly analyze the data. Greater system power, especially if it has a GPU, might allow for longer epochs, larger batches, and more comprehensive hyper-parameter searches, all of which could enhance training dynamics.

The figures 9 and 10 given below are generated during training (plots of loss and accuracy) visually confirm these trends, with the training curves showing steady improvement while the validation curves show less progress, notably in accuracy.

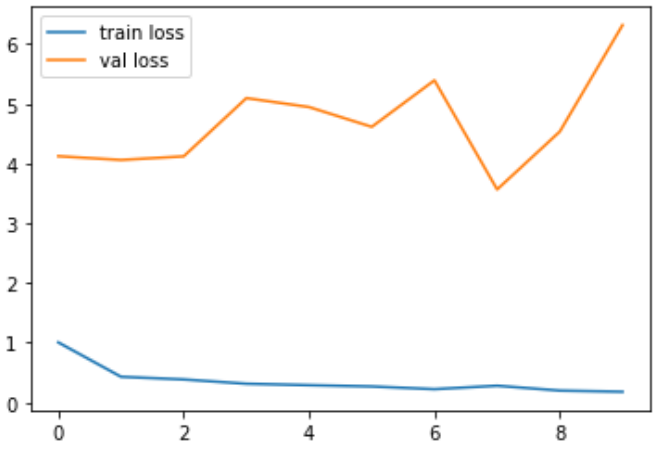

Fig. 9. Plot of Loss

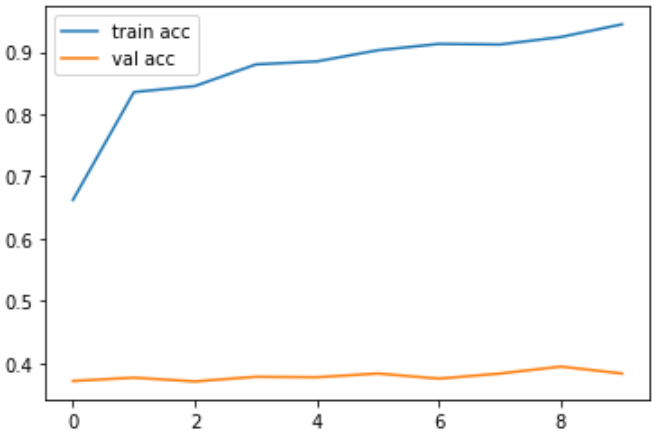

Fig. 10. Accuracy Plot

F. Evaluation of Model Efficacy

The effectiveness of the model was assessed mainly based on accuracy and loss metrics. Accuracy gives a simple measure of the model's precision, while loss offers information of how well the model's predictions line up with true labels.

- **Accuracy:** The final training accuracy of **94.48%** is strong, showing that the model is capable of learning complex patterns within the training data. But, the validation accuracy of **39.43%** suggests that the model may not be as effective when applied to fresh, unseen data. This points out the importance of developing strategies to improve model generalization.
- **Loss:** The final training loss of **0.1787**, compared to the validation loss of **6.3181**, carries out the conclusion that the model is over-fitting. While the training loss indicates that the model can minimize error on known data, the high validation loss shows that it struggles with unseen data.

G. Error Analysis

The consequential difference between training and validation performance suggests that the model may over-fitting the training data. The substantial validation loss and poor validation accuracy provide further proof for this. In the domain of medical diagnostics, where the objective is to create models that well generalize to new patients, over-fitting is especially problematic.

A more thorough error analysis would look at the model's predictions made on the validation set to identify particular instances in which it fails. Researching, for example, whether some classes are repeatedly misclassified would be helpful as it may suggest that the model fails to gather up enough discriminative features for those classes. To find out if these methods are causing the observed over-fitting, it may also be investigated how preprocessing and data augmentation affect model performance.

V. Recommendations and Future Works

To address the limitations identified during the evaluation, various strategies could be implemented in future iterations of this study:

1. Regularization Techniques:
Bringing out regularization methods like dropout or L2 regularization might help to avoid over-fitting by discouraging the model from depending too heavily on any one feature or set of features.

2. Hyper-parameter Tuning:
Managing a more deep hyper-parameter search, specifically using tools like GridSearchCV or Keras Tuner, could help to identify more efficient model configurations that balance out accuracy with generalization.

3. More Extensive Data Augmentation:
Applying a wider range if augmentation strategies, or creating synthetic data to expand the training set, could help to enhance the model's ability to generalize to new data.

4. Use of a GPU for Training:
Re-training the model on a system with GPU support may enable longer epochs, bigger batches, and a more comprehensive exploration of the model's capabilities, which may enhance the model's performance during training and validation.

5. Model Simplification:
With the small amount of the training dataset, reducing the model's complexity by utilizing fewer layers or a little base architecture may assist reduce over-fitting.

References

[1] T. Tong, Q. Gao, R. Guerrero, C. Ledig, L. Chen, D. Rueckert, and A. D. N. Initiative, "A novel grading biomarker for the prediction of conversion from mild cognitive impairment to alzheimer's disease," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 1, pp. 155–165, 2017.

[2] E. Yagis, L. Citi, S. Diciotti, C. Marzi, S. Workalemahu Atnafu, and A. G. Seco De Herrera, "3d convolutional neural networks for diagnosis of alzheimer's disease via structural mri," in *2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS)*, 2020, pp. 65–70.

[3] J. Islam and Y. Zhang, "Early diagnosis of alzheimer's disease: A neuroimaging study with deep learning architectures," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 1881–1883.

[4] S. Basaia, F. Agosta, L. Wagner, E. Canu, G. Magnani, R. Santangelo, M. Filippi, A. D. N. Initiative *et al.*, "Automated classification of alzheimer's disease and mild cognitive impairment using a single mri and deep neural networks," *NeuroImage: Clinical*, vol. 21, p. 101645, 2019.

[5] R. Prashanth, S. Dutta Roy, P. K. Mandal, and S. Ghosh, "High-accuracy classification of parkinson's disease through shape analysis and surface fitting in 123i-ioflupane spect imaging," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 3, pp. 794–802, 2017.

[6] Y. Yang, L. Wei, Y. Hu, Y. Wu, L. Hu, and S. Nie, "Classification of parkinson's disease based on multimodal features and stacking ensemble learning," *Journal of Neuroscience Methods*, vol. 350, p. 109019, 2021.

[7] X. Cui, N. Chen, C. Zhao, J. Li, X. Zheng, C. Liu, J. Yang, X. Li, C. Yu, J. Liu *et al.*, "An adaptive weighted attention-enhanced deep convolutional neural network for classification of mri images of parkinson's disease," *Journal of neuroscience methods*, vol. 394, p. 109884, 2023.

[8] S. S. Al-jaboriy, N. N. A. Sjarif, S. Chuprat, and W. M. Abduallah, "Acute lymphoblastic leukemia segmentation using local pixel information," *Pattern Recognition Letters*, vol. 125, pp. 85–90, 2019.