Summer 2023

An Explainable Deep Learning Prediction Model for Severity of Alzheimer's Disease From Brain Images

Godwin O. Ekuma
Missouri State University, goe948s@missouristate.edu

AN EXPLAINABLE DEEP LEARNING PREDICTION MODEL FOR SEVERITY OF ALZHEIMER'S DISEASE FROM BRAIN IMAGES

A Master's Thesis

Presented to

The Graduate College of

Missouri State University

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science, Computer Science

By

Godwin Ekuma

August 2023

AN EXPLAINABLE DEEP LEARNING PREDICTION MODEL FOR SEVERITY OF ALZHEIMER'S DISEASE FROM BRAIN IMAGES

Computer Science

Missouri State University, August 2023

Master of Science

Godwin Ekuma

ABSTRACT

Deep Convolutional Neural Networks (CNNs) have become the go-to method for medical imaging classification on various imaging modalities for binary and multiclass problems. Deep CNNs extract spatial features from image data hierarchically, with deeper layers learning more relevant features for the classification application. The effectiveness of deep learning models are hampered by limited data sets, skewed class distributions, and the undesirable "black box" of neural networks, which decreases their understandability and usability in precision medicine applications. This thesis addresses the challenge of building an explainable deep learning model for a clinical application: predicting the severity of Alzheimer's disease (AD). AD is a progressive neurodegenerative disorder that affects the brain and could result in dementia. Early detection of AD is crucial for more precise treatment and enhanced patient outcomes. The diagnosis and prognosis of AD rely heavily on neuroimaging information, particularly Magnetic Resonance Imaging (MRI). The research developed a deep learning model framework that integrates a local data-driven interpretation method (SHapley Additive exPlanation values) to explain the relationship between the predicted AD severity from the neural network and the input MR brain image. This thesis addresses the skewed class distribution using the synthetic minority oversampling technique. To evaluate the performance of the proposed framework, the study performed a comparative analysis using three CNN models: DenseNet121, DenseNet169, and Inception-ResNet-v2. The framework shows high sensitivity and specificity in the test sample of subjects with varying levels of AD severity. To facilitate a better understanding of model performance, five key AD neurocognitive assessment outcome measures and the APOE genotype biomarker were correlated with model misclassifications.

KEYWORDS: deep learning, convolutional neural network, Alzheimer's disease, magnetic resonance imaging, transfer learning, data augmentation, explainability, classification, medical imaging.

AN EXPLAINABLE DEEP LEARNING PREDICTION MODEL FOR SEVERITY OF ALZHEIMER'S DISEASE FROM BRAIN IMAGES

By

Godwin Ekuma

A Master's Thesis
Submitted to the Graduate College
Of Missouri State University
In Partial Fulfillment of the Requirements
For the Degree of Master of Science, Computer Science

August 2023

Approved:

Tayo Obafemi-Ajayi, Ph.D., Thesis Committee Chair

Siming Liu, Ph.D., Committee Member

Mukulika Ghosh, Ph.D., Committee Member

Julie Masterson, Ph.D., Dean of the Graduate College

ACKNOWLEDGEMENTS

I would like to acknowledge everyone who contributed to my academic accomplishments. First, my wife and two sons, who supported me with love and understanding. Without you, I could never have reached this current level of success.

Secondly, Dr. Tayo Obafemi-Ajayi (thesis committee chair), members of my thesis committee, and Dr. Daniel Hier (domain expert) each of whom has provided patient advice and guidance throughout the research process. Thank you all for your unwavering support.

TABLE OF CONTENTS

Introduction 1

Background 5
- Deep Learning in a Nutshell 5
- Convolutional Neural Network 7
- Basic Structure of a Convolutional Network 8
- Examples of CNN Architectures 10
- Transfer Learning 16
- Overview of Magnetic Resonance Imaging 17

Literature Review 19
- CNN-Based Alzheimer's Disease Severity Prediction 19
- Explainable AI Methods for CNN-Based AD Severity Prediction 21

Methodology 24
- Data Acquisition and Preprocessing 24
- Class Balancing 27
- Construction of CNN Learning Model 28
- Model Validation 31
- Model Explainability 32
- Clinical Relevance Assessment 33

Experimental Results and Analysis 35
- Experimental Setup 35
- Performance Comparison of the DenseNet121 vs DenseNet169 vs Inception-ResNet-v2 36
- Comparison of Results with other Existing Models Reported in Literature 39

Model Explanation and Clinical Relevance 42

Conclusion and Future Work 47

References 48

LIST OF TABLES

Table 1. Demographic and overall neurocognitive assessment of study sample. 25

Table 2. Performance of the CNN models on the AD severity prediction based on the test dataset. 38

Table 3. Performance comparison of proposed model with prior models. 41

Table 4. Clinical relevance of Inception-Resnet-V2 model outcomes for correctly predicted AD vs. AD predicted as MCI or CN 43

Table 5. Clinical relevance of Inception-Resnet-V2 model outcomes for correctly predicted MCI vs. MCI predicted as AD or CN 46

LIST OF FIGURES

Figure 1. Alzheimer's disease progression stages. 2

Figure 2. Structure of multilayer perceptron with one hidden layer. 7

Figure 3. Illustration of the architecture of a CNN showing convolution, pooling, and fully connected (dense) layers. 8

Figure 4. Graphical depiction of convolution operation with a 2x2 filter. 9

Figure 5. The DenseNet framework features links connecting every layer to all the following layers. 11

Figure 6. CNN model architecture for DenseNet 121 and DenseNet 169. The main differences lie in the size of the dense blocks. 12

Figure 7. CNN model architecture for Inception-ResNet-v2. 14

Figure 8. Schema Illustrations for Inception-ResNet blocks (a) Inception-ResNet-A, (b) Inception-ResNet-B, and (c) Inception-ResNet-C modules. 15

Figure 9. Illustration of MRI variations (a) TI-weighted MR image. (b) T2-weighted MR image 18

Figure 10. Proposed learning framework for AD detection and severity classification 24

Figure 11. MRI preprocessing pipeline: (a) Raw brain MRI. (b) Skull stripping. (c) Spatial normalization. (d) Center axial slice images in severity order: CN, MCI, AD. 26

Figure 12. Explainable learning framework for prediction of AD severity. FC: Fully connected layer. ReLU: Rectified linear unit activation function 29

Figure 13. Performance of CNN model on the validation set during training for varying epoch levels across the 5-fold cross-validation (a) DenseNet121 (b) DenseNet169 (c) Inception-ResNet-v2 (d) accuracy of models on validation vs. test data. 37

Figure 14. AUC ROC curve for (a) DenseNet121 (b) DenseNet169. 40

Figure 15. Visualization of prediction for MRIs labeled AD (a) correctly predicted as AD and (b) incorrectly predicted as MCI or CN. 44

Figure 16. Visualization of prediction for MRIs labeled MCI (a) correctly predicted as MCI and (b) incorrectly predicted as AD or CN. 45

INTRODUCTION

Deep learning is a subfield of machine learning which involves building computational models to train artificial intelligence systems. Unlike most traditional machine learning models, deep learning models can automatically extract relevant input features without expert knowledge. Automated feature engineering makes deep learning advantageous in fields like bioinformatics and medical imaging, where determining relevant features is difficult. Advancement in the use of technology and electronic health records in the medical field over the past few decades has led to a tremendous increase in data availability to mine for knowledge discovery and prediction models. The data available spans a wide range from clinical to biospecimen and various imaging data. The use of diverse imaging modalities, such as magnetic resonance imaging (MRI), positron emission tomography (PET), cerebrospinal fluid (CSF), and computerized tomography (CT) scans, has resulted in major strides in the diagnosis and treatment of many diseases including neurological disorders [1]. Deep learning models are used to diagnose various neurological disorders, such as Alzheimer's disease and traumatic brain injury [2] [3] [4, 3], through extensive analysis of these imaging modalities.

This thesis investigates the use of deep learning models (specifically convolutional neural networks) for medical imaging translational research in predicting Alzheimer's disease (AD) severity. AD is a neurological disorder caused by brain nerve cell damage, resulting in language, memory, and thinking problems and eventually leading to dementia [1]. Although these problems may seem new to the person experiencing them, the changes in their brain that led to these symptoms start occurring around 20 years before any noticeable symptoms [1]. AD accounts for 60-80% of dementia cases, and age is the most significant known risk factor for developing Alzheimer's, with most cases occurring in individuals over 65 [5]. As shown in Figure 1,

Alzheimer's disease develops gradually, starting with minor changes in the brain and progressing to more severe symptoms that affect memory and physical abilities. This progression have three broad stages: preclinical Alzheimer's disease or clinically normal (CN) stage, mild cognitive impairment (MCI) due to Alzheimer's disease, and dementia (AD) due to Alzheimer's. The latter stage is characterized by significant symptoms that impede daily functioning and accompanied by evidence of Alzheimer's-related brain changes on laboratory tests [5].

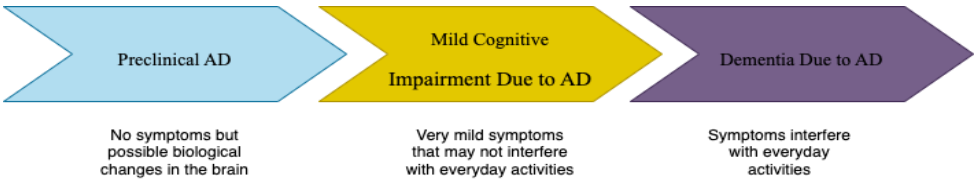


Figure 1. Alzheimer's disease progression stages.

In 2014, there were about 5.0 million Americans aged 65 and older who had dementia, this number is expected to increase to 13.9 million by 2060. The cost of caring for elderly individuals with AD and other types of dementia is significantly greater than for those without these conditions. AD-related diseases incured a cost of $355 billion in the United States in 2021 [1]. Consequently, it is crucial to develop computer-based tools that can assist physicians in accurately diagnosing and predicting the survival prospects of AD patients.

In the past decade, multiple deep learning models have been proposed for the prediction of AD severity and progression of disease [2] [3] [6] [7]. Though strides have been made, some limitations need to be addressed as follows:

i) Insufficient training data/unbalanced distribution of classes: Training a DL model effectively requires a large amount of annotated learning data, which can be challenging in the

medical imaging. Due to ethical considerations, physician-annotated data is costly, unbalanced in class ratio, time-consuming to collect, and often not permitted for cross-institutional use. When insufficient data is available, or the class ratio in the dataset is unbalanced, deep learning models tend to be overfitted and less accurate [3].

ii) Choosing the right deep learning model and fine-tuning its hyperparameters: There is no standardized method for selecting a deep learning algorithm that is suitable for your data [8]. Moreover, deep learning tools are not universally applicable since they rely on hyperparameters (such as weight, number of neurons, activation function, optimizer, learning rate, batch size, and epochs) that must be optimized for each dataset to achieve optimal performance. These hyperparameters require careful adjustment to maximize the predictive accuracy of the model [8] [9].

iii) Lack of interpretability: Interpretability is a critical issue in bioinformatics since it affects the adoption of deep learning methods in medicine and the safety of patients. Deep learning models are often regarded as 'black box' algorithms that offer little insight into how they arrive at their results, which can leave users and even developers with little understanding of the underlying process [10]. As a result, there is growing skepticism about the potential of AI, even though interest in it is increasing, and sometimes, over-optimism prevails.

This study seeks to overcome current limitations by developing an interpretable deep learning model capable of accurately predicting AD severity using brain MRI scans. We perform a comparative analysis of three models) to determine the best-performing model for AD prediction while also addressing the challenges posed by limited and unbalanced class ratios associated with the available data. The learning framework will integrate transfer learning, a technique that improves learning in a new task through the transfer of knowledge from a related task that has already been learned [11]. This method is commonly used to enhance the performance of deep neural networks models in the absence of abundant training data and can also aid in optimizing hyperparameters such as learning rate and batch size to improve the model's performance on the target task [9]. In summary, the contributions of this thesis are:

i) Formalize a pipeline to process raw or semi-processed brain MR images for use in a DL model.

ii) Construct a deep learning framework based on CNN models that integrates synthetic minority oversampling technique (SMOTE) to learn and predict the severity of AD from an input brain MR image. We conduct a comparative analysis of three CNN models to evaluate the effectiveness of the proposed framework.

iii) Integrate a local model interpretation method that uses SHapley Additive exPlanation (SHAP) [12] values to explain the relationship between the predicted AD severity and input brain image. In addition, we correlate the prediction performance on the images with five AD neurocognitive assessment measures and the APOE genotype biomarker to further explain and interpret the model's outcome for increased understandability.

This thesis is structured as follows. In the background section, we provide a brief overview of the field of deep learning, its application in neuroimaging as well as a description of the CNN models and key theoretical concepts needed to understand this research work. The literature review section summarizes current techniques for classifying AD severity through deep neural networks. Subsequently, we present our proposed framework for AD severity prediction based on brain MRI scans in the methodology section. The experiments and results section provides a comprehensive analysis of our study's findings, including the performance of our model. We interpret the results using a deep explainer and analysis of cognitive assessment data, and finally we conclude by stating the limitations of our approach and future research directions.

BACKGROUND

In this section, we briefly described key concepts, including convolutional neural networks, deep-dense and inception-residual neural networks, transfer learning, and class balancing utilized in this work to provide some context.

Deep Learning in a Nutshell

Deep learning (DL) is a subfield of artificial intelligence (AI) that originated from machine learning (ML). AI aims to teach machines to learn and automate intellectual tasks that are usually performed by humans without explicit programming [13]. In AI systems, machines are considered rational actors that aim to achieve the best possible outcome or one they believe will achieve it [3]. ML focuses on developing methods for training intelligent systems by exposing them to input and output data relevant to a task. Feature extraction is a crucial aspect of ML models that identifies parameters with predictive power and reduces input data dimensionality. However, feature extraction can be time-consuming and may not produce accurate results. DL approaches have been developed to overcome this limitation by allowing for the automatic detection and interpretation of high-level features from input data [3]. DL eliminates the feature extraction step from the training process, which saves time and effort for both machine learning experts and domain experts.

Deep neural network architectures, commonly known as DL models, can be achieved by increasing the size of the hidden layers in a neural network. DL models consist of multiple layers that progressively learn more meaningful representations of data. The term "deep" refers to the incorporation of numerous layers, leading to a greater depth of the model [13]. In contrast, traditional ML approaches typically rely on learning only one or two layers of representation,

hence the term "shallow learning." DL methods automatically learn various layers of representations from training data, using neural network (NN) models to build complex, layered representations. According to Kohonen [1], NNs are interconnected networks of simple elements that emulate the functioning of a biological nervous system, and their hierarchical organization enables them to understand and respond to real-world objects.

The perceptron algorithm is a type of single-layer neural network used for binary classification, and it was one of the earliest NN models inspired by biological neurons [2]. The human brain processes information through millions of interconnected neurons that transmit signals via excitation and inhibition. The perceptron emulates this behavior using an activation function to determine the neuron's active or inactive state and a few weights that learn to classify patterns by adjusting themselves. However, the perceptron's limited ability to recognize complex patterns led researchers to develop NNs with multiple layers of the perceptron to solve more intricate nonlinear problems. A multilayer perceptron (MLP), depicted in Figure 2, contains an input layer, at least one hidden layer, and an output layer, with multiple neurons in each layer. Neurons in adjacent layers are interconnected, while those in the same layer are independent [3].

The MLP training process consists of two stages: forward propagation and backpropagation. In forward propagation (also known as the forward pass), the input is passed through the network in a forward direction to produce an output. Each hidden layer receives the input neuron, a corresponding weight, and threshold and then applies the activation function to determine if the neuron is active or inactive. The backpropagation algorithm updates the weights of the MLP based on its performance relative to the predicted values, evaluated by comparing the difference between the MLP's output and actual labels using an appropriate loss function and optimizer.

There exist various categories of DL architectures, such as convolutional neural networks (CNN) [14], recurrent neural networks [14], Restricted Boltzmann Machines [14], autoencoders [15], and generative adversarial networks [16]. We utilize CNN models in this work, hence we present a brief overview of CNNs in particular.

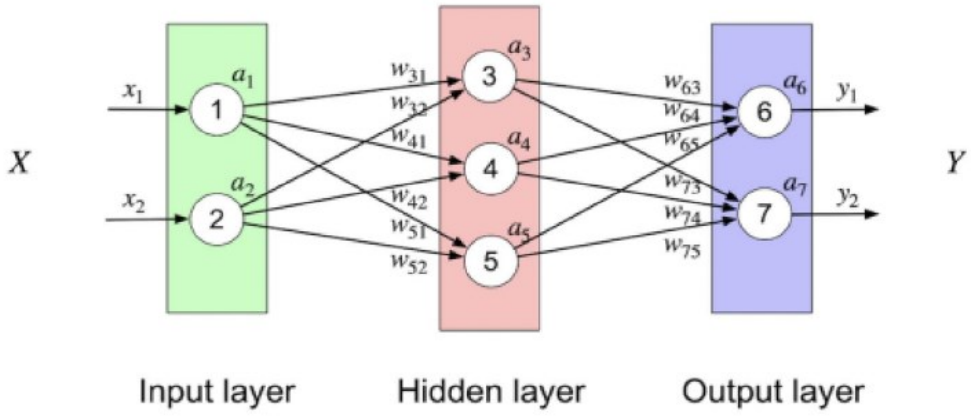


Figure 2. Structure of multilayer perceptron with one hidden layer. Adapted from Figure 1 in [3, 7].

Convolutional Neural Network

Convolutional neural networks were designed to handle inputs that are structured in a grid format and exhibit significant spatial dependencies within local regions of the grid [14]. An example of such grid-structured data is a 2D image. One of the key attributes that distinguish a CNN from other deep learning models is their utilization of a mathematical operation known as convolution. Convolution involves the computation of a dot product between a set of weights arranged in a grid-like pattern and input data that is also structured in a grid format, with data drawn from various spatial locations within the input volume. As a result, convolutional neural networks are characterized as networks that employ the convolution operation at least once, with most CNNs utilizing this operation across multiple layers [14].

Basic Structure of a Convolutional Network

In a CNN architecture (illustrated in Figure 3), it is typical to find four primary types of layers, namely convolution, pooling, activation, and fully connected (FC) layers. These layers are all structured as 3-dimensional grids characterized by their height, width, and depth. Furthermore, each layer is linked to a specific parameter called a filter or kernel (usually 3-dimensional but smaller than the layer) and produces output referred to as feature maps or activation maps.

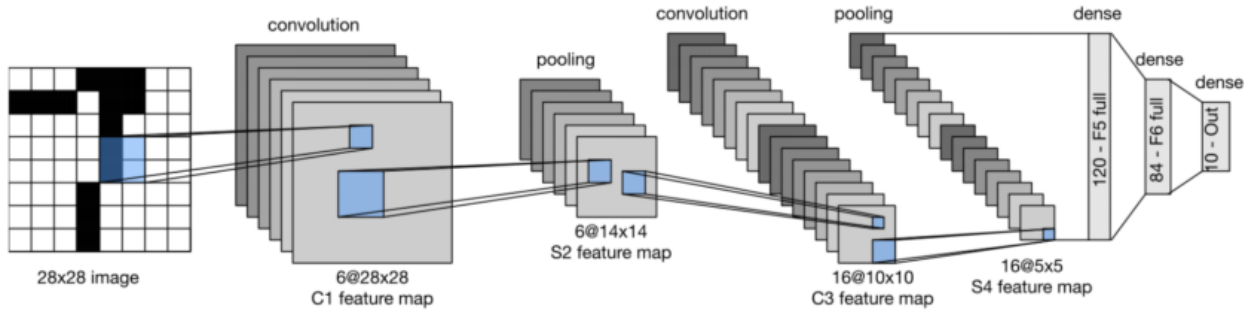


Figure 3. Illustration of the architecture of a CNN showing convolution, pooling, and fully connected (dense) layers. Adapted from [17]

Convolutional Layer. The convolutional layer performs the convolution operation. Suppose that the input for the q^{th} layer has dimensions of $L_q \times B_q \times d_q$, where L_q represents the height or length, B_q represents the width or breadth, and d_q represents the depth. Also, assume that $F_q \times F_q \times d_q$ is the filer of the q^{th} Layer. During the convolution process, the filter is positioned at each possible location within the image or hidden layer to ensure that it completely covers the image, and a dot product is then computed between the $F_q \times F_q \times d_q$ parameters within the filter and the corresponding grid of data in the input volume. To carry out this dot product, the relevant entries within the 3-dimensional area of the filter and input volume are

viewed as vectors of equal size [14]. As shown in Figure 4, in the forward pass, each filter will move from left to right first, then up to down. The distance between each move is called stride.

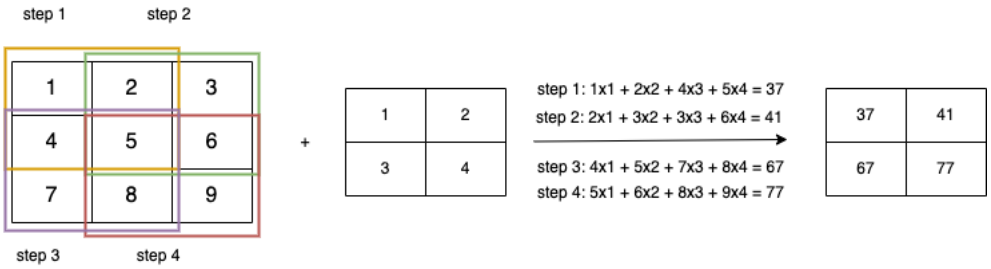


Figure 4. Graphical depiction of convolution operation with a 2x2 filter.

Activation Layer. The activation layer applies the activation function to the feature maps. Rectified linear unit (ReLU) activation function is a commonly used activation function for CNN. It is applied at the end of a hidden unit or convolution to introduce nonlinear complexities and create $L_q \times B_q \times d_q$ thresholded values [14]. The resulting values are transmitted to the subsequent layer. Since ReLU only involves a straightforward one-to-one relationship between activation values, the dimensions of a layer remain unaltered as a result of this operation.

Pooling Layer. Pooling is an operation that operates on small regions of a layer, typically in the form of a square grid with dimensions of $P_q \times P_q$. The resulting layer generated from pooling operations will maintain the same depth as the original layer, in contrast to filters. During the pooling process, the average or highest value among the values present in each square region of $P_q \times P_q$ within the d_q activation maps is selected. If a stride of 1 is utilized, the resulting layer's dimensions will be $(L_q - P_q + 1) \times (B_q - P_q + 1) \times d_q$ [14]. Pooling operation substantially decreases the spatial dimensions of every activation map.

Fully Connected Layer. The final layer in a CNN is the FC or dense layer. Every feature map in the final spatial layer is connected to each neuron in the first FC layer in a dense fashion. It is common to use multiple FC layers to enhance the computational capacity toward the end of a CNN. FC layers are designed to suit the problem at hand. The FC layer for a classification problem is different from that of a regression or segmentation problem [14]. In a classification task, the output layer (last FC layer) is connected to every neuron in the second to last layer and has a weight assigned to it. The type of activation function used on this layer, whether logistic, softmax, or linear, depends on the specific application, such as classification or regression [14].

Examples of CNN Architectures

Several CNN models have been developed over the years that have won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [18], which evaluates algorithms for object detection and image classification at a large scale. Examples of these models include VGGNet [19], ResNet [20], Inception v3 [21], Inception-ResNet-v2 [22], DenseNet [23], AlexNet [24], and others. This study utilizes two different architectures of DenseNet (121 and 169) and Inception-ResNet-v2. To provide a better understanding of these architectures and their relevance to our study, we present a brief overview of each network.

DenseNet: As CNNs become deeper, input or gradient information passing through many layers can diminish and "wash out" before reaching the network's end [23]. To address this issue, the dense convolutional network (DenseNet) was created with dense connections linking all layers directly to each other and enabling the flow of information across layers and preventing gradients from vanishing. Each layer receives additional input from all preceding layers and

passes its own feature maps to all subsequent layers, as shown in Figure . Concatenation is used to combine features, so the ℓth layer has ℓ inputs consisting of feature maps from all preceding convolutional blocks, and its feature maps are passed to all L-ℓ subsequent layers. This creates L(L+1)/2 connections in an L-layer network, unlike traditional CNN architectures that only have L connections. The essential components of ImageNet DenseNet include the dense block, transition layers, and growth rate parameters. The dense block contains multiple layers of densely connected 1x1 and 1x3 convolution operations, with each operation preceded by batch normalization and a ReLU unit. Transition layers, located between dense blocks, downsample feature maps, reduce the number of channels, and control parameter growth using 1x1 convolution and 2x2 average pooling. The growth rate parameter regulates the number of feature maps a layer receives from its preceding layers.

The DenseNet model has several variations, and this study focuses on DenseNet121 and DenseNet169, offering additional context on these models. Figure 6 illustrates the structural difference between these two models.

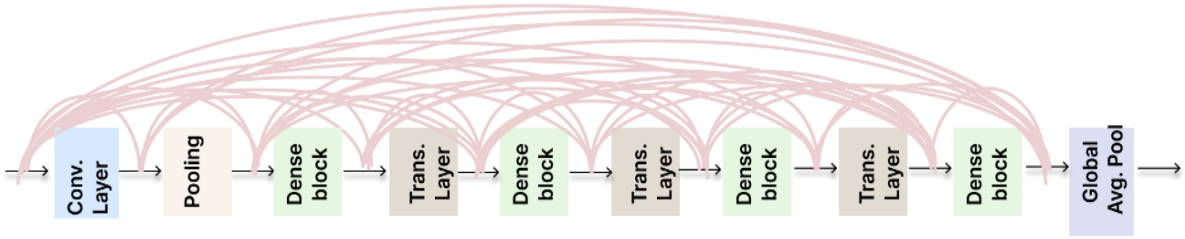


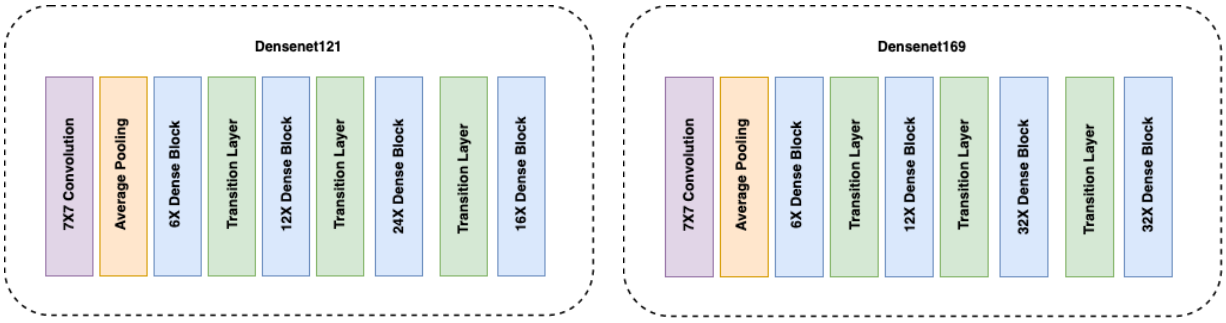
Figure 5. The DenseNet framework features links connecting every layer to all the following layers. Variants like DenseNet121 and DenseNet169 differ in the number of dense layers within the blocks. "Conv." implies "Convolution," while "Trans." denotes "Transition.

DenseNet121. The DenseNet-121 architecture (depicted in Figure 6a) is composed of four dense blocks, each consisting of multiple convolutional layers, batch normalization, and a

rectified linear unit (ReLU) activation function. Each dense block is followed by a transition layer that performs a down-sampling of the feature maps and reduces the number of channels.

The first layer of the DenseNet-121 model is a convolutional layer with 64 filters and a kernel size of 7x7, followed by a max pooling layer with a stride of 2. This is followed by the four dense blocks, each containing several convolutional layers, batch normalization, and ReLU activation. The last dense block is followed by a global average pooling layer, which aggregates the feature maps into a single vector. Finally, there is a fully connected layer with Softmaxactivation that outputs the predicted class probabilities. DenseNet-121 has approximately 7 million parameters, and to its relatively small size, it is also a popular choice for real-time applications where computational resources are limited.

Densenet169. DenseNet-169 (depicted in Figure 6b) consists of 4 dense blocks, each containing several convolutional layers with small 3x3 filters. The number of filters in each dense block is 256, 512, 1280, and 1664, respectively. Each dense block is followed by a transition layer, which consists of a batch normalization layer and a 1x1 convolutional layer that reduces the number of channels and downsamples the spatial resolution of the feature maps. The



(a) DenseNet 121 (b) DenseNet 169

Figure 6. CNN model architecture for DenseNet 121 and DenseNet 169. The main differences lie in the size of the dense blocks.

down-sampling factor is set to 2 in the first three transition layers, while the last transition layer does not perform downsampling.

The first layer of the DenseNet-169 model is a convolutional layer with 64 filters and a kernel size of 7x7, followed by a max pooling layer with a stride of 2. The final output of the network is produced by a global average pooling layer that aggregates the feature maps into a single vector, followed by a fully connected layer with Softmaxactivation that outputs the predicted class probabilities. DenseNet-169 has approximately 14 million parameters, which is larger than the number of parameters in DenseNet-121. This allows DenseNet-169 to achieve higher accuracy on image classification benchmarks but also requires more computational resources and longer training times.

Inception-ResNet. The inception residual network (Inception-ResNet) neural network architecture is an extension of two other neural network architectures: the Inception network and the Residual network (ResNet). The Inception network uses parallel convolutional layers of varying sizes to handle the trade-off between the width and depth of the network. Meanwhile, ResNet introduced residual connections to improve the training convergence by addressing the vanishing gradient problem.

Inception-ResNet combines these two architectures by using the Inception module for feature extraction and ResNet's residual connections to improve the gradient flow during training. The Inception module comprises parallel branches of different convolutional layers, and the outputs are concatenated and fed to the next layer. This module enables the network to capture features at different resolutions and scales, making it ideal for image classification tasks. In addition, Inception-ResNet uses residual connections to allow gradients to propagate directly

to earlier layers, bypassing intermediate layers that could weaken them, which further improves the training convergence of the network.

Inception-ResNet has two variations, including Inception-ResNet-v1 and Inception-ResNet-v2. Although the working principles of Inception-ResNet v1 and Inception-ResNet v2 are similar, Inception-ResNet-v2 has higher accuracy but also comes with a higher computational cost compared to the Inception-ResNet-v1 network. The work utilizes the Inception-ResNet-v2 shown in figure 7. The Inception-residual network is composed of 3 main blocks, including:

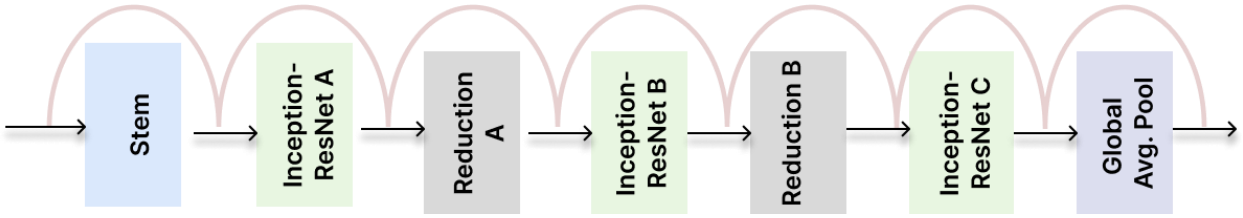


Figure 7. CNN model architecture for Inception-ResNet-v2.

Stem Block. The stem block of Inception-ResNet-V2 is the initial part of the network that processes the input image. It consists of a series of convolutional layers, followed by pooling layers, which downsample the spatial dimensions of the input. The stem block is designed to extract low-level features such as edges and corners from the input image.

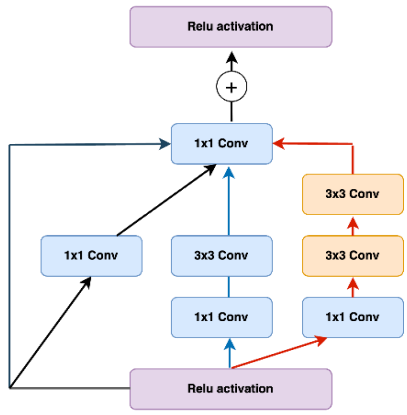
Inception-ResNet Block. The Inception-ResNet block is a key building block of the Inception-ResNet-V2 architecture. It consists of multiple branches, each with its own set of convolutional and pooling layers. These branches are concatenated together to form a composite feature map that captures information at multiple scales. The Inception-ResNet block is designed to allow the network to learn rich representations of the input that are robust to variations in the image. It contains several types of Inception-ResNet blocks, which are illustrated in Figure 8:

i. Inception-ResNet-A block: This block is composed of seven convolution operations and three parallel branches. Each of the convolutions applies a different kernel size (1x1, 3x3, 1x3, and 3x1) to the input. The outputs of these branches are concatenated and passed through a bottleneck layer before being added to the input using a residual connection.

ii. Inception-ResNet-B block: This block is similar to the Inception-ResNet-A block with three parallel branches but with five convolutions with different kernel sizes (1x1, 1x7, and 7x1).

iii. Inception-ResNet-C block: This block is similar to the Inception-ResNet-B block, but it replaces the 1x7 and 7x1 convolutions with 1x3 and 3x1 convolutional layers.

(a)



(b)

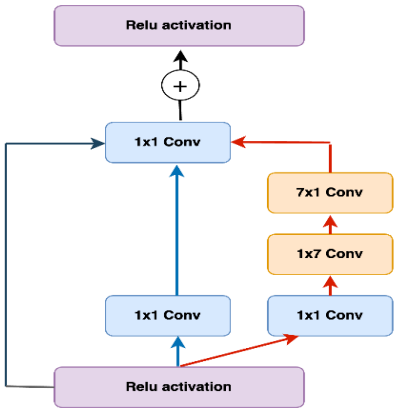


Figure 8. Schema Illustrations for Inception-ResNet blocks (a) Inception-ResNet-A, (b) Inception-ResNet-B.

(c)

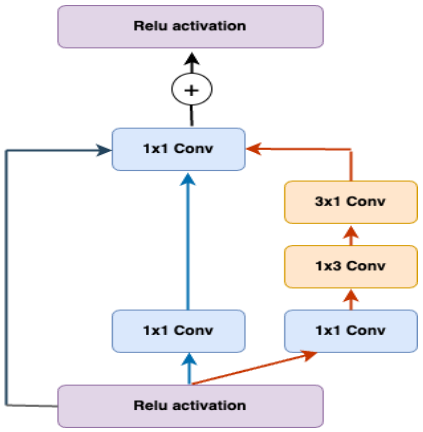


Figure 8 continued. Schema Illustrations for Inception-ResNet blocks (c) Inception-ResNet-C modules

Reduction Block. The reduction blocks in Inception-ResNet-V2 are used to reduce the spatial dimensions of the feature maps produced by the Inception-ResNet blocks. This is done to decrease the computational cost of the network and improve its efficiency. The reduction blocks typically involve a combination of pooling and convolutional layers and are designed to preserve the information in the feature maps while reducing their size. There are two sets of reduction blocks in the Inception-ResNet-v2:

i. Reduction-A block: This block is used to reduce the spatial dimensionality of the input. It consists of a 3x3 convolutional layer with stride two followed by a 3x3 max pooling layer.

ii. Reduction-B block: This block is also used to reduce the spatial dimensionality of the input. It consists of a 1x1 convolutional layer, a 3x3 convolutional layer with stride 2, and a 3x3 max pooling layer.

Transfer Learning

Transfer learning is a technique used to transfer knowledge learned in one or more source tasks and use it to improve learning in a related target task [25]. Transfer learning aims to

enhance learning in the target task by utilizing knowledge from the source task. Transfer can improve learning through three main measures. The first measure is the initial performance that can be achieved in the target task by using only the transferred knowledge, without any further learning, in comparison to the initial performance of an untrained agent. The second measure is the time it takes to completely learn the target task with the transferred knowledge compared to the time it takes to learn it from scratch. The third measure is the final level of performance that can be achieved in the target task with transfer compared to the final level without transfer [11]. Fine-tuning is a common technique in transfer learning, which involves the following steps [26]:

i. Create a CNN model by copying the weights and parameters of a pre-trained source model.

ii. Unfreeze some of the top layers in the pre-trained source model. In deep CNNs, the layers become more specialized as they go higher up. The initial layers learn basic and universal features that can be applied to various types of images, while the higher layers become more tailored to the specific dataset that the model was trained on. The purpose of fine-tuning is to adjust these specialized features to function with the new dataset instead of replacing the general learning.

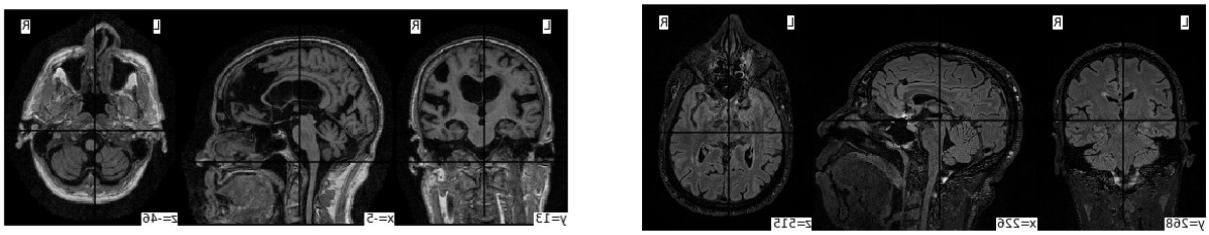
iii. Add new trainable layers and an output layer with the number of outputs that correspond to the number of categories in the target dataset.

iv. Train the target model on the target dataset. The new trainable and the output layers are trained from scratch, while the parameters of all other layers are fine-tuned based on the parameters of the source model.

Overview of Magnetic Resonance Imaging

MRI is a diagnostic technique that utilizes non-invasive imaging technology to detect various diseases and structural anomalies. It is extensively employed in the early detection and assessment of various brain-related illnesses, such as Alzheimer's disease, stroke, multiple sclerosis, and brain tumors [27]. MR images come in three different orientations: axial, sagittal,

and coronal. As illustrated in Figure 9, there are also several types of contrast images. T1-weighted MRI amplifies the signal of fatty tissue and diminishes the signal of water, while T2-weighted MRI amplifies the signal of water. In this thesis, we utilize the T1-weighted MRI as it is best for best at showing the anatomy of the brain tissues that would be learned by the DL models.



(a) (b)

Figure 9. Illustration of MRI variations (a) TI-weighted MR image. (b) T2-weighted MR image

LITERATURE REVIEW

CNN-Based Alzheimer's Disease Severity Prediction

Alzheimer's disease classification using deep learning and MRI has been widely researched by scholars. Yousry et al. [27] proposed a 5-layer end-to-end deep learning framework that utilizes scanned MRI to predict whether a patient has AD and to what degree. The framework includes a CNN model composed of four convolutional layers, and max pooling is performed after each convolutional layer. The authors performed two different AD classification experiments, binary and multiclassification, and evaluated the effect of various learning parameters on disease classification, including sample size, batch size, activation functions, loss functions, dropout, and augmentation of input samples. The framework achieved high training accuracy, 99.8% for binary classification and 97.5% for multiclassification, with large image sizes and dropouts helping to improve classification accuracy.

Murugan et al. [28] proposed a DEMentia NETwork (DEMNET) that uses MRI images for the early diagnosis of Alzheimer's disease and dementia. The authors of the paper used a dataset from the Kaggle open-source platform that comprises four stages of dementia: mild dementia, moderate dementia, non-demented, and very mild dementia. The novelty of their work is the use of a series of DEMNET blocks for the extraction of discriminative features. A DEMNET block consists of two 2D convolutions plus ReLU activation functions, followed by batch normalization and max pooling. The model used the SMOTE to balance class sizes and achieved a testing accuracy of 95.23%, AUC of 97%, and Cohen's Kappa value of 0.93.

In [29], Jain et al. describe a CNN based approach for AD classification from MR brain images. The authors used the VGG16 architecture as the base model for transfer learning and added a fully connected layer and a dropout layer, followed by a two-way classification layer for

binary classification or a three-way classification layer for multiclass classification. They evaluated their approach on MRI data from the Alzheimer's Disease Neuroimaging Initiative (ADNI) data repository and achieved a validation accuracy of 95.73% for three-way classification. The authors conclude that transfer learning can be used to build a classification model with comparable performance to a model trained from scratch.

Lee et al. [30] proposed a deep CNN with a data permutation scheme for the classification of AD using structural MRI data. They employ a data augmentation strategy that involves randomly permuting the voxels of the sMRI data to generate more samples for training the CNN. They also use an outlier detection and removal technique to further improve the quality of the training data. Their CNN model consists of eight convolutional layers and three fully connected layers. They evaluate their model using a dataset from ADNI and report high classification accuracy for both binary and multiclass classification tasks. They conclude that their approach, which combines deep learning with data augmentation and outlier detection, can improve the accuracy of Alzheimer's disease classification using sMRI data.

Basaia et al. [31] presented an automated method for the classification of AD and MCI using a single MRI scan and CNN. The authors used a large dataset of MRI scans from ADNI to develop a CNN model that can differentiate between AD, MCI, and healthy control (HC) participants. The CNN model was trained using a transfer learning approach on a pre-trained VGG16 network, which was then fine-tuned on the ADNI dataset. The study found that the proposed method achieved high accuracy in distinguishing between AD, MCI, and HC participants, with an overall accuracy of 87.15%, sensitivity of 91.96%, and specificity of 82.52%. The authors suggested that this method has the potential to improve the accuracy of AD diagnosis and to aid in the early detection of the disease.

Helaly et al. [32] proposed a deep learning-based approach for the early detection of AD. The approach involves preprocessing of MRI images using various techniques, such as skull stripping and registration, to enhance the features of the brain. The proposed method also uses CNN with three convolutional layers and a fully connected layer to classify MRI scans as either normal or AD. The dataset used for training and testing the model consists of 576 MRI scans, with 288 scans from normal individuals and 288 from AD patients. The proposed model achieved an accuracy of 98.6%, sensitivity of 98.7%, and specificity of 98.5%, demonstrating the potential of the deep learning approach for early detection of AD.

In [33] Kokkalla et al. proposed a DL model based on Inception-ResNet-v2 for the classification of brain tumors into three categories: meningioma, glioma, and pituitary tumor. The proposed model uses a combination of dense blocks and inception blocks, which are connected with residual connections to improve training and reduce overfitting. The authors also employ data augmentation techniques to improve the generalization of the model. The proposed model was trained and evaluated on a dataset of brain MRI scans, achieving an accuracy of 98.3% on the test set. The results show that the proposed model outperforms other state-of-the-art deep learning models for brain tumor classification. The authors suggest that their model could be used in clinical settings to assist radiologists in the accurate and efficient diagnosis of brain tumors.

Explainable AI Methods for CNN-Based AD Severity Prediction

Explianable AI pertains to methods and techniques for creating AD applications that end-users can comprehend and interpret [34]. Lada et al. [35] proposed a comprehensive framework for interpreting ML models in neuroimaging based on three assessment levels: model level,

feature level, and biology level. Model-level assessment evaluates the model as a whole, the feature-level assessment identifies significant features for prediction, and the biology-level assessment aims to prove neuroscientific plausibility. These assessment levels can be classified into two main categories: ante-hoc, which focuses on understanding how the model generates its results, and post-hoc, which centers on elucidating why the model produces a particular outcome [10].

Many explanation methods at the feature level focus on attributing a deep learning model's prediction to its input features. Visualization techniques are used to provide a clear explanation by highlighting important regions in input images or internal features that strongly influence the outputs [36]. These methods can be classified into two categories: propagation-based and perturbation-based. Propagation-based methods involve backpropagating gradients through the network to identify the most significant features in the output. Some well-known methods in this category are Class Activation Map (CAM), Activation Maximization, DeconvNet, and Layer-wise Relevance Propagation (LRP). In contrast, perturbation-based methods modify the input and monitor the resulting output changes to identify effective features. The Occlusion Map is a popular method in this category. Some attribution-based methods exhibit both propagation and perturbation characteristics [36]. Examples of such methods are Grad-CAM and Ablation-CAM [36]. Bae et al. [37] developed a 3D-CNN to predict conversion from MCI to AD from sMRI and identified structural brain regions that contributed to the conversion using Occlusion Map. Bron et al. [38] validated the generalizability of AD classification in the prediction of conversion from MCI to Alzheimer's using CNN (and Support Vector Machines). They further visualized the regions that contributed to the classifications using a guided backward propagation approach. Chakraborty et al. [39] performed multiclass classification of

Alzheimer's using 3D-CNN and showed that the model pays attention to the important regions using the CAM explanation technique. Sudar et al. [40] trained a VGG-16-based CNN for AD classification and identified stages of AD using Layer-wise Relevance Propagation. Shojaei et al. [41] performed binary classification of Alzheimer's using 3D-CNN from MRI and proposed a combination of Occlusion Map and Backpropagation-based AI explanation methods to extract the most important brain regions in Alzheimer's.

Novelty of Proposed Approach. While these studies have shown high accuracy in classification, to the best of the scope of this research, no studies have performed any AD-related studies that provide an explanation at the neurocognitive and/or biological assessment level for their models. A neurobiological assessment can validate a model by evaluating its plausibility based on converging evidence from other types of neuroscientific data [35], such as biomarkers and neurocognitive assessment scores. To address this issue, we validate model predictions with APOE genotype data and standardized neurocognitive clinical assessment scores for AD diagnosis. Furthermore, previous studies that have explained the features of a model have mostly used perturbation or propagation algorithms. We utilize a unified model attribution framework that combines several explanations [36] methods (such as LIME, DeepLift, and LRP) to attribute an effect to each feature and sum the effects to produce SHAP values that explain the relationship between predicted severity class and input brain MRI.

METHODOLOGY

This section explains the overall learning framework applied to AD classification in this study. Figure 10 shows that it comprises six phases: data acquisition and preprocessing, class balancing with SMOTE, construction of CNN learning model framework, model validation, model explainability, and clinical relevance assessment.

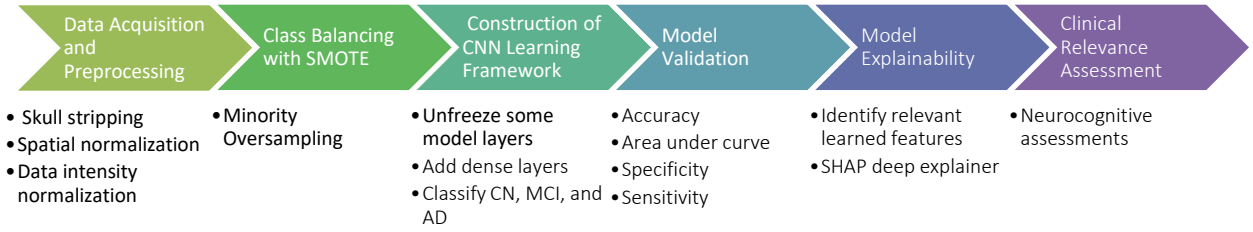


Figure 10. Proposed learning framework for AD detection and severity classification

Data Acquisition and Preprocessing

This thesis examined a sample of patients from the Alzheimer's Disease Neuroimaging Initiative (ADNI) data repository [42]. The database is a collection of data from a multicenter study that aims to develop biomarkers for the early detection and tracking of Alzheimer's disease. The database includes multiple cohorts of patients, each with different types of neuroimaging data that have undergone various acquisition and preprocessing phases. A subset of T1-weighted MR images from the ADNI-1 cohort that had the largest sample of 3D images processed with the N3 correction standard in the same image acquisition plane were chosen. Figure 11a shows a sample raw and unprocessed MRI image. The sample included 325 images from patients with Alzheimer's disease, 595 images from cognitively normal patients, and 1024 images from patients with mild cognitive impairment. This totaled 1944 images from 488 unique

patients, some of which were taken on different visit dates. All participants were between 50 and 80 years old, and a brief demographic description of the images by class is presented in Table 1.

Table 1. Demographic and overall neurocognitive assessment of study sample.

(n)	AD (325)	MCI (1024)	CN (595)
Male/Female	149/176	588/436	276/316
Age	72.5 ± 5.4	72.1 ± 5.5	74.7 ± 3.7
Education (years)	14.2 ± 3.1	15.6 ± 3.0	15.9 ± 2.9
MMSE	21.1 ± 5.4	25.7 ± 3.7	29.2 ± 1.0
CDR Global	17.3 ± 7.8	7.2 ± 7	0.2 ± 1.0
FAQ	1.0 ± 0.5	0.6 ± 0.3	0 ± 0.1.0

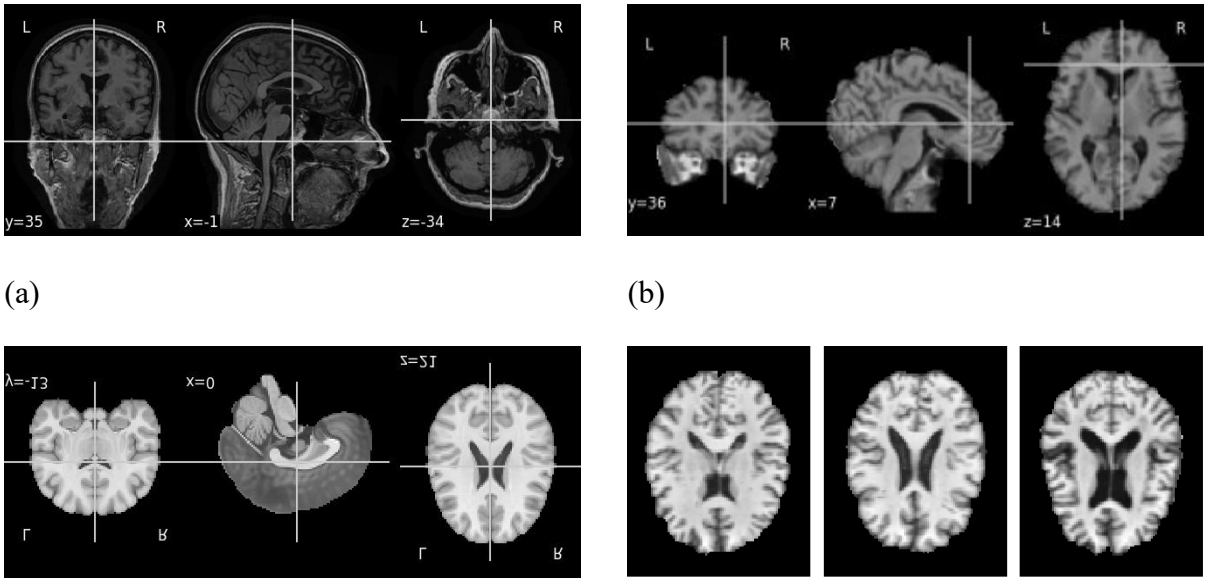
MMSE: Mini-Mental State Examination; CDR: Clinical Dementia Rating. FAQ: Functional Assessment Questionnaire.

Variations in MRI scans, such as magnetic field strength, image resolution, contrast, orientation, and patient positioning, can impact model performance. Therefore, preprocessing to standardize all subjects and imaging modalities is a crucial step that affects prediction performance. Skull stripping, spatial normalization, data intensity normalization, and image slicing are used to preprocess the images.

Skull Stripping. Skull stripping is performed to eliminate non-brain tissue voxels such as skin, fat, muscle, neck, and eyeballs from the images. Several skull-stripping methods exist in literature [43]. We utilize the functional MRI software library (FSL) brain extraction tool (BET) in this work [44]. Figure 11b illustrates the output image after applying BET to the initial image obtained from ADNI.

Spatial Normalization. Spatial normalization normalizes for image orientation and voxel spacing, which may vary between images, even when acquired from the same scanner. The main objective of this preprocessing is to limit the variations in positioning, orientation, shape, and size of the images in our study. We used the FSL's linear image registration tool [45] to linearly register all scans to the T1 MNI 152 Template with 2mm Isotropy. The spatial normalization preprocessing process outputs 3D image files with a uniform size of 91x109x109. Figure 11c illustrates the output of spatial normalization of an MRI to a template.

Image Slicing. To prepare the 3D image for a 2D convolutional neural network (CNN), each sample is sliced into sagittal, coronal, and axial orientations using the Nibabel [46] image slicer. Each sample produced three sets of 2D images, with the center slice of the axial orientation selected for further processing. Figure 11d displays the central slices of CN, MCI, and AD patients.



(a) (b)

(c) (d)

Figure 11. MRI preprocessing pipeline: (a) Raw brain MRI. (b) Skull stripping. (c) Spatial normalization. (d) Center axial slice images in severity order: CN, MCI, AD.

Data Intensity Normalization. This is also known as voxel-based normalization. It is used to standardize the pixel intensity values so that they fall within a specific range. It involves transforming the pixel values to the axial slices such that the mean of all values is 0 and the standard deviation is 1. This can improve the accuracy and reliability of subsequent image analysis using the CNN models.

Class Balancing

When collecting medical data, a common problem is the imbalance between healthy controls and cases. Although the overall patient distribution in the ADNI-1 cohort is relatively even, the set of 1944 MR images collected was heavily skewed towards 52.6% MCI, 30.6% CN, and 16.7% AD. To prevent overfitting and biased learning models due to this imbalance, we used SMOTE [47] to balance the class samples. This approach generates synthetic examples in feature space by oversampling the minority class. It achieves this by introducing examples along the line segments that connect any or all of the k nearest neighbors from the minority class. The SMOTE process is described in Equation 1, where a feature vector X_0 is selected from the minority class, and one of its *k* nearest neighbors X is randomly chosen. The difference between X and $X_0$is computed, and a new synthetic data point is generated at a random point in the line segment by connecting the feature vector and the selected neighbo, using a uniform random variable w in the range [0,1]. The neighbors from the *k* nearest neighbors are selected randomly depending on the desired level of oversampling.

$$Z = X_0 + w(X - X_0) \quad (1)$$

Construction of CNN Learning Model

The model for predicting AD severity from MR brain images is composed of a CNN learning model and an explainability extension, as shown in Figure 12. Three CNN architectures are evaluated, namely DenseNet121, DenseNet169, and Inception-ResNet-v2. To overcome the long training times of CNN models on images, transfer learning is utilized to enhance efficiency. This is achieved by using pre-trained models developed from standard computer vision benchmark data as a starting point for training new models on a related problem. The base model is initialized with Imagenet training weights, and some higher layers are unfrozen to allow the CNN model to encode more subtle features from the brain MR images. The fine-tuning layers consist of a global average pooling layer, two dense layers with ReLU activation function (Equation 2), L1 regularization (Equaton 4), and a dropout layer, and a fully connected layer with the softmax activation function for the three classes. The RMSprop (Equations 6 -7) optimizer is employed to minimize the categorical cross-entropy loss function (Equation 3).

Given an input value *x*, the ReLU activation function *f* defined by Equation 2, is the maximum of that element, and 0:

$$f(x) = max(0, x) \tag{2}$$

Given a true label vector y_{true} and a predicted probability distribution vector y_{pred}, the categorical cross entropy loss function is defined by Equation 3:

$$H(y_{true}, y_{pred}) = -\sum_{i=1}^{C} y_{true,i}\, log(y_{pred,i}) \tag{3}$$

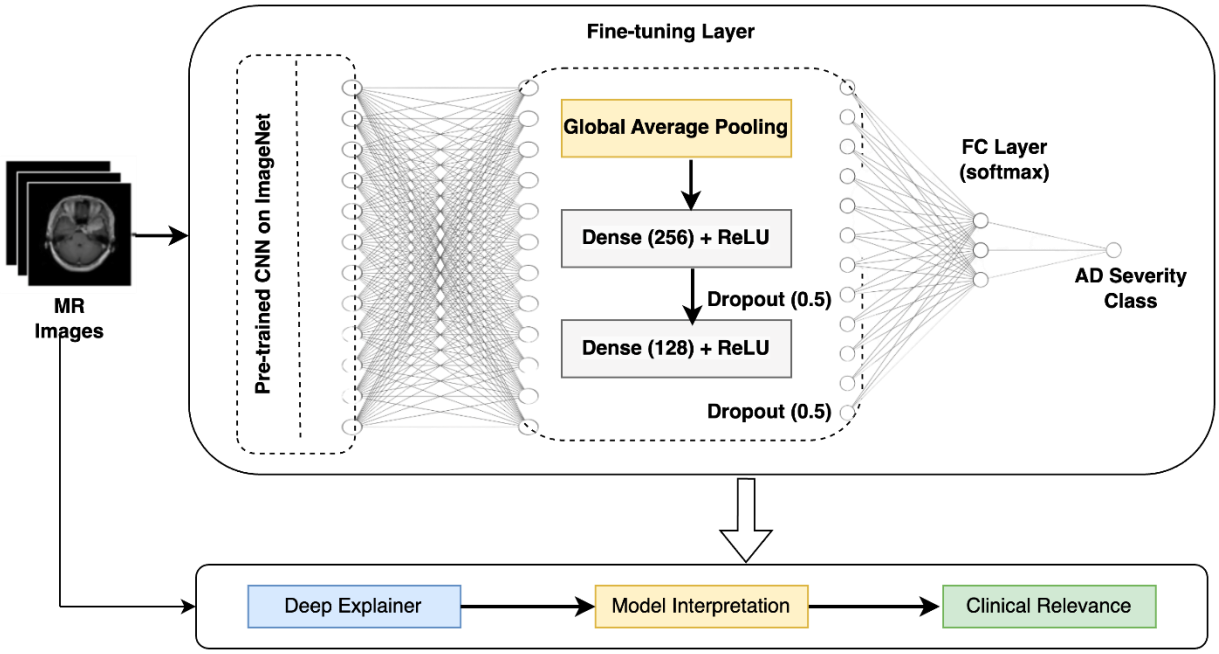


Figure 12. Explainable learning framework for prediction of AD severity. FC: Fully connected layer. ReLU: Rectified linear unit activation function

Where c is the number of classes, $y_{true,i}$ is the true label for the i-th class (1 if the sample belongs to that class, 0 otherwise), and $y_{pred,i}$ is the predicted probability of the sample belonging to the i-th class, computed by the softmax activation function.

L1 regularization (also known as Lasso regularization) is a technique used to prevent overfitting in machine learning models by adding a penalty term to the loss function that encourages the model to use fewer features. The effect of the L1 regularization penalty is to force many of the weights to become exactly zero, effectively removing them from the model. This leads to a simpler and more interpretable model with fewer features, which is less prone to overfitting. The L1 regularization penalty term is defined in Equation 4 as the sum of the absolute values of the model's weights multiplied by a regularization parameter λ:

$$\text{L1 regularization penalty } = \lambda * \Sigma|w| \quad (4)$$

Here, w is the weight parameter of the model, Σ is the sum of all weights, and λ is a hyperparameter that controls the strength of the regularization. The L1 regularization penalty term is added to the model's prediction error. The complete loss function with L1 activity regularization for our model is defined in Equation 5:

$$Loss\ Function = H(y_{true}, y_{pred}) + \lambda * \Sigma|w| \quad (5)$$

The RMSprop algorithm also aims at minimizing the loss function. It achieves this goal by computing an exponentially weighted moving average of the squared gradients and uses this to normalize the gradient update step. The update process for RMSprop can be described mathematically as follows:

Initialize the exponentially weighted moving average of the squared gradients $s = 0$. For each iteration of the optimization algorithm, compute the gradient of the loss function with respect to the model parameters as $g_t = \nabla_\theta J(\theta_t)$. Update the gradient of the loss function with respect to the model parameters using Equation 6:

$$s_t = \beta s_{t-1} + (1 - \beta) g_t^2 \quad (6)$$

Where β is a hyperparameter that controls the weighting of the past gradients in the moving average. Also, compute $\Delta\theta_t$ The using Equation 7:

$$\Delta\theta_t = -\frac{\eta}{\sqrt{s_t + \epsilon}} g_t \quad (7)$$

Where η the learning rate and ϵ is a small constant to prevent division by zero. Finally, the model parameters using Equation 8:

$$\theta_{t+1} = \theta_t + \Delta\theta_t \quad (8)$$

Model Validation

To assess the performance of the model, various metrics are utilized, including classification accuracy, sensitivity/recall, specificity, confusion matrix, and the area under the curve and receiver operating characteristic curve (AUC-ROC). Accuracy measures how effectively the model can predict true positives and negatives within the classes, and it can be calculated using Equation 9:

$$Accuracy = \left(\frac{TP + TN}{TP + TN + FP + FN}\right) \quad (9)$$

Where TP means the true positive, TN means the true negative, FP means the false positive, and FN means the false negative. The sensitivity, also known as true positive rate (TPR), indicates the model's ability to locate all positive samples. The formula for calculating sensitivity is given by Equation 10:

$$Sensitivity = \left(\frac{TP}{TP + FN}\right) \quad (10)$$

Specificity, also known as false positive rate (FPR), is the opposite of specificity. It is the ability of the model to locate all negative samples. Sensitivity is mathematically defined in Equation 11:

$$Specificity = \left(\frac{TN}{TN + FP}\right) \tag{11}$$

The ROC curve is a graphical representation of the relationship between sensitivity and specificity. AUC (average value of sensitivity for all possible values of specificity) is a measure of how well the model performs in predicting which image group it belongs to. As AUC values increase, this indicates better overall diagnostic performance for the model when it's applied to predicting each image group.

Model Explainability

Shapley additive explanations (SHAP) [12] is implemented as a data-driven local interpretation method to unravel the black box deep learning model [48]. The core idea of SHAP is based on the Shapley value, which is an important concept in game theory. Through the transfer of concepts from game theory, the outcomes of 'games' can be considered analogous to the predictions generated by ML models, and the 'players' of games can be considered analogous to the features in ML models. By calculating the Shapley value, the contribution of each 'player' to the outcome of the 'game' can be quantified just as the contribution of each feature to the ML prediction can be calculated.

The Deep explainer is used to compute SHAP values, which determine the contribution of each feature in an image to predict the severity of AD. The number of unique prediction

classes corresponds to the number of images that SHAP generates to explain a prediction. In the context of three-class AD severity, SHAP calculates the importance of each pixel's feature and generates three explainable images per class (CN, AD, and MCI). Pixel importance is denoted by color, with red indicating a positive correlation and blue indicating a negative correlation with the predicted value. The intensity of the color indicates the degree of impact a feature has on the prediction. Both red and blue are significant, with opposite effects on the output. The color intensity is critical because it highlights the features that contribute to the model's classification as "feature importance." Therefore, the SHAP plot offers valuable insights into how various features contribute to the neural network's output and identifies the key regions that influence the predictions.

Clinical Relevance Assessment

A set of outcome measures selected by domain experts can be used to evaluate whether identified groups have clinical significance. We selected six AD outcome measures and biomarkers, including Mini-Mental State Examination (MMSE), the Clinical Dementia Rating (CDR), the Functional Activities Questionnaire (FAQ), the Alzheimer Disease Assessment Scale Cognitive score (ADAS-Cog), the Digital Span Score, and the Apolipoprotein E (APOE) genotype. We briefly describe these measures to provide a context for interpretation.

The MMSE is a pen-and-paper test that measures cognitive abilities such as attention, orientation, memory, and visuospatial skills. The maximum score is 30, with 28-30 considered normal, 26-27 indicating MCI, and below 25 suggesting AD. The CDR is a global rating scale that evaluates cognitive, behavioral, and functional aspects of dementia on a scale of 0-3. The FAQ measures instrumental activities of daily living and is rated on a 3-point ordinal scale. The

Alzheimer Disease Assessment Scale includes cognitive and noncognitive sections, with the cognitive section (ADAS-Cog) consisting of standard tests of language, comprehension, memory, orientation, and visual-spatial ability. Higher scores on the ADAS indicate poorer performance. The Digit Span score assesses forward and backward repetition of numbers.

APOE is a gene that provides instructions for making a protein called apolipoprotein E [49]. There are three types of the APOE gene, called alleles: ε2, ε3, and ε4. Everyone has two copies of the gene (one inherited from each parent), and the combination determines your APOE genotype—people can have one of six possible combinations: ε2/ ε2, ε2/ ε3, ε2/ ε4, ε3/ ε3, ε3/ ε4, or ε4/ ε4. Having two copies of the APOE ε4 allele is strongly linked to an increased risk of developing AD. Following that, individuals with the combination of one APOE ε3 allele and one APOE ε4 allele (ε3/ε4) also have an elevated risk compared to those with two copies of ε3. Conversely, individuals with the combination of two APOE ε2 alleles (ε2/ε2) tend to have a lower risk of developing AD. [50].

We examine the cognitive scores of correctly and incorrectly classified outcomes of AD and MCI predictions. The purpose is to identify factors that may have impeded the CNN model's performance on these tasks. A critical aspect is whether the image prediction aligns with other biological data. A subject matter expert evaluated the SHAP plot findings and the neurocognitive outcome measures to assess the clinical significance of the results.

EXPERIMENTAL RESULTS AND ANALYSIS

Experimental Setup

All experiments were implemented using Python and TensorFlow ML API [51] on an NVIDIA GEFORCE GTX 1050 GPU machine. We implemented all experiments using Python and TensorFlow's ML API [42] on a machine equipped with an NVIDIA GEFORCE GTX 1050 GPU. The experiments utilized three models: DenseNet121, DenseNet169, and Inception-ResNet-v2, which were pre-trained on ImageNet and are available on TensorFlow. TensorFlow considers each operation and module in a network as separate layers, including convolutional blocks and auxiliary layers like activation, batch normalization, pooling, and global average pooling. The DenseNet121, DenseNet169, and Inception-ResNet-v2 architectures have 433, 601, and 785 layers, respectively. To fine-tune the models, we unfroze the later blocks, with DenseNet121 at the 313th layer, DenseNet169 at the 369th layer, and Inception-ResNet-v2 at the 616th layer.

To balance the skewed distribution of 1944 images, we applied SMOTE, which generated a balanced dataset with 1024 images for each severity class, totaling 3072 images. We split the dataset into training (80%) and testing (20%) subsets. It is worth noting that the SMOTE-generated images were intentionally constrained to the training data, and the test set consisted only of real images. During training, we automatically applied data augmentation techniques, including horizontal flipping, height and width shifts within ranges of 0.05 and 0.1, rotation within 5 degrees, and zooming within a range of 0.15. This approach aimed to increase data variability and improve the model's robustness. We trained the model using stratified 5-fold cross-validation with a batch size of 32 and a learning rate of 0.00001. Finally, we employed the best-performing fold training model on the test set. We performed multiple experiments by varying the number of epochs (100,250,500,1000) to conduct a comparative analysis of the three CNN models.

Performance Comparison of the DenseNet121 vs DenseNet169 vs Inception-ResNet-v2

Both DenseNet and Inception-ResNet networks aim to improve the flow of information across the network layers and reduce the vanishing gradient. The primary difference lies in the connectivity patterns of the layers within the same block and how information is moved from one layer to the other. The Inception-ResNet uses sparse and residual connections to add the output of one layer to another layer deeper in the network. In contrast, DenseNet uses dense connections to concatenate the output of each layer to the input of all subsequent layers.

Figure 13 displays the training accuracies of three models and compares them to their testing accuracies. DenseNet121 has a relatively high training accuracy of 95.33% but performed poorly with unseen samples, obtaining a testing accuracy of only 77%. Based on the literature, DenseNet169 was expected to perform better than DenseNet121 because it has more dense layers. However, it only slightly outperformed DenseNet121 during training with an accuracy of 95.93%, and even worse with unseen samples, with a testing accuracy of only 75%. The significant difference between the training and testing accuracies for both models indicates that they are overfitting and unable to generalize with unseen samples. This suggests that the dense connectivity pattern of DenseNet may not be suitable for AD severity prediction. Moreover, increasing the network size does not significantly improve the performance and may lead to further overfitting.

Inception-ResNet-v2, compared to the DenseNet models, has a slightly lower training accuracy of 95.12% but a much higher testing accuracy of 94%. The training and testing accuracies for the Inception-ResNet-v2 are comparable, indicating that it is better at generalizing with unseen samples. The network also performed better at epoch 500, indicating that it requires fewer training iterations to produce good results. Inception-ResNet-v2 has a more complex

architecture than DenseNet121 and DenseNet169. A combination of Inception modules and ResNet blocks can help capture more intricate image features. This additional complexity may have helped Inception-ResNet-v2 learn more complex data representations and achieve better performance.

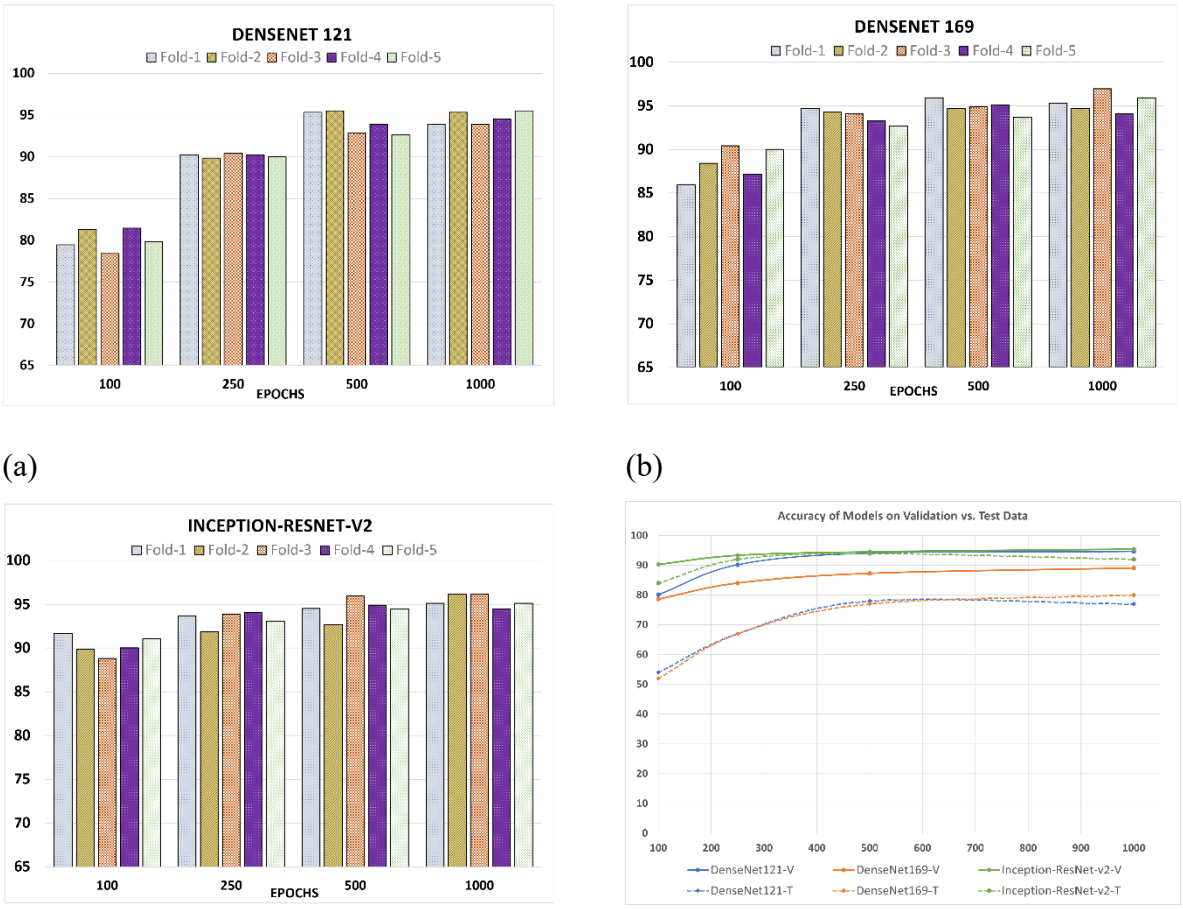


Figure 13. Performance of CNN model on the validation set during training for varying epoch levels across the 5-fold cross-validation (a) DenseNet121 (b) DenseNet169 (c) Inception-Res-Net-v2 (d) accuracy of models on validation vs. test data.

When comparing the sensitivity and specificity results of the three models in Table 2, we can see that each model has different strengths and weaknesses for each diagnostic group. The

first model has high specificity for all three groups, indicating that it is effective at correctly identifying individuals without MCI or AD as negative (true negatives). However, the sensitivity for healthy controls is lower than that for MCI or AD, indicating that the test is less effective at correctly identifying healthy individuals who do not have MCI or AD as negative (true negatives).

Table 2. Performance of the CNN models on the AD severity prediction based on the test dataset.

		AD		MCI		CN	
Epoch	ACC [1]	SEN [2]	SPE [3]	SEN	SPE	SEN	SPE
			DenseNet121				
100	54.15	15.61	99.76	78.54	49.02	82.44	68.29
250	67.48	30.24	100	92.68	62.2	89.02	79.51
500	77.56	54.63	99.76	89.27	77.32	89.27	88.78
1000	76.91	49.27	100	95.61	72.44	92.93	85.85
			DenseNet169				
100	52.36	19.02	99.76	98.05	30.49	98.29	40
250	67.15	35.61	99.51	95.61	56.59	94.63	70.24
500	75.93	58.54	99.51	95.12	68.29	96.1	74.15
1000	75.12	55.12	98.54	97.56	64.88	99.27	72.68
			Inception-ResNet-v2				
100	84.07	78.05	96.83	95.12	80.98	98.29	79.02
250	91.87	89.27	98.05	91.22	93.17	96.59	95.12
500	93.66	92.2	98.78	96.1	92.93	98.78	92.68
1000	92.03	89.76	98.05	96.1	91.22	98.78	90.24

[1] ACC: Accuracy. [2] SEN: Sensitivity. [3] SPE: Specificity.

In contrast, the second model has high specificity for CN and AD but lower specificity for MCI. It also has a high sensitivity for MCI and AD, indicating that it is better at correctly identifying individuals with MCI or AD as positive (true positives). However, the sensitivity for CN is relatively low, indicating that the test may miss some healthy individuals who do not have MCI or AD. The third model has relatively balanced sensitivity and specificity for CN and MCI but lower sensitivity and higher specificity for AD. This suggests that the test may be more effective at ruling out AD in healthy and MCI individuals but may miss some true positive AD cases.

Regarding the AUC values (see Figure 14), Inception-ResNet-v2 has the highest AUC scores across all three classes (AD: 0.95, MCI: 0.95, and AD 0.96). The AUC scores for DenseNet121 (AD: 0.75, MCI: 0.84, CN: 0.89) and DenseNet169 (AD: 0.77, MCI: 0.81, CN: 0.86) are relatively close to each other, with DenseNe121 having slightly better performance.

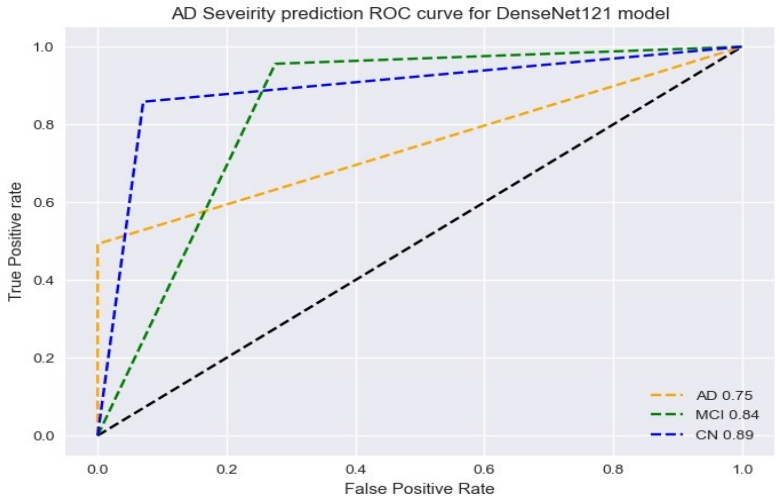
Overall, Inception-ResNet-v2 is the most effective model for classifying new data accurately, as it has the highest test prediction accuracy, sensitivity, and AUC and is least likely to overfit during training.

Comparison of Results with other Existing Models Reported in Literature

In Table 3, a comparison is presented between the proposed framework and state-of-the-art models. The metrics used for comparison are accuracy, sensitivity, precision, and specificity, which are evaluated against models discussed in existing literature. These models have been trained for either a multiclass or binary class problem using the ADNI dataset. Although the accuracy of the models in existing literature is better than that of the proposed model, our model is comparable in terms of sensitivity, specificity, and precision. Furthermore, while existing models

end their evaluation based on model evaluation metrics, our framework goes beyond performance evaluation. In the next section, we provide a feature level interpretation and correlation of model predictions with APOE genotype biomarkers and neurocognitive assessment data.

(a)



(b)

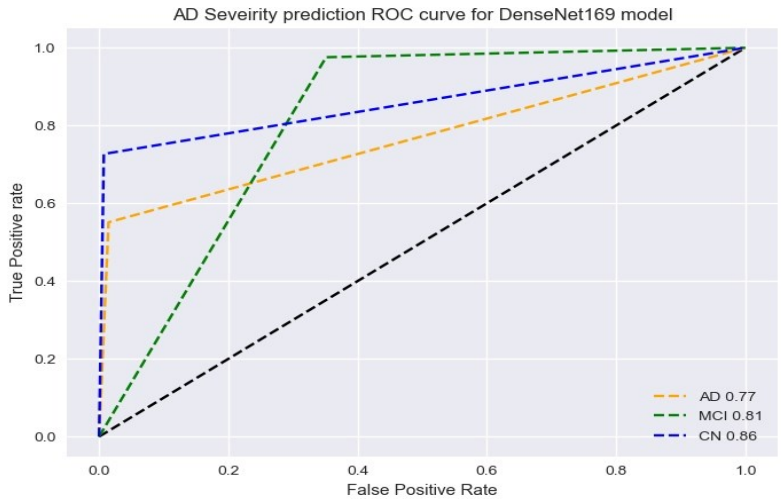


Figure 14. AUC ROC curve for (a) DenseNet121 (b) DenseNet169.

(c)

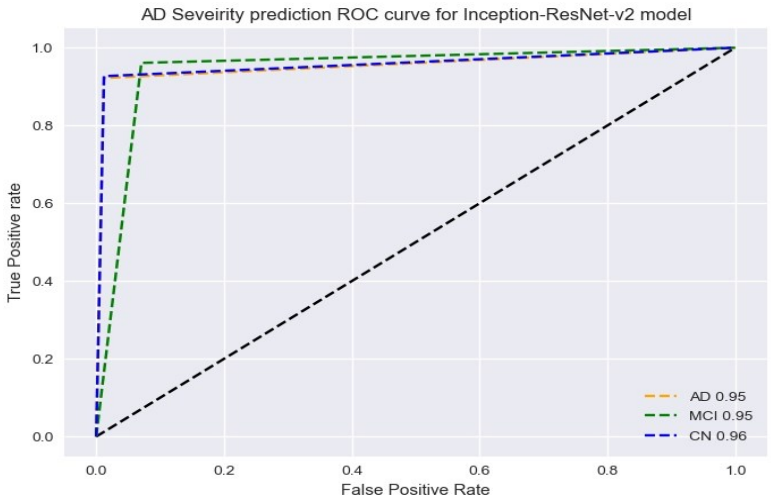


Figure 14 continued. AUC ROC curve for (c) Inception-ResNet-v2.

Table 3. Performance comparison of proposed model with prior models.

Method	Classification Type	Accuracy	Sensitivity	Precision	Specificity
Yousry et al. [27]	Multi classification	97.50	N/A [1]	N/A	N/A
Murugan et al. [28]	Multi classification	95.23	95.00	96.00	N/A
Jain et al. [29]	Multi classification	95.73	96.00	96.33	N/A
Lee et al. [30]	Multi classification	98.06	95.77	98.69	95.19
Basaia et al. [31]	Binary classification	99.20	98.90	N/A	99.50
Helaly et al. [32]	Multi classification	97.00	94.00	96.00	N/A
Proposed	Multi classification	93.66	95.70	95.00	94.80

[1] Not available

MODEL EXPLANATION AND CLINICAL RELEVANCE

We will focus only on explaining the Inception-ResNet-v2 model using SHAP visualization plots, as the sensitivity and specificity analysis of the three models indicate that it performs better than the others. The progression of Alzheimer's disease is associated with changes in the brain, including widening of the interhemispheric fissure, enlargement of cortical sulci and ventricles, and thinning of the corpus callosum [52]. There may also be subtle changes in the white and grey matter of the brain parenchyma, with patients with more severe Alzheimer's disease expected to have larger ventricles and cortical sulci. The SHAP values (red pixels) surrounding the ventricles and cortical sulci suggest that the neural networks may rely on these areas to identify subject images with probable Alzheimer's disease.

A neurology expert reviewed the SHAP plots (Figures 15 and 16). As expected, the SHAP intensity in the plots was highest for the class selected by the model. For example, when the model correctly classified an Alzheimer's disease image, the AD class SHAP plots had the highest values. This also held for correctly classified MCI images. The spatial distribution of SHAP values in the images provided insight into the areas of the brain that the neural network model used the most in making its classification. The domain expert evaluation showed that the model relied on examining the size of the ventricles and cortical sulci to identify probable Alzheimer's disease. Fewer SHAP intensities were observed over the white or grey matter within the brain parenchyma.

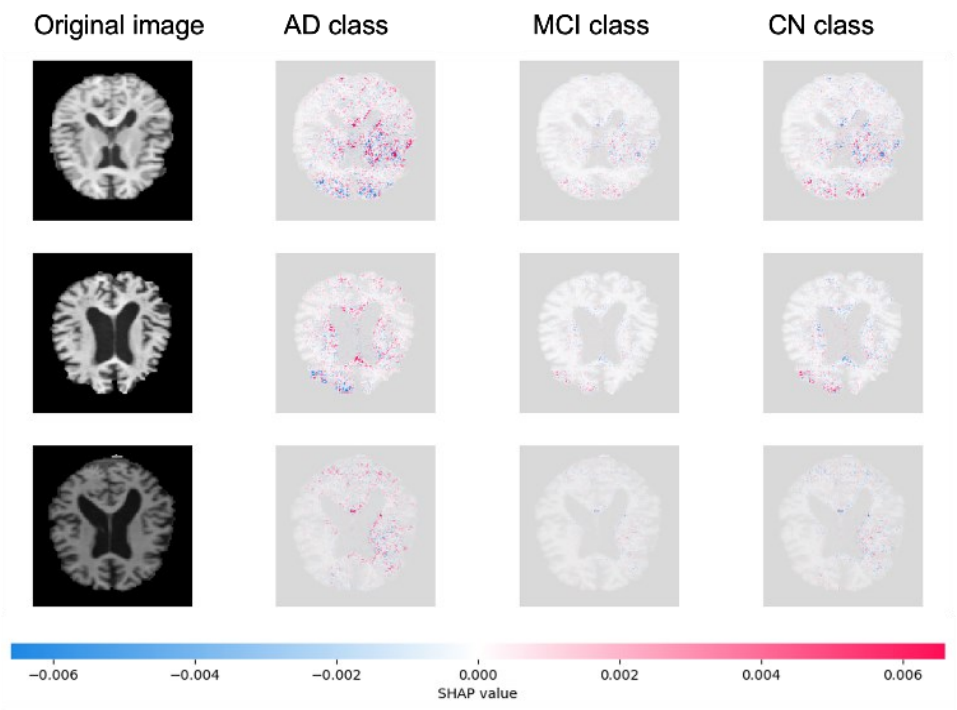
Insights into the misclassified images can be obtained from Tables 4 and 5. Generally, AD cases that were mistakenly classified as MCI or controls showed milder disease characteristics, as indicated by higher MMSE scores and lower ADAS-cog scores. It is worth noting that even the AD case that was incorrectly labeled as a control had a relatively high

MMSE score of 26 and a low CDR of 0.5. The majority of accurately classified AD images were associated with high severity APOE genotype alleles. Interestingly, approximately half of the incorrectly classified AD images exhibited lower severity based on the presence of lower APOE alleles. This suggests that while the model may have misclassified a few correct samples, some of the misclassified samples could have been labeled incorrectly, as indicated by the lower severity of APOE. This trend is observed in both the correctly and incorrectly classified cases.

Table 4. Clinical relevance of Inception-Resnet-V2 model outcomes for correctly predicted AD vs. AD predicted as MCI or CN

	Correct AD	Incorrect MCI	Incorrect CN
N	193	11	1
Gender (Male/Female)	86/107	4/7	1
Age	72.19 ± 5.75	72.05 ± 4.48	79.10
FAQ Total	17.05 ± 7.58	19.2 ± 11.61	0
MMSE	20.94 ± 5.54	22.9 ± 3.14	26.00
Digit span total	22.5 ± 14.72	21.0 ± 14.49	0
CDR Global	0.97 ± 0.49	1.0 ± 0.81	0.50
ADAS-Cog	24.31 ± 11.64	21.42 ± 15.83	0
APOE ε4 (Allele 1) %	30.61	0	0
APOE ε4 (Allele 2) %	69.64	7.14	0

(a)



(b)

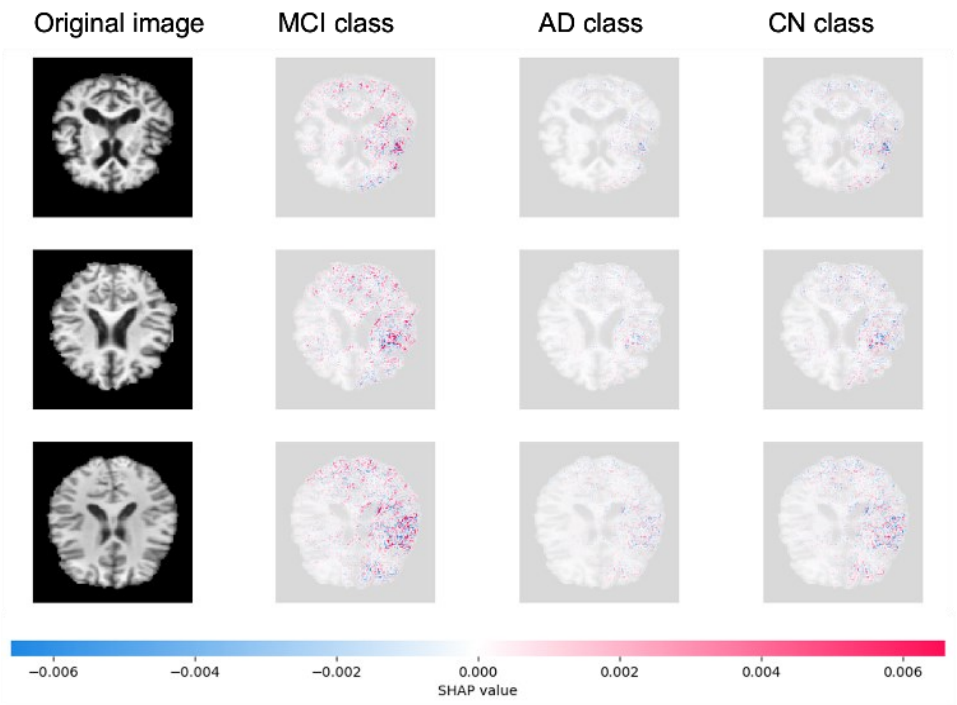
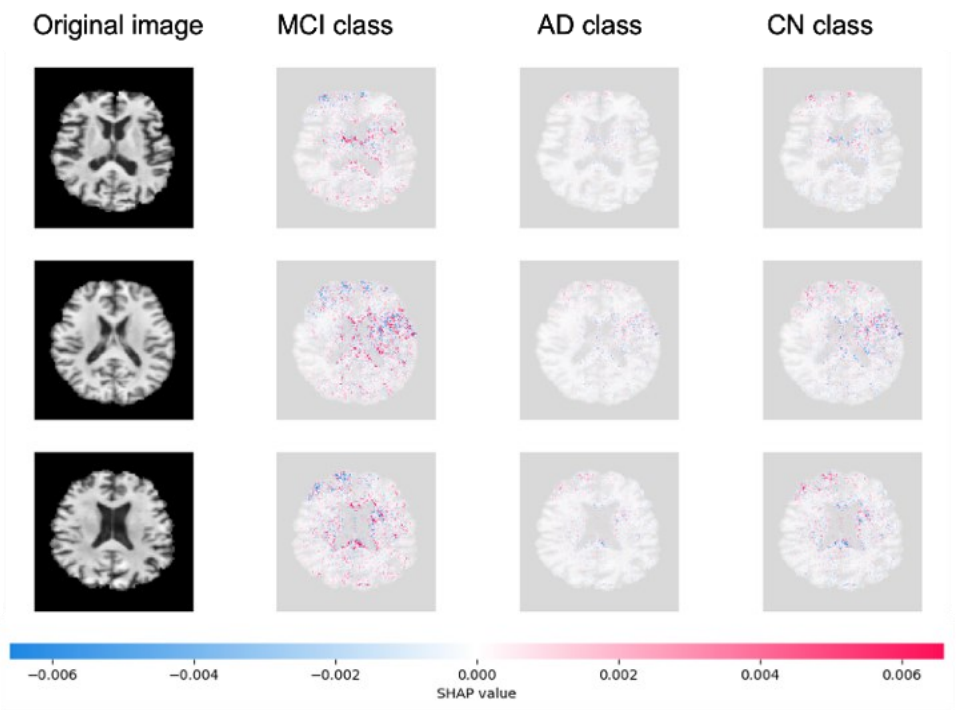


Figure 15. Visualization of prediction for MRIs labeled AD (a) correctly predicted as AD and (b) incorrectly predicted as MCI or CN. Each image is associated with three outputs per image. Red pixels indicate positive SHAP values, increasing the class probability, while blue pixels indicate negative SHAP values, decreasing it.

(a)



(b)

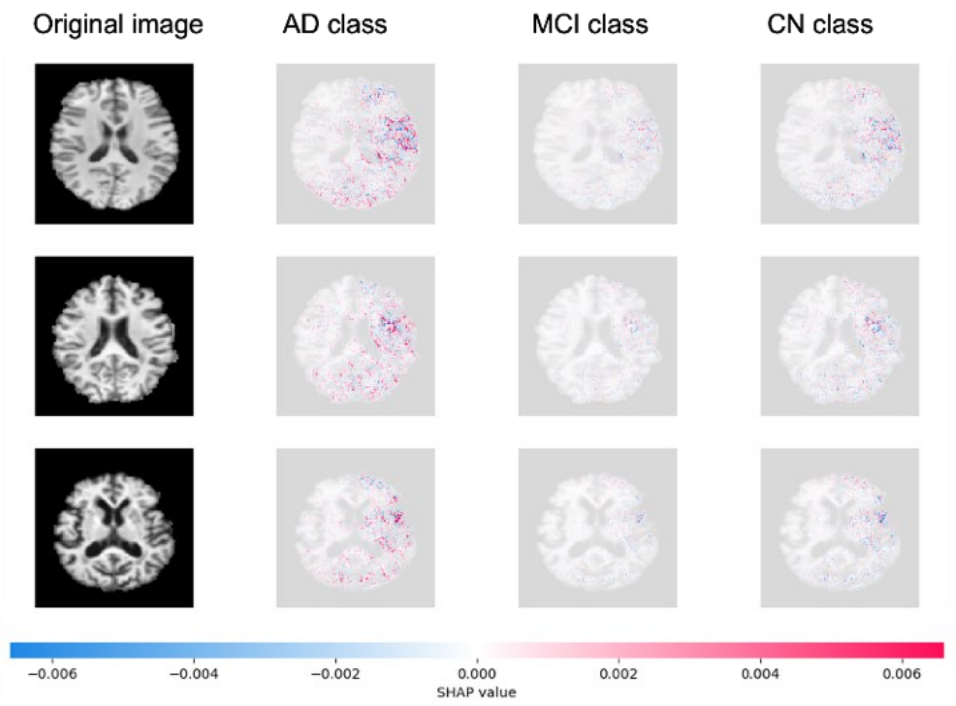


Figure 16. Visualization of prediction for MRIs labeled MCI (a) correctly predicted as MCI and (b) incorrectly predicted as AD or CN. Each image is associated with three outputs per image. Red pixels indicate positive SHAP values, increasing the class probability, while blue pixels indicate negative SHAP values, decreasing it.

Table 5. Clinical relevance of Inception-Resnet-V2 model outcomes for correctly predicted MCI vs. MCI predicted as AD or CN

	Correct MCI	Incorrect AD	Incorrect CN
N	196	5	4
Gender (Male/Female)	105/91	3/2	1/3
Age	71.95 ± 5.39	76.34 ± 2.49	67.15 ± 4.69
FAQ Total	7.21 ± 6.75	2	10
MMSE	25.42 ± 4.11	25.67 ± 1.53	28.33 ± 0.58
Digit span total	36.48 ± 13.93	56	33.00
CDR Global	0.58 ± 0.26	0.5 ± 0	0.5 ± 0
ADAS-Cog	14.55 ± 8.18	15	12.67
APOE ε4 (Allele 1) %	21.62	0	0
APOE ε4 (Allele 2) %	51.22	2.45	4.88

CONCLUSION AND FUTURE WORK

This thesis presents a framework for preprocessing and analyzing MRI for predicting AD severity using three CNN models. Inception-ResNet-v2 sensitivity values were evaluated for CN, MCI, and AD, and it was found that the model has a good ability to identify individuals with MCI or AD as positive. Still, it is less effective at identifying healthy controls who do not have MCI or AD as negative. On the other hand, specificity values for CN, MCI, and AD are relatively high, indicating the model has a good ability to correctly identify healthy individuals who do not have MCI or AD as negative. Still, it is less effective at correctly identifying individuals with MCI as negative, potentially leading to a higher false positive rate.

The high specificity and low sensitivity suggest that the MRI-based diagnostic model is better at ruling out MCI or AD in healthy individuals than detecting the presence of MCI or AD in those with the condition. This may be due to various factors, such as imaging technique choice or criteria for defining MCI or AD. As Alzheimer's disease progresses, the brain's surface becomes thinner, ventricles enlarge, and the corpus callosum thins. Patients with less severe AD are expected to have smaller ventricles and a larger brain surface, and the SHAP explanations show that the predicted CN sample has more red pixels on the surface. In contrast, the predicted AD sample has fewer red pixels, further supporting the model evaluation results.

The study has some limitations, such as the image-slicing step only selecting the middle axial slice, which may not have the best discriminant information for a sample. A larger dataset could result in more accurate predictions and better discrimination between MRI artifacts and brain abnormalities. Nonetheless, deep CNNs show promise for predicting the severity and outcome of AD. Further investigation is needed to improve the Inception-ResNet-v2-based model's ability to identify the AD and CN groups.

REFERENCES

[1] Alzheimer's Association Report, "2022 Alzheimer's disease facts and figures," *Alzheimer's & dementia: the journal of the Alzheimer's Association,* vol. 18, no. 4, pp. 700-789, 2022.

[2] D. Ravi, C. Wong, F. Deligianni, M. Berthelot, J. reu-Perez, B. Lo and G.-Z. Yang, "Deep learning for health informatics," *IEEE journal of biomedical and health informatics.,* vol. 21, no. 1, pp. 4--2, 2016.

[3] M. Torres-Vel'azquez, W.-J. Chen, X. Li and A. B. McMillan, "Application and construction of deep learning networks in medical imaging," *IEEE transactions on radiation and plasma medical sciences,* vol. 5, pp. 137-159, 2020.

[4] D. Yeboah, H. Nguyen, D. B. Hier, G. R. Olbricht and T. Obafemi-Ajayi, "A deep learning model to predict traumatic brain injury severity and outcome from MR images," in *2021 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, 2021.

[5] P. Scheltens, B. De Strooper, M. Kivipelto, H. Holstege, G. Chételat, C. E. Teunissen, J. Cummings and W. M. Van der Flier, "Alzheimer's disease," *The Lancet,* vol. 397, no. 10284, pp. 1577-1590, 2021.

[6] S. Grueso and R. Viejo-Sobera, "Machine learning methods for predicting progression from mild cognitive impairment to Alzheimer's disease dementia: a systematic review," *Alzheimer's research & therapy,* vol. 13, no. 1, p. 162, 2021.

[7] T. J. Saleem, S. R. Zahra, F. Wu, A. Alwakeel, M. Alwakeel, F. Jeribi and M. Hijji, "Deep learning-based diagnosis of Alzheimer's disease," *Journal of Personalized Medicine,* vol. 12, no. 5, p. 815, 2022.

[8] A. Couckuyt, R. Seurinck, A. Emmaneel, K. Quintelier, D. Novak, S. Van Gassen and Y. Saeys, "Challenges in translational machine learning," *Human Genetics,* vol. 141, no. 9, pp. 1451-1466, 2022.

[9] M. Hon and N. M. Khan, "Towards Alzheimer's disease classification through transfer learning.," in *IEEE International conference on bioinformatics and biomedicine (BIBM)*, 2017.

[10] C. Combi, B. Amico, R. Bellazzi, A. Holzinger, J. H. Moore, M. Zitnik and J. H. Holmes, "A manifesto on explainability for artificial intelligence in medicine," *Artificial Intelligence in Medicine,* vol. 133, p. 102423, 2022.

[11] L. Torrey and J. Shavlik, "Transfer Learning," in *Handbook of Research on Machine Learning Applications and Trends*, IGI Global, 2010, pp. 242-264.

[12] L. S. Lundberg SM, "A unified approach to interpreting model predictions," *Advances in neural information processing systems,* vol. 30, 2017.

[13] F. Chollet, Deep learning with Python, Simon and Schuster, 2021.

[14] C. C. Aggarwal and others, Neural Networks and Deep Learning, Springer Cham, 2018.

[15] I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, MIT Press, 2016.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative adversarial nets Advances in neural information processing systems," *arXiv preprint arXiv:1406.2661,* 2014.

[17] M. R. Hasan, "Deep Learning," School of Computing, University of Nebraska-Lincoln, 2021. [Online]. Available: https://engineering.unl.edu/hasan/deep-learning/.

[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, Karpathy, rej, A. Khosla, M. Bernstein and others, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV),* vol. 115, no. 3, pp. 211-252, 2015.

[19] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556,* 2014.

[20] K. a. Z. X. He, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

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

[22] C. Szegedy, S. Ioffe, V. Vanhoucke and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the AAAI conference on artificial intelligence*, 2017.

[23] G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[24] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks.," *Communications of the ACM,* vol. 60, no. 6, pp. 84-90, 2017.

[25] Q. Yang, Y. Zhang, W. Dai and S. J. Pan, "Introduction," in *Transfer Learning*, Cambridge University Press, 2020, p. 3–22.

[26] A. Zhang, Z. C. Lipton, M. Li and A. J. Smola, "Dive into Deep Learning," *arXiv preprint arXiv:2106.11342,* 2021.

[27] Y. AbdulAzeem, W. M. Bahgat and M. Badawy, "A CNN based framework for classification of Alzheimer's disease," *Neural Computing and Applications,* vol. 33, pp. 10415 -10428, 2021.

[28] S. Murugan, C. Venkatesan, M. Sumithra, X.-Z. Gao, B. Elakkiya, M. Akila and S. Manoharan, "DEMNET: a deep learning model for early diagnosis of Alzheimer diseases and dementia from MR images," *IEEE Access,* vol. 9, pp. 90319-90329, 2021.

[29] R. Jain, N. Jain, A. Aggarwal and D. J. Hemanth, "Convolutional neural network based Alzheimer's disease classification from magnetic resonance brain images," *Cognitive Systems Research,* vol. 57, pp. 147-159.

[30] B. Lee, W. Ellahi and J. Y. Choi, "Using deep CNN with data permutation scheme for classification of Alzheimer's disease in structural magnetic resonance imaging (sMRI)," *IEICE TRANSACTIONS on Information and Systems,* vol. 102, no. 7, pp. 1384-1395, 2019.

[31] S. Basaia, F. Agosta, L. Wagner, E. Canu, G. Magnani, R. Santangelo, M. Filippi, A. D. N. Initiative and others, "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.

[32] B. M. H. A. Helaly HA, "Deep learning approach for early detection of Alzheimer's disease," *Cognitive computation,* pp. 1-17, 2021.

[33] S. Kokkalla, J. Kakarla, I. B. Venkateswarlu and M. Singh, "Three-class brain tumor classification using deep dense inception residual network," *Soft Computing,* vol. 25, pp. 8721-8729, 2021.

[34] Z. Sadeghi, R. Alizadehsani, M. A. Cifci, S. Kausar, R. Rehman, P. Mahanta, P. K. Bora, A. Almasri, R. S. Alkhawaldeh, S. Hussain and others, "A Brief Review of Explainable Artificial Intelligence in Healthcare," *arXiv preprint arXiv:2304.01543,* 2023.

[35] L. Kohoutov'a, J. Heo, S. Cha, S. Lee, T. Moon, T. D. Wager and C.-W. Woo, "Toward a unified framework for interpreting machine-learning models in neuroimaging," *Nature protocols,* vol. 15, no. 4, pp. 1399-1435, 2020.

[36] X. Bai, X. Wang, X. Liu, Q. Liu, J. Song, N. Sebe and B. Kim, "Explainable deep learning for efficient and robust pattern recognition: A survey of recent developments}," *Pattern Recognition,* vol. 120, p. 108102, 2021.

[37] J. B. Bae, S. Lee, W. Jung, S. Park, W. Kim, H. Oh, J. W. Han, G. E. Kim, J. S. Kim, J. H. Kim and others, "Identification of Alzheimer's disease using a convolutional neural network model based on T1-weighted magnetic resonance imaging," *Scientific reports,* vol. 10, no. 1, p. 22252, 2020.

[38] E. E. Bron, S. Klein, J. M. Papma, L. C. Jiskoot, V. Venkatraghavan, J. Linders, P. Aalten, P. P. De Deyn, G. J. Biessels, J. A. Claassen and others, "Cross-cohort generalizability of deep and conventional machine learning for MRI-based diagnosis and prediction of Alzheimer's disease," *NeuroImage: Clinical,* vol. 31, p. 102712, 2021.

[39] S. a. S. M. Chakraborty, J. Park and S. Aich, "Early Detection of Alzheimer's Disease from 1.5 T MRI Scans Using 3D Convolutional Neural Network," in *Proceedings of International Conference on Smart Computing and Cyber Security: Strategic Foresight, Security Challenges and Innovation (SMARTCYBER 2020)*, 2021.

[40] K. M. Sudar, P. Nagaraj, S. Nithisaa, R. Aishwarya, M. Aakash and S. I. Lakshmi, "Alzheimer's Disease Analysis using Explainable Artificial Intelligence (XAI)," in *2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, 2022.

[41] S. Shojaei, M. S. Abadeh and Z. Momeni, "An evolutionary explainable deep learning approach for Alzheimer's MRI classification," *Expert Systems with Applications,* vol. 220, p. 119709, 2023.

[42] M. W. Weiner, P. S. Aisen, C. R. Jack Jr, W. J. Jagust, J. Q. Trojanowski, L. Shaw, Saykin, r. J, J. C. Morris, N. Cairns, L. A. Beckett and others, "The Alzheimer's disease neuroimaging initiative: progress report and future plans," *Alzheimer's & Dementia,* vol. 6, no. 3, pp. 202-211, 2010.

[43] P. Kalavathi and V. S. Prasath, "Methods on skull stripping of MRI head scan images—a review," *Journal of digital imaging,* vol. 29, no. 3, pp. 365-379, 2016.

[44] M. Jenkinson, C. F. Beckmann, T. E. Behrens, M. W. Woolrich and S. M. Smith, "Fsl," *Neuroimage,* vol. 62, no. 2, pp. 782-790, 2012.

[45] M. Jenkinson, P. Bannister, M. Brady and S. Smith, "Improved optimization for the robust and accurate linear registration and motion correction of brain images," *Neuroimage,* vol. 17, no. 2, pp. 825-841, 2002.

[46] M. Brett, C. J. Markiewicz, M. Hanke, M.-A. Côté, B. Cipollini, P. McCarthy, D. Jarecka, C. P. Cheng, Y. O. Halchenko, M. Cottaar, E. Larson, S. Ghosh, D. Wassermann and S. Gerhard, *nipy/nibabel: 4.0.0,* Zenodo, 2022.

[47] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of artificial intelligence research,* vol. 16, pp. 321-357, 2002.

[48] Y. Liang, S. Li, C. Yan, M. Li and C. Jiang, "Explaining the black-box model: A survey of local interpretation methods for deep neural networks," *Neurocomputing,* vol. 419, pp. 168-182, 2021.

[49] A. L. Lumsden, A. Mulugeta, A. Zhou and E. Hypp"onen, "Apolipoprotein E (APOE) genotype-associated disease risks: a phenome-wide, registry-based, case-control study utilising the UK Biobank," *EBioMedicine,* vol. 59, p. 102954, 2020.

[50] S. Bookheimer and A. Burggren, "APOE-4 genotype and neurophysiological vulnerability to Alzheimer's and cognitive aging," *Annual review of clinical psychology,* vol. 5, pp. 343-362, 2009.

[51] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, Davis, y, J. Dean, M. Devin and others, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *{arXiv preprint arXiv:1603.04467,* vol. 2016.

[52] K. A. Johnson, N. C. Fox, R. A. Sperling and W. E. Klunk, "Brain imaging in Alzheimer disease," *Cold Spring Harbor perspectives in medicine,* vol. 2, no. 4, p. a006213, 2012.