Universität Rostock

Traditio et Innovatio

Master Thesis

Detection of Frontotemporal Dementia by Learning Few Training Samples

Submitted by:

Zain Ul Haq

Marticulation Number: 217205313

M.Sc in Electrical Engineering

(Specialization in Information Technology)

Submitted on:

23.05.2022

Supervisors:

Dr. rer. hum. Martin Dyrba

Prof. Dr.-Ing. Thomas Kriste

Universität Rostock Traditio et Innovatio

Statutory Declaration

I hereby declare that I have written this thesis independently, have not submitted it elsewhere for examination purposes and has not been published. The thesis have not used any sources or aids other than those indicated.The thesis was not yet, even in part, used in any of examination or as a course performance.

Rostock, 23.05.2022

Zain Ul Haq

Acknowledgments

I would like to express my gratitude to Dr. rer. hum. Martin Dyrba and Vadym Gryshchuk for their immense support and invaluable guidance as my supervisor throughout the thesis. I am very honored to have the opportunity to work in the area of deep learning for the medical domain. Martin and Vadym's ideas, vision, and enthusiasm have always been motivating for me to overcome the problems that occurred during the thesis work. Without their indispensable support, this thesis would not have been completed successfully.

I would also like to thank Prof.Dr.Ing.Thomas Kirste for allowing me to work on this thesis topic. His invaluable and continuous support to the whole group of MMIS for having a perfect research environment.

A special thanks to all my professors throughout my Master's study at the University of Rostock, who contributed to the development of my academic knowledge and provides me an opportunity to excel in the skills that I possess today.

I special thanks to the Alzheimer's Disease Neuroimaging Initiative (ADNI) and The Frontotemporal Labor Degeneration Neuroimaging Initiative (NIFTD) for providing the data to conduct this study.

Finally, I am forever thankful to my mother, Prof.Bushra Ansar for his blessing and prayers. Without her immense support, love, and motivation, it would not have been possible what I have achieved so far.

Universität Rostock Traditio et Innovatio

Abstract

Faculty of Computer Science and Electrical Engineering

Universitat Rostock

Master of Science in Electrical Engineering

(Specialization in Information Technology)

Detection of Frontotemporal Dementia by Learning Few Training Samples

by Zain Ul Haq

Background: Convolutional neural networks (CNNs) achieve the high classification accuracy for detecting frontotemporal dementia with a large number of training samples based on magnetic resonance imaging (MRI) scans, but they didn‘t achieve good diagnostic accuracy with few training samples. One important reason is that in the medical domain, the acquisition is quite hard and complicated due to patients‘ privacy concerns. Recently developed a few-shot learning methodology that deals with the data insufficiency problem. Few-shot learning methodology proposes the strategies through which we resolve the problem of data insufficiency and achieve the classification performance as same as with a large number of training samples. We investigate the detection of frontotemporal dementia using only a few MRI scans for training.

Methods: We utilized the transfer learning and few-shot learning methodologies to overcome the problem of a few available training samples. Firstly, we created the feature extraction model that is trained on the large ADNI dataset (a total of 662 samples). This developed model is the convolutional neural network that learns feature representations based on ADNI MRI scans. Furthermore, we transfer the representations learned by the feature extraction model to the model that is trained on the small FTD dataset (a total of 279 data samples) by following a model perspective-based embedding learning methodology of few-shot learning.

Results: We developed the CNN models utilizing the transfer learning methods that learn the optimal feature representations. The CNN model with the fine-tuning method based on the ADNI dataset achieves the Alzheimer's disease classification accuracy of 0.97. Secondly, we achieved the classification accuracy of FTD disease with only 20 training samples of 0.63. As we increase the training samples up to 40 we achieved the FTD diagnostic accuracy of 0.75.

Keywords: Convolutional Neural Networks, Few-Shot Learning, Magnetic Resonance Imaging, Transfer Learning, Frontotemporal Dementia, Alzheimer’s Disease.

TABLE OF CONTENTS

List of Abbrevations

- **AI:** Artificial Intelligence
- **ML:** Machine Learning
- **DL:** Deep Learning
- **AD:** Alzheimer's Disease
- **FTD:** Frontotemporal Dementia
- **ConvNet or CNN:** Convolutional Neural Network
- **ANN:** Artificial Neural Network
- **ADNI:** Alzheimer's Disease Neuroimaging Initiative
- **NIFTD:** The Frontotemporal Labor Degeneration Neuroimaging Initiative
- **CN:** Control Normal
- **LMCI:** Late Mild Cognitive Impairment
- **SGD:** Stocastic Gradient Descent
- **FSL:** Few-shot learning

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Applicability of Artificial Intelligence in Medical Imaging

In the early 2000's, the attractiveness of the Deep Neural Network can be subjected to increased computational power and a large number of datasets available. Due to the provision of high processing power and the large availability of large datasets, these large computations can be done through an advanced GPU (Graphical Processing Units), which is supposed to be a significant milestone achieved in the area of computer vision [31][32]. Afterwards, the modified versions of Neural Network architecture and the optimal model performances become more and more popular. Artificial Intelligence (AI) methods have benefited the learning-based methodologies in many research areas. AI methods are data-driven, which can identify the patterns in the data very efficiently with large training data so they can beat the domain experts which enables businesses and industries to have more insights into the data. There are numerous applications of Artificial Intelligence (AI) and Machine Learning (ML) that include speech recognition, Natural Language Processing (NLP), Robotics, and Medical Imaging. Image Classification has become the key task in many computer vision applications. In the Medical Imaging domain, diagnoses of many diseases can be done via 2D/3D Medical Resonance Imaging (MRI), X-Ray, and CT scans to extract the detailed features from the 3D images.

Especially in the healthcare area, there are many applications, in which AI is playing a vital role. For instance, an algorithm that can detect atrial fibrillation in an Electrocardiogram (ECG) achieve a specificity of 83.4% and sensitivity of 82.3% [33]. Another interesting application in the automatic classification of lung nodules is computed tomography (CT) scans for example in lung cancer screening. The research in [34] provides an algorithm that can detect lung cancer with an error rate of only 4.59%. Recently, an automatic lung cancer classification model with an improved model performance by analyzing patients' current and previous CT scans achieved a state-of-the-art performance of 94.4% AUC [35]. With the increasing popularity of AI, over the last two decades, the demand for medical imaging is also increased. It experienced in USA and Canada the annual increase trend of medical examination, with an annual growth of 11.6% from 2000 to 2006 and 3.7% from 2013 to 2016 [36].

AI methods are benefitting the analysis of MRI head scans. For instance, Don et al. proposed an algorithm for the detection and segmentation of brain tumors, which have an accuracy of 86% in the Multimodal Brain Tumor segmentation challenge (BRAT) [37]. In [38] proposed an algorithm for diagnosing Alzheimer's disease (AD) by analyzing MRI head scans with an accuracy of up to 90%. Several researches have been proposed for detecting AD by analyzing the MRI head scans [39] [40]. Other diseases can also be classified by exploiting the machine learning methods such as Frontotemporal dementia (FTD) and several algorithms have also been proposed for the automatic classification of frontotemporal dementia [41][42][43].

1.2 Importance of CNNs in Computer Vision

Optimal medical image classification plays a vital part in clinical care and treatment. For instance, X-ray analysis is the finest tool to diagnose pneumonia, but identifying pneumonia from a chest X-ray requires an expert radiologist which is a rare and expensive resource in some areas. Long ago, traditional machine learning methods such as support vector machines (SVM) have been used for classifying medical images. These standard machine learning methods have the following disadvantages: the performance of these methods is sub-optimal, and the selection of features is time-consuming according to different objects [55]. The deep neural networks (DNN), especially the convolutional neural networks (CNN) are commonly used in a wide range of medical image classification tasks and achieved a great performance after the real breakthrough of CNNs in 2012, the introduction of the deep neural network known as "AlexNet" [31]. Other research on medical image classification by CNN has achieved a performance beating human experts. The research presented by kermany et.al [54] proposes a transfer learning method to classify 108,309 Optical coherence tomography (OCT), and the average performance of 6 human experts is equal to the weighted average error.

The problem with deep neural networks is that it requires a large amount of training data to train CNNs due to the increasing depth of the CNNs. The process of medical image collection is hard, as the collecting and annotating medical image data have limitations of data privacy concerns and the requirement for time-consuming expert explanations.

1.3 Challenge of few training medical image samples

As stated earlier in the previous sections, there has been an increasing trend in the usage of medical imaging. Thus, every year we are available with more and more medical scans, which could be used as training data for CNNs. As we know medical data collection is hard, most of the data is confronted with patient's privacy concerns and no open source is available for storing these medical scans. In other areas of computer vision, there are an enormous number of public datasets available such as kinetics 400-datasets. This dataset is used for video classification tasks and to study their effects on action recognition [56]. Medical image datasets are generally smaller and few publicly medical image databases are available for usage. For instance, Alzheimer Disease Imaging Initiative (ADNI) consists of a total of 1500 MRI head Scans [57]. The challenge of using a large number of medical scans is the generalizability of data to different tasks. There is a significant difference in contrast in different MRI sequences, although they have the same pathologies. Hence, it is difficult to combine different MRI scans to develop a larger benchmark dataset [58].

Compared to other domains in computer vision, where we have a large amount of data available to train CNNs. In the medical area, we have a limited amount of data to train CNNs. However, we have numerous solutions to deal with this problem such as data augmentation techniques, transfer learning, and the new emerging methodology in machine learning known as Few-shot learning. We address the problem of few training samples in our work by few-shot learning methodologies, specifically embedding learning (discussed in 4.3 Embedding Learning).

1.4 Few-Shot Learning

Deep learning applications have been characterized as data-intensive, but it is often hindered by the lack of training samples, specifically in the medical domain, where the data acquisition with supervised information is very hard due to privacy, safety, or ethical issues. Recently, the emerging machine learning method known as Few-shot Learning (FSL) can tackle this problem. FSL proposed the idea that using prior knowledge, FSL algorithms can quickly generalize to new tasks consisting of fewer training samples with supervised information. FSL methods deal with three perspectives: (a) data,

which uses prior knowledge to augment the data size; (b) model, which uses the prior knowledge to reduce the hypothesis size; (c) algorithm, which uses prior knowledge to find the best hypothesis in the given hypothesis space. The formal FSL definition is defined as [69]:

"FSL is a type of machine learning problem (denoted by experience E, classification task T, and the performance measure P) Where E contains only a limited number of supervised information for the target T."

In our thesis, we considered the model perspective of FSL. In model perspective methods we use the prior knowledge to constrain model hypothesis space. Which results in a much smaller hypothesis space. Due to the reduced hypothesis space, we only have to optimize the small hypothesis space and as a result, new tasks can easily generalize with small training samples. Model-based perspective is further divided into different strategies [69].

Strategy	Prior Knowledge	How to reduce hypothesis space
Multi-task learning	Other tasks with their datasets	Share parameters
Embedding learning	Embedding learned from other tasks	maps samples to a reduced embedding space in which similar and dissimilar samples can easily differentiate
Learning with external memory	Embedding learned from another task to interact with memory	Refine samples using key-value pairs stored in memory
Generative model	The prior model learned from another task	Constrain the form of distribution

TABLE :1. 1 FEW-SHOT LEARNING MODEL PERSPECTIVE BASED STRATEGIES [69]

1.5 Challenge of Computational Power for 3D CNNs

The challenge in training a deep neural network is the requirement of high computational power. Compared to 2D images, CT and MRI scans consisted of multiple slices, which results in 3D volume [59]. To highlight the difference between 2D and 3D images, the images used by [6] have the size of 224 x 224 which consisted of 50,176 pixels. The MRI head scan from ADNI that have a dimension of 192 x 192 x 160 consisted of 5 898 240 voxels. Hence, we require more computational power to process these 3D MRI scans. For instance, the "3D-ResNext" which is trained on a kinetics dataset requires the computational power of 8 Tesla P100 NVIDIA GPUs [61]. In the training of our models in this work, we used Google Colab GPUs.

Transfer learning uses pre-trained CNN networks to reduce the amount of training data needed and additionally it also benefited from the less computational power required for training the CNN. There are publicly available 2D pre-trained networks, which are trained on the ImageNet datasets such as Inception-V3, Inception-V4, Resnet, or the VGG16 network. However, there are very few 3D pre-trained networks available in Pytorch due to the lack of training data and the requirement of high computational power. We used the pre-trained Resnext-18 architecture for the feature extraction of 3D MRI head scans, which is briefly explained in chapter 5.

In this thesis, we utilized the transfer learning and few-shot learning methodologies to overcome the problem of fewer training samples. Firstly, we create the feature extraction model that is trained on the large ADNI dataset which is used as prior knowledge for the downstream task of FTD disease. This model is a convolutional neural network that will utilize transfer learning methods. The goal of

this CNN model is to learn optimal feature representations that help in the downstream task to classify FTD disease. Secondly, we transfer the representations learned by the feature extraction model to the model that is trained with the fewer training samples of the FTD dataset by following a model perspective-based strategy known as Embedding learning of few-shot learning.

1.6 Research Objective

Convolutional neural networks are traditionally state-of-the-art tools for extracting features in image data. Outstanding results have been achieved by the CNNs with 2D/3D image data samples such as classification, medical image classification, segmentation and pattern recognition, etc. A significant breakthrough has been achieved by the introduction of the deep neural network in 2012 known as AlexNet [31], followed by a modified architecture of CNNs like VGG [30], GoogleNet [28], and Resnet [29], etc. These significant performance achievements are due to optimizing the network architectures so that the model can learn more detailed features, and the availability of a large number of training datasets such as ImageNet, MNIST, COCO, CIFAR, etc. Compared to 2D image data, there are very few researches available for the 3D image data, as there are no standard methods developed so far, due to the lack of a large amount of annotated datasets and the requirement of high computational power. Most popular publicly available 3D datasets in the medical domain known as ADNI [57] have been used in many kinds of research such as ADDTLA [61], GWAS [62], and Voxel-based visualization for FTD disease [63]. In each mentioned research for the 3D image data, the 3D images have been pre-processed with different techniques.

CNN's have the power to extract meaningful features from the raw or pre-processed 3D MRI head scans such as the hippocampus, brain shape, cortical thickness, atrophy of temporal lobes, etc. CNN networks have proved that it can extract meaningful features from 3D MRI scans that aid us in the early diagnosing of dementia and the progression of dementia. Researches in [61], [62], [63], [64], presents promising results in diagnosing the types of dementia by analyzing the 3D MRI head scans. Although the high accuracies have been achieved by CNN networks for 3D MRI data, none of the papers have reported promising results with fewer training samples of 3D MRI data. Currently, there has been great demand that the models can perform with few training samples and achieve comparable performance as with a large number of training samples.

Therefore, this thesis has two goals,

- Develop the feature extraction model using the Alzheimer's disease data from ADNI and evaluate the model using a gold-standard cross-validation strategy and on different evaluation metrics.
- Use the feature extraction model based on Alzheimer's disease data as prior knowledge for classifying the Frontotemporal dementia (FTD) in a few-shot learning manner.

1.7 Thesis Structure

In the remaining part of this thesis documentation, classifying FTD disease with few training samples is explained with different methodologies. So far, in this chapter, the motivation, challenges, and research objectives are addressed. The rest of the chapters of this thesis are organized as follows:

Chapter 2 (Related Work) reviews the related literature on medical image classification for Alzheimer's disease and frontotemporal dementia.

Chapter 3 (Theoretical Basis) illustrates dementia and its subtypes. Moreover, the concepts that are essential for understanding this work. Introduce with CNN networks and discussed its architecture in detail. Afterwards, the training process and learning approach were discussed. Finally, the chapter illustrates the concept of transfer learning with their available methods and the regression algorithms.

Chapter 4 (Methods) illustrates the methods that have been used in this work to classify Alzheimer's and frontotemporal dementia diseases.

Chapter 5 (Implementation) illustrates all the model strategies that have been developed to classify the downstream task of frontotemporal dementia.

Chapter 6 (Results and Discussion) reviews the results obtained from our developed models and discuss each obtained result in detail.

Chapter 7 (Conclusion) summarizes all this work in a structured way.

Chapter 2

Related Work

However, CNNs have achieved a good classification with a large amount of training samples, one of the major challenges in training deep convolution networks is less amount of training samples. As discussed earlier, in medical imagining the major problem is the acquisition of a large amount of training samples such as MRI scans or CT scans, etc. due to patient's privacy concerns or clinical policies. Usually, the classification of Alzheimer's disease or sub-types of dementia is confronted with the limitation of MRI head scans which is a vital source in training deep CNNs. Another challenge during the classification of medical images is the limited computational power due to the processing of 3D MRI scans. Researches overcome these shortcomings by employing the transfer learning method. Yi et.al [62] used a transfer learning method to train a 3D convolutional neural network (CNN) based on MRI head scans from the screening stage in the ADNI consortium to extract image features that reflect Alzheimer's disease progression. Marica et.al [65] effectively classified Alzheimer's disease by using the power of transfer learning and also optimized the architecture of the deep network. They initialize the architecture with the pre-trained weights from large benchmark datasets consisting of natural images and achieve a state-of-the-art performance than existing deep learning methods. Another research proposed by Taher M.Ghazal et.al [61] classified the different stages of AD (mild demented: MD, very mild demented: VMD, Non- demented: ND, moderated demented: MOD) by employing transfer learning methods and achieves 91.70% accuracy for this multi-classification task. The whole classification process is depicted below graphically which is almost similar to other deep learning-based methods for the classification of AD:

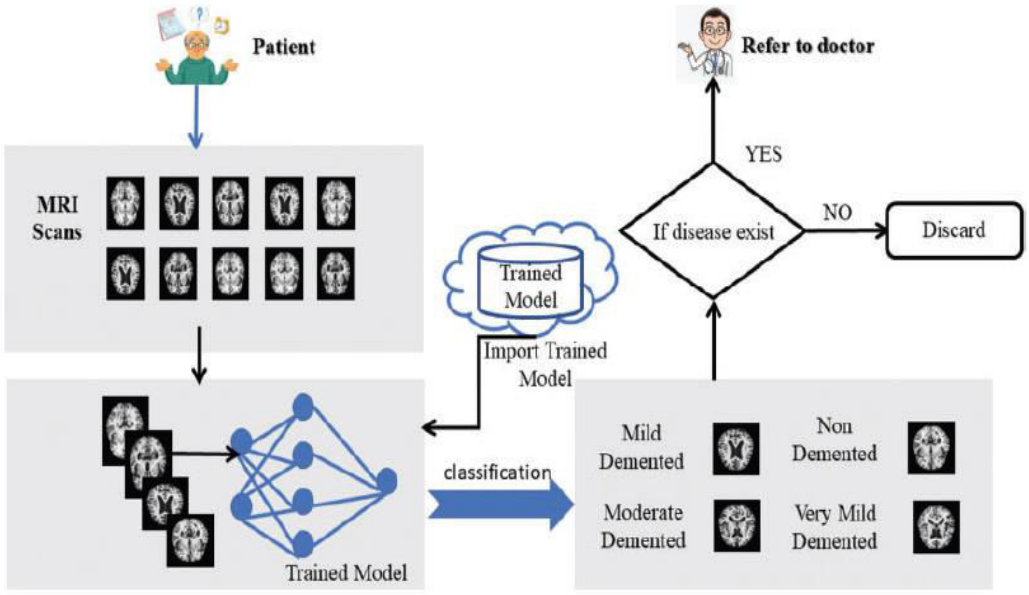

FIGURE 2. 1 PROCESS OF DIAGNOSING AD AND ITS SUB-TYPES [45]

Hekung-II et.al [49] propose other techniques to advance the model performance and overcome the limitation of training samples by using different modalities that give complementary information and improves the model's accuracy. Another research by [48] proposes using three imaging modalities of biomarkers such as MRI, FDG-PET, and CSF for the classification of AD/MCI. The research achieves good performance with the combined modalities method as compared to a single modality-based

classification of AD/MCI. Recent research for detecting the FTD by using MRI biomarkers achieves an accuracy of 83% for the differentiation of FTD with other diagnosis groups of dementia [51].

Although researchers have employed different methodologies to overcome the problem of limited computational power and less training data. However, we also exploit the power of transfer learning to boost our model performance, but the research dedicated to classifying the FTD was with not more than even 10 training samples which are very few. We investigated this problem in our work and used the emerging methodology in machine learning known as Few-shot learning. Recent research [66] has been done to study the activation maps of the brain using the Few-shot learning approach. They developed a neuroimaging benchmark dataset and compared multiple learning methods, including meta-learning and as well as backbone networks. They concluded with their study that the few-shot method can decode the brain signals efficiently. Another research for the task of medical image segmentation by using few-shot learning proposed a new network for a few-shot image classification known as a prototypical network [67].

The baseline research [68], is also considered in our work to classify FTD with very few training samples. The baseline research is based on the embedding learning methodologies. Meta-learning algorithms focus on learning algorithms that can rapidly adapt to new test time tasks with very few data samples, which is considered as the backbone concept for few-shot classification tasks. In the baseline research, they proposed that good learned representation are more powerful for few-shot classification task as compared to the existing meta-learning algorithms. Meta-baseline [69] research which is also a motivation behind this baseline research provides a simple meta-learning process for a few-shot classification task. The baseline research is evaluated on the ImageNet dataset, which motivates us with this embedding learning concept and implemented on FTD classification task.

Chapter 3

Background

This chapter covers a brief overview of Alzheimer's and frontotemporal dementia diseases and the overall concepts that are very essential for understanding this thesis. It introduces the formal definition of dementia disease and discusses the sub-types of dementia. It mainly highlights the factors of both diseases, brain regions affected by these dementias, and the possible remedies for these diseases. Furthermore, the popular network in the area of Deep Learning known as Convolutional Neural Networks (CNNs) is the state of the art technique for image-based classification tasks. CNNs are a specialized architecture of the Artificial Neural Network. Before understanding the CNNs, first, the theory of Artificial Neural Networks needs to understand.

3.1 Dementia

Dementia is defined as a brain disorder that is characterized by a chronic decline in brain functionality due to loss or damage to neurons in the brain. The number of factors involved in developing dementia is advanced age, genetic disorder, traumatic brain injuries, and environmental factors [52]. The disease dementia is a widespread term of brain disorder that results in several symptoms that disturbs the normal brain functions such as thinking, intellectual abilities, memory recollection, and the use of language, and also affects the patient's daily life activities [52].

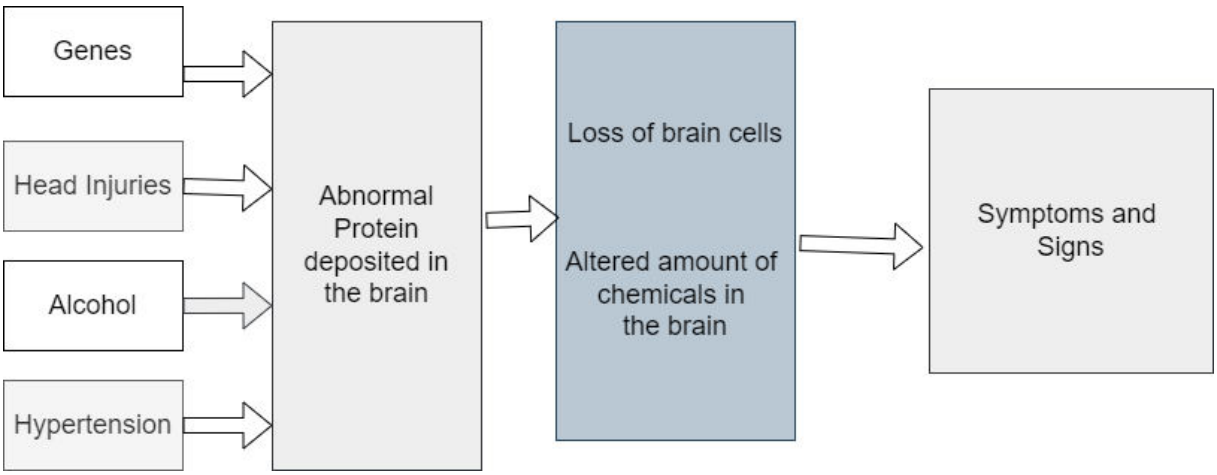

FIGURE :3. 1 DEPICTS THE SCHEMATIC VIEW OF DEMENTIA [52]

This work proposes the computer-aided diagnosis (CAD) system for the early detection of Alzheimer's disease and the Frontotemporal dementia (FTD) using MRI biomarkers.

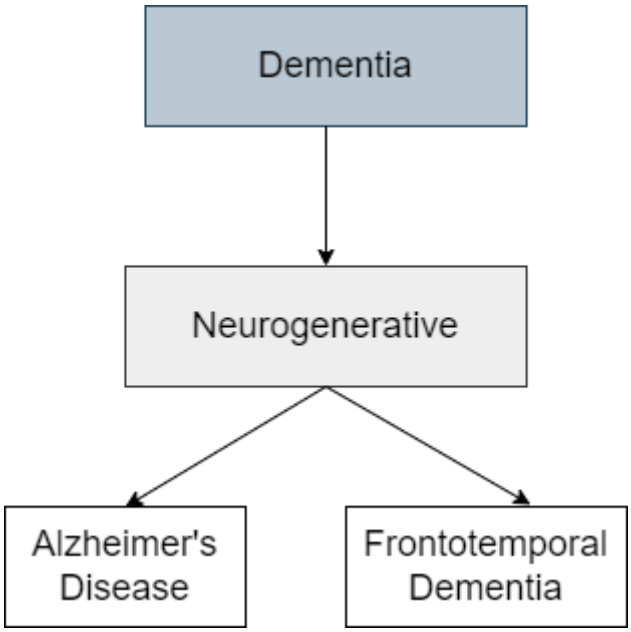

FIGURE :3. 2 DEPICTS THE SUB-TYPES OF NEURODEGENERATIVE DEMENTIA [53].

3.1.1 Alzheimer's Disease

Alzheimer's disease (AD), is a long-lasting neurodegenerative disease affecting the death of brain nerve cells and tissue loss. Usually, it starts slowly and worsens over time. AD usually occurs in the later part of the age or mainly affects the old people, it is not a normal ageing process [45]. The most common symptoms of AD are memory loss and progressive behavioral and intellectual characteristics. According to World Health Organization (WHO), around 55 million people are affected by AD, and by the end of 2030, it is expected that the number of patients will rise to 78 million overall [44]. Currently, there is no standard drug or cure developed to stop or reverse the progression of Alzheimer's disease. The only way to tackle this disease is an early diagnosis and stop the progression of AD. Early diagnosis of AD slows down the degradation of cognitive processes and preserves the life quality as long as possible [46]. Mild Cognitive impairment (MCI) is an intermediate stage between the normal brain and the AD, wherein a person has a gentle change in the psychological capacity that is only to their nearby or surrounding individuals. Progression time lies in the range between 6 months to 3 years. MCI patients that are at a high risk of progressing AD can provide important information for the treatment of the disease [47].

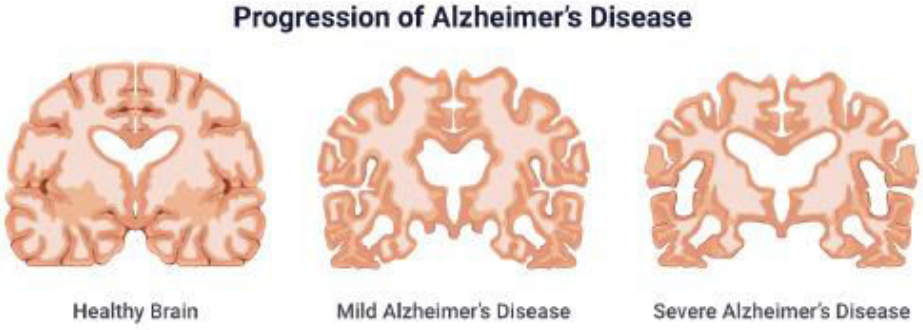

FIGURE :3. 3 DEPICTS THE DIFFERENT CORONAL SLICES FROM A HEALTHY BRAIN, MCI, AND THE AD [44]

FIGURE :3. 4 CORONAL VIEWS OF A HEALTHY BRAIN (LEFT), MCI (MIDDLE), AND THE AD (RIGHT) IN THE NIFTI FORMAT

The brain parts which are initially damaged with the AD are involved in memory, including the entorhinal cortex and hippocampus. When the size of the hippocampus shrinks, the episodic memory and spatial memory are affected badly. Due to the progression of the disease, lately, it also affects the cerebral cortex which is responsible for language, reasoning, and social behavior. Finally, the other parts of the brain are also damaged. Figure 1.1 shows the different coronal slices of the brain that represents how the brain is affected in each stage. Neuroimaging biomarkers are playing a vital role in classifying the different stages of AD or to detect MCI progression to AD [48] [49]. Magnetic resonance imaging (MRI) is the most common biomarker used by physicians to diagnose AD. Physicians in neurology investigate the diagnosis through image or signal analysis [46].

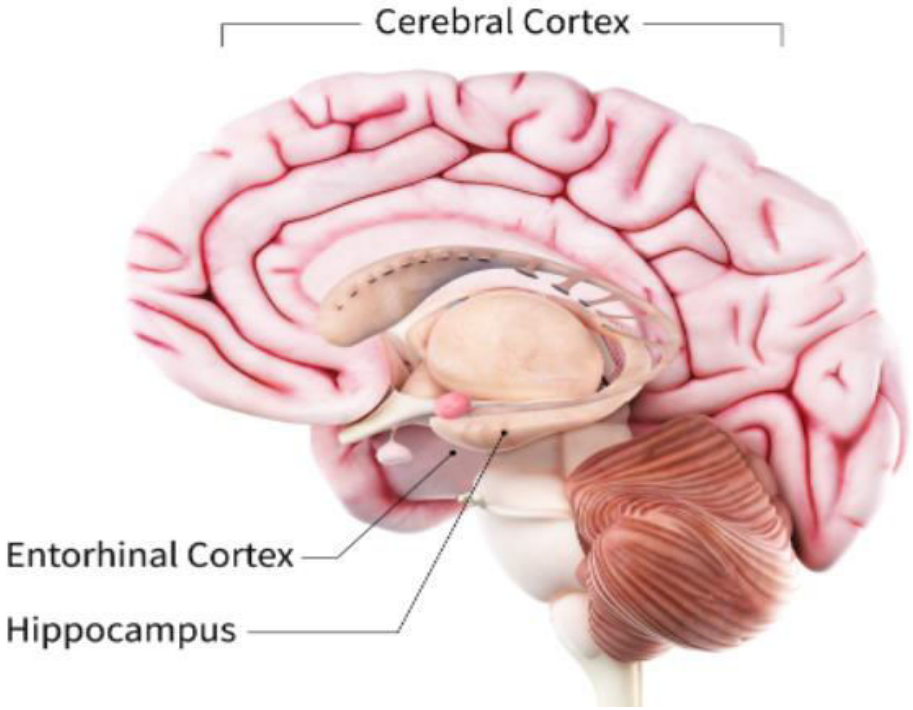

FIGURE :3. 5 DEPICTS THE PARTS OF THE BRAIN THAT ARE AFFECTED BY THE AD [44]

3.1.2 Frontotemporal Dementia

Frontotemporal lobar degeneration (FTLD) is a common reason for early-onset degenerative dementia. FTD is a genetically and also clinically syndrome of heterogeneous, which is attributed to overlying clinical symptoms. These syndromes constituted of changes in behavior, language motor function, and degeneration of the brain regions, frontal and temporal lobes [42]. The defined clinical syndromes are sub-divided in the FTLD spectrum known as bvFTD, svPPA, and nfvPPA. BvFTD which is characterized by early behavioral and executive deficits, svPPA is associated with the semantic variant primary progressive aphasia and the nfvPPA is associated with progressive disorder of speech, grammar, and word output [41] [43]. FTD affects the cognitive and social decline relative to adult capability. FTD most commonly occurs between the age of 40-82 years. The fundamental phenotypes of this disease are a disorder of social activity and execution of any function.

MRI scans interpretation widely depends on the intuition and experience of clinicians, via MRI scans aid clinicians diagnose FTD as an assisting tool [43]. The atrophy of frontal and temporal lobes on MRI scans, with relative preservation of the posterior areas, illustrates the imaging hallmark of frontotemporal labor degeneration. For bvFTD the regions with the most prominent gray matter atrophy are typically the frontal lobes and the interior cingulate cortex, whereas nfvPPA is mostly found on the left-sided in inferior-frontal and insular cortices. For svPPA, the atrophy is predominantly observed in left-sided anteroinferior temporal lobes and temporal gyrus [51].

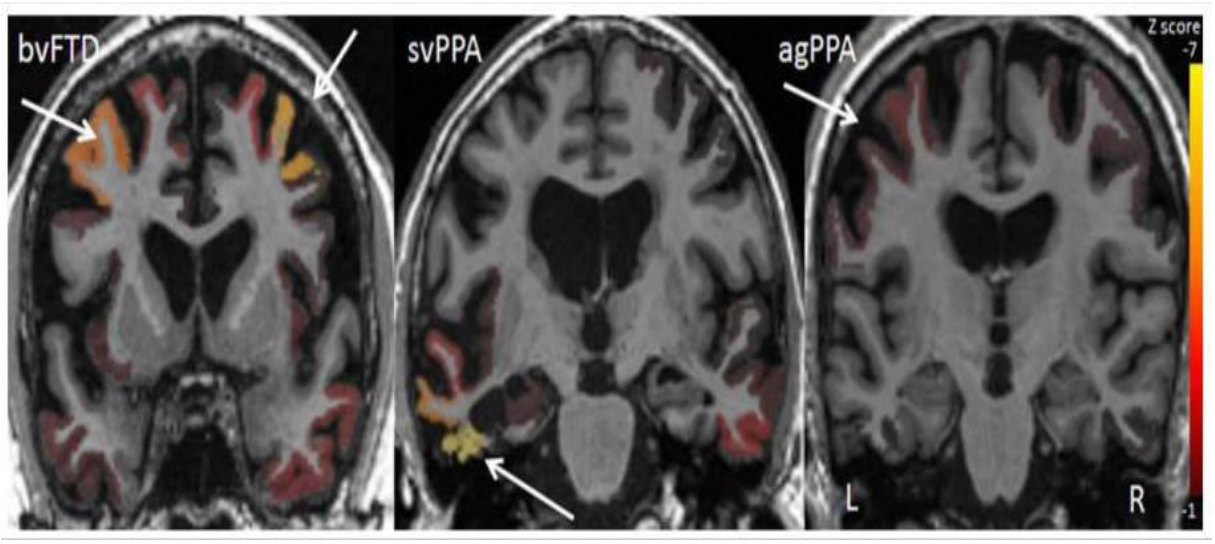

FIGURE :3. 6 DEPICTS THE DIFFERENT VARIANTS OF FRONTOTEMPORAL DEMENTIA [50]

3.2 Convolutional Neural Networks

In computer vision applications, Convolutional Neural Networks (ConvNet or CNN) which is a class of Artificial Neural Networks (ANN) is considered a state of the art method for solving vision tasks. When solving the vision task, ANN with feed-forward architecture is not feasible because of these two problems: Each layer in ANN is connected to its corresponding layers which results in too many parameters in the network [4]; It requires a large amount of memory and large computational power. For instance, considering a CIFAR-10 dataset image size of 32 x 32 x 3 pixels that requires a total number of 3072 weights and this number is still manageable with ANN architecture but these fully-connected architectures would not scale to larger images. Considering an image size of 224 x 224 x 3 which sums up to 150,258 weights for a single fully-connected neuron for the first layer, the number of parameters would increase exponentially in the following layers of the network [6]. This large number of parameters would lead the network to overfit. Another deficiency of the fully

connected architecture is that it completely ignores the spatial information of the image. This shows that it cannot receive information on the correlation between local features of the image [5].

Convolutional Neural Network architecture is the solution for these problems which we faced in fully-connected Networks. The architecture of CNN is identical to the human brain. Moreover, the connectivity pattern of the neurons is also identical to the human brain and the whole organization is inspired by the visual cortex [8]. CNN is developed to automatically and adaptively learn spatial hierarchical features, from low to high features [7]. CNN architecture is composed of different types of layers known as convolution layer, pooling layer, activation functions, and the fully connected layer. In the following sections, these blocks of CNN will be briefly discussed.

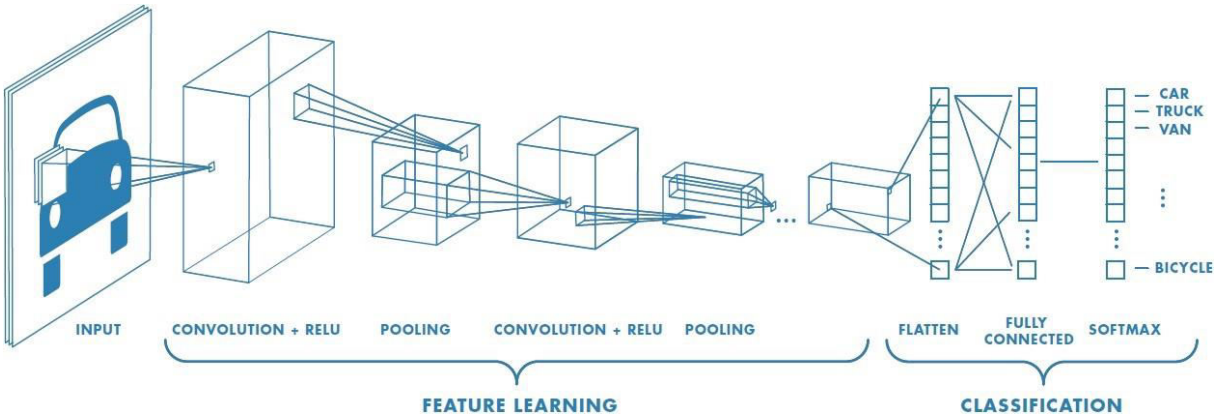

FIGURE :3. 7 A BASIC ARCHITECTURE WITH TWO CONVOLUTIONAL LAYERS FOLLOWED BY A POOLING LAYER AND THE LAST LAYER IS A FULLY-CONNECTED LAYER THAT IS RESPONSIBLE FOR PREDICTING CLASS SCORES [10]

3.2.1 Convolutional Layer

The convolutional layer is the core block or the fundamental layer of the convolutional neural network (CNN). This layer is responsible to extract features from an input image and it typically consists of a combination of linear and non-linear operations such as convolution function and activation functions. The convolution layer performs a special type of linear operation used for feature extraction, where we applied a small square of inputs known as kernels or filters across the whole input image. This element-wise multiplication of each Kernal value with each value of the input image is computed at each positon of the input image and summed up to obtain a feature map. Multiple filters can be applied to the input image to obtain an arbitrary number of feature maps and each feature represents a different feature of the image [7].

Figure 3.9 shows the convolutional operation. This operation requires two hyperparameters that are the size of the kernel and the number of filters to obtain the depth of the feature maps. Usually, the Kernel size which is normally considered are (3 x3), (5 x 5), and (7 x 7) and to retain the size of the input image the padding should be a default value of zero.

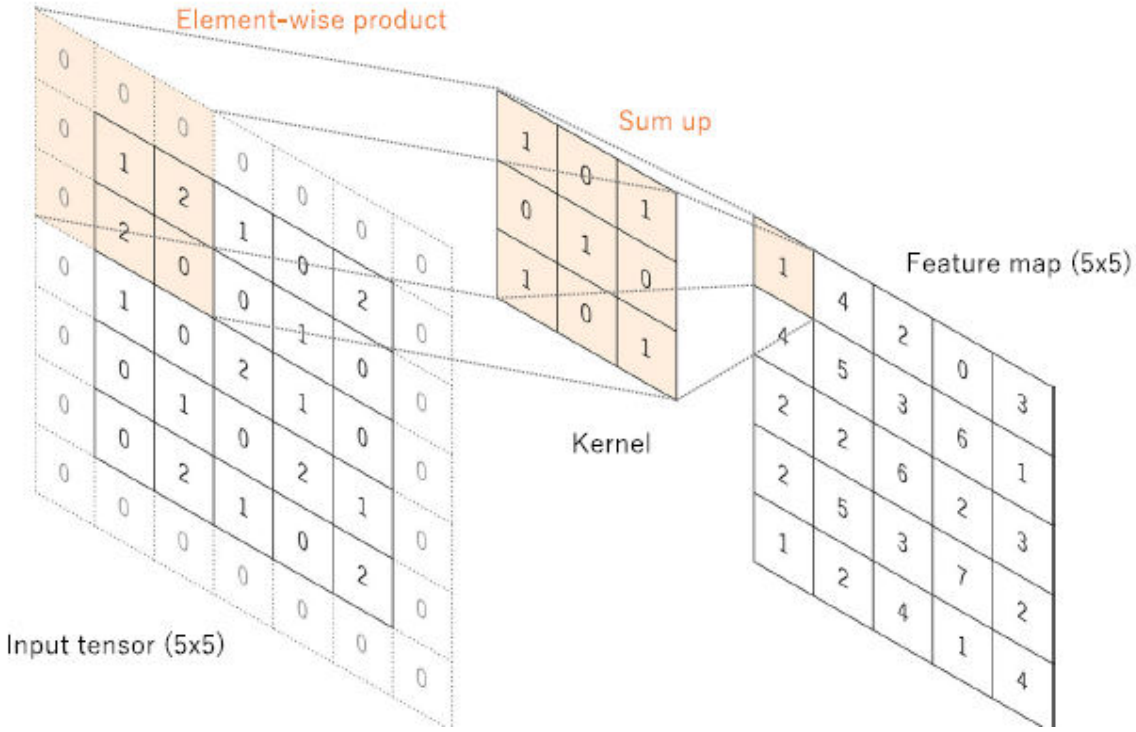

FIGURE :3. 8 CONVOLUTION OPERATION IN THE IMAGE USES (3 X 3) KERNEL SIZE WITH STRIDE=1, PADDING =1, OBTAINS THE SAME FEATURE MAP SIZE AS OF THE INPUT IMAGE (5 X 5). THE ABOVE IMAGE FEATURE MAPS RESULTS, 1 = (0 X 1) + (0 X 0) + (0 X 1) + (0 X 0) + (1 X 1) + (2 X 0) +(0 x 1) [7]

Stride is the number of pixels that moves over the input image. The kernel moves one pixel after each filter operation with the default value (stride = 1). When (stride =2) the kernel moves 2 pixels each time and so on. The output feature map can be calculated using the formula with stride=s, padding=p, input image = (n x n) and filter size (n x n) as (($\frac{n+2p-f}{s}+1$) x ($\frac{n+2p-f}{s}+1$)). In the case of RGB images, the image and filter size can be modified as: (n x n x n_c) and (n x n x n_c). The depth of the output feature is determined by the number of filters used in the convolution operation. The output of the convolution function propagates through an activation function that produces the non-linear representation to extract the complex features of the input image which can then map the input data to the classification output. The kernel/filters setting linked to CNN can be adjusted during the training of the network [7].

3.2.2 Pooling Layer

The pooling layer is responsible for down-sampling which reduces the dimension of the feature map and keeps the spatial information. Due to the decrease in feature map dimension, the learnable parameters are also decreased for the following convolution layers and it introduces a translation invariance to small shifts and distortions. While the hyperparameters such as kernel size, padding, and stride work similarly as in the convolution operation. The most common type of pooling method is the max pooling. Max pooling extracts the maximum value from each of the input patches and discards all the values [11]. A common practice for the max-pooling function is with the filter size of (2 x 2) and the stride of 2. Other two types of pooling functions exist such as minimum pooling and global average pooling.

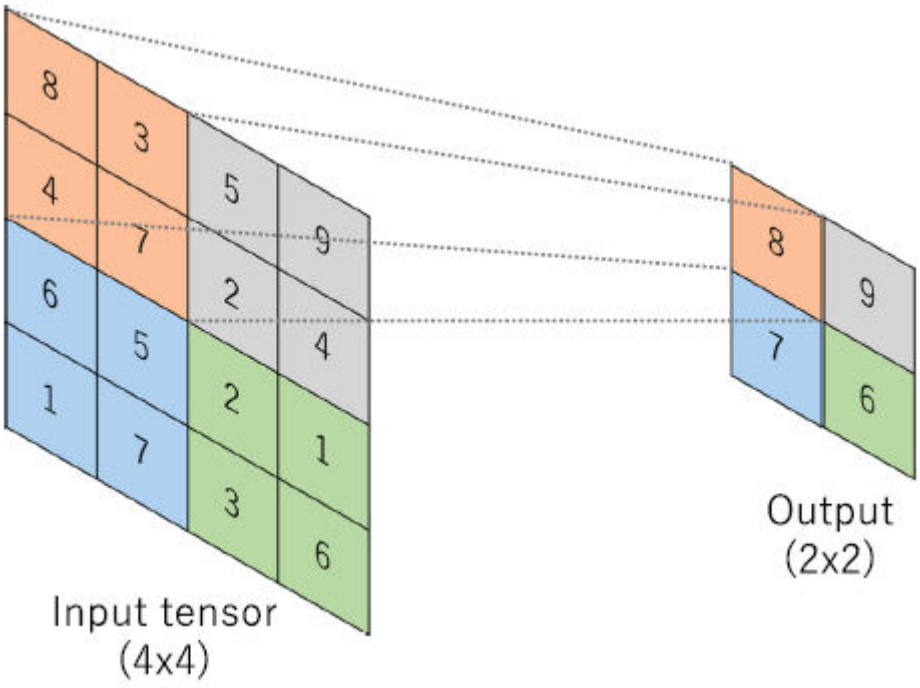

FIGURE :3. 9 MAX POOLING OPERATION ON (4 X 4) IMAGE PRODUCES THE FEATURE MAP SIZE OF (2 X 2) WITH STRIDE=2, AND PADDING = 0 [7]

3.2.3 Activation Function

The activation function has the fundamental role in training the deep neural networks and the choice of these activation functions has a significant role in task performance. Nowadays, the ReLU activation function is commonly used for training deep neural networks. ReLU activation function can easily be optimized as compared to the sigmoid or tanh functions, because of its simplicity and its effectiveness the gradients easily flow when the input to the ReLU activation function is positive [7] [12]. The activation function transforms the linear input into non-linear output and it helps the network to extract complex features and patterns from the image. The optimal choice of activation function is very important for a network to minimize the loss during backpropagation by exploiting the gradient descent algorithm (discussed in the following section of this chapter). There are many other types of activation functions such as sigmoid, softmax, ReLU, LeakyReLU, tanh, etc. [13].

$$\text{Sigmoid, } \sigma(x) = \frac{1}{1+\mathrm{e}^{-\mathrm{x}}} \quad \textbf{(3.1)}$$

$$\text{tanh, } \sigma(x) = \tanh(x) \quad \textbf{(3.2)}$$

$$\text{ReLU, } \sigma(x) = \max(0, x) \quad \textbf{(3.3)}$$

$$\text{Softmax, } \sigma(x) = \frac{e^{x_i}}{\sum_j e x_i} \quad \textbf{(3.4)}$$

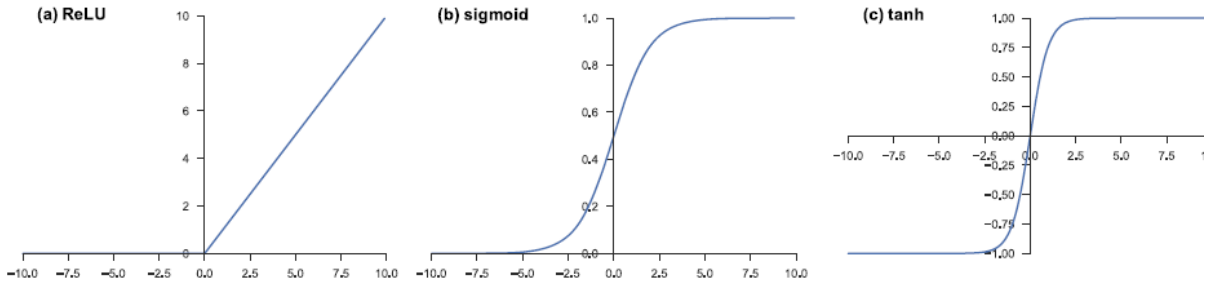

FIGURE :3. 10 COMMON ACTIVATION FUNCTIONS USED IN DEEP NEURAL NETWORKS (A) RELU (B) SIGMOID AND (C) TANGENT HYPERBOLIC [7]

The sigmoid function is also defined as the squash function because it maps the input value between 0 and 1. This is a non-linear activation function and the gradient below or above the defined value would close to zero. In backpropagation, the gradient becomes zero, the neuron becomes useless and the network stops learning the learnable parameters. This problem in deep learning is known as the vanishing gradient problem. The tanh function which is known as the zero-centered function lies in the range of -1 to 1. These tanh and sigmoid functions caused a problem in the backpropagation and did not optimally address the vanishing gradient problem. The ReLU activation function which is a widely used activation function solves the problem of vanishing gradient problem. The ReLU function allows all the positive values of the input and it maps all the negative values to zero. ReLU is proved to be the best activation function in the hidden layers [13]. In the last layer of the neural network, the softmax function is mostly used. The softmax function produces the probabilistic values for each class of the classification. This function normalizes the real values from the last fully connected layer to the target class probabilities, where each value lies in the range between 0 and 1 and the sum of the class probabilities is equal to one [7][13].

3.2.3 Normalization Layer

Normalization layers used in CNNs are responsible to make the output of each layer into a specific range that experimentally shows faster convergence. It is only evident in the deep neural network when the size of the normalization should be large enough. There are various normalization techniques introduced in deep neural networks such as Batch Normalization, Group normalization, weight normalization, etc. and each normalization technique normalizes the values differently. However, the contribution of these normalization layers is minimal and these normalization layers are no more in use [4].

3.2.4 Fully Connected Layer

The last layer of the CNN is known as the fully-connected layer (fc) in which all neurons of the layer have a connection with all the previous layer neurons. Before the fully-connected layer, the last output feature map of the convolution layer is normally flattened which means it translates a 1D array of numbers that are further connected to one or more fully-connected layers. This is referred to as dense layers where each input is connected to every output by a learnable weight [7]. As the features extracted by the convolution layers and down sampled by the pooling function in the network, they are mapped by multiple fully-connected layers followed by a (ReLU) activation function to the final outputs of the network, such as the probabilistic values of each class in the classification task.

3.3 Training Convolutional Neural Network

Training of CNN means adjusting the learnable parameters of the networks such as the size of the kernel, number of filters, stride, padding, weight, and biases of fully-connected layers to minimize the difference between the predicted outputs of the network with the target label. Ground truth and predicted output difference are calculated by using the loss function. Initially, the network is initialized with the random learnable parameters (weight and biases) of a certain range, model performance is calculated for the specific kernel during the forward pass and the weights are calculated using the loss function. The learnable parameters are being updated to minimize the loss calculated in the forward pass by an optimization algorithm and the process is known as backpropagation [7].

3.3.1 Loss Function

The loss function is also known as the cost function. It measures the difference between the predicted outputs of the network via the forward pass and the given ground truth labels. The common loss function used for the multi-class classification task is the cross-entropy, Mean Squared Error (MSE) is used for regression to continuous values. The main goal of the loss function is to minimize the cost function calculated in the forward pass and update the learnable parameters in each iteration of the training. The loss function is defined as one of the hyperparameters which needs to be determined while optimizing the network for some specific problem [4].

3.3.2 Backpropagation

Minimizing the cost function during the forward pass calculation and updating the learnable parameters such as kernel/weights, hence improving the model performance is known as Backpropagation. Evaluating the model performance could be calculated using the loss function such as MSE or cross-entropy. Backpropagation is responsible for only calculating the gradients and updating learnable parameters in each iteration.

Optimizing the model performance can be done using the algorithm known as Gradient descent. These gradients of the loss function identify in which direction the loss function has the sharpest increase and update the learnable parameters in the negative direction of the gradient to minimize the loss for desired prediction outputs, hence, it is named Gradient descent. The loss function is calculated across all the weights of the network and to adjust these learnable weights we need to apply the partial derivatives. The hyperparameter needs to be considered as the learning rate which is the step size by which the network can improve. The learning rate value lies in the range of (0.1 to 0.0001). Computation of weights is defined as [25] [17],

$$\Delta\omega = r\ x\ (\frac{\partial C}{\partial \omega_1} + \frac{\partial C}{\partial \omega_2}, \dots, + \frac{\partial C}{\partial \omega_q}) \quad \textbf{(3.5)}$$

Where $\Delta\omega$ is referred to as the updated weight vector. C is the cost function of the model, 'r' is denoted as the learning rate and 'q' denotes the number of weights in the network. Due to memory constraints, it cannot be possible to process the samples of a large dataset in one iteration. So computing the gradient of the loss function can be done using a small subset of the dataset at once which is known as mini-batch. This method is known as Stochastic Gradient Descent (SGD). Optimizing the neural network with a hyperparameter of learning rate along with the mini-batch size is very important [15]. Many advanced versions of SGD have been proposed and widely used such as RMSprop, AdaDelta, AdaGrad, AdaMax, NAdam, Adam, etc. Due to the robustness and effectiveness for a wider range of applications, Adam is the most popular optimizer [14].

3.3.4 Hyperparameters

Adjustment of the learnable parameters such as kernel/filters, weight, and biases of the network can be done by training a CNN or neural network. These learnable parameters are known as parameters. We have discussed a few of the hyperparameters in the previous sections that control the behavior of the learnable parameters known as Hyperparameter [7]. These hyperparameters are defined as the characteristics of the CNN that affect the general architecture or their functionality. These hyperparameters can be defined before the training of the model and can have a significant influence on the model's performance [19]. Important hyperparameters are defined below:

- Batch Size: Small portion of the dataset which is parallelly executed by the network (Efficiently use of the computational resources)
- Epochs: Number of iterations, in each iteration the whole data is passed through the network
- Loss Function: This function is used to compute the loss (error) of a batch
- Optimizer: The optimizer is used to minimize the error of the loss function (optimize the model performance)
- Learning rate: The step size in which the weights are updated in the optimizer
- Activation Function: Layer-specific activation function that squashes the output feature value into the specific function value range.

Each neural network or CNN has its specific hyperparameter setting to achieve the best network performance. Usually, we applied brute force methodology known as grid search. In this grid search methodology, we specified the specific value range for each of the hyperparameters that has to be optimized. For instance, Batch size: [16, 32, 64], learning rate: [0.1, 0.001, 0.0001]. The model's performance is being evaluated on each of the hyperparameter settings after complete training of the model. In this case, the grid search algorithm has to complete the three complete training sessions. After completing the training session, we evaluated the model performance on the decided performance metrics and kept those hyperparameter results that provide the best model's performance [19].

3.4 Data Preparation for Training and Evaluation

For training of a model or neural network data preparation is very essential. Typically, CNN or neural networks require a large amount of data to train the model, where the whole dataset is divided into three parts such as training set, validation set, and test set [7].

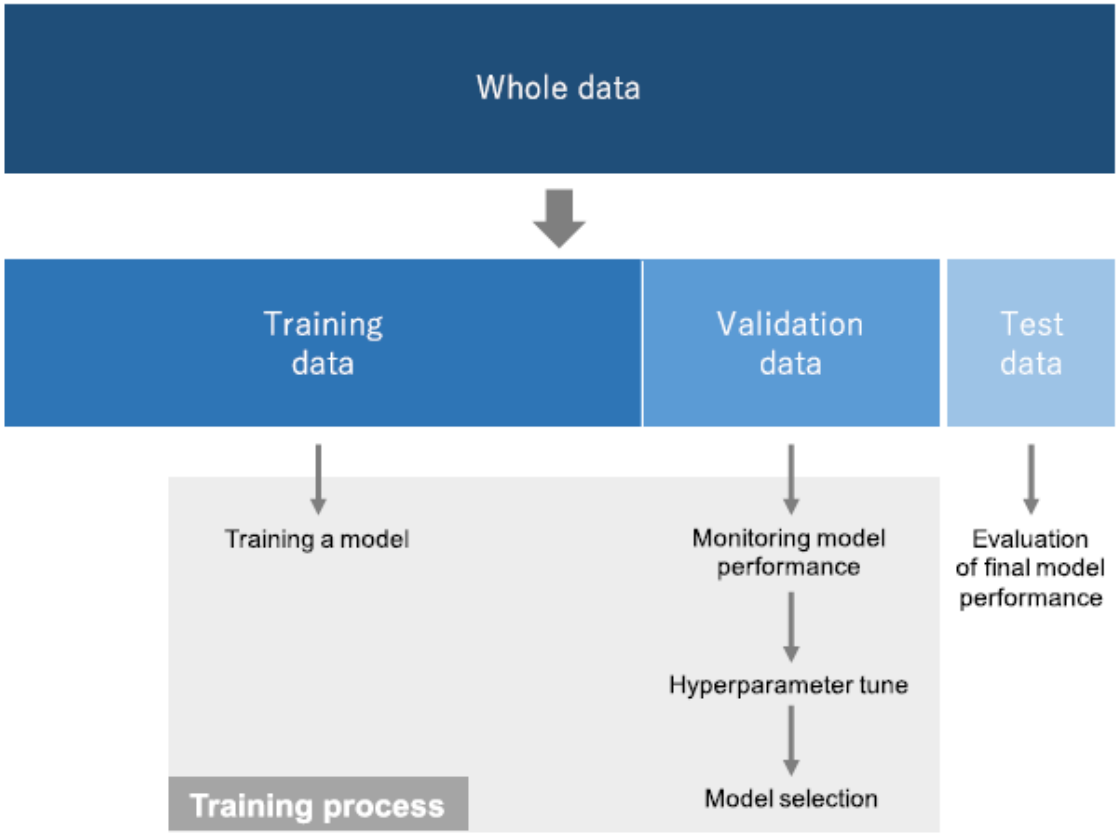

FIGURE :3. 11 DATA PREPARATION AND TRAINING PROCESS OF THE MODEL [7]

The most optimal way of evaluating the model performance in Machine Learning is Cross-Validation. In k-cross validation we randomly divide the dataset into k-folds without replacement. One k-fold is held out as a test set and the remaining set are available for training the model. Generally, k-cross validation is used for model tuning, to determine the optimal hyperparameter settings that result in the best generalization performance. K-cross validation is considered the robust and gold standard technique for evaluating the model's performance [17].

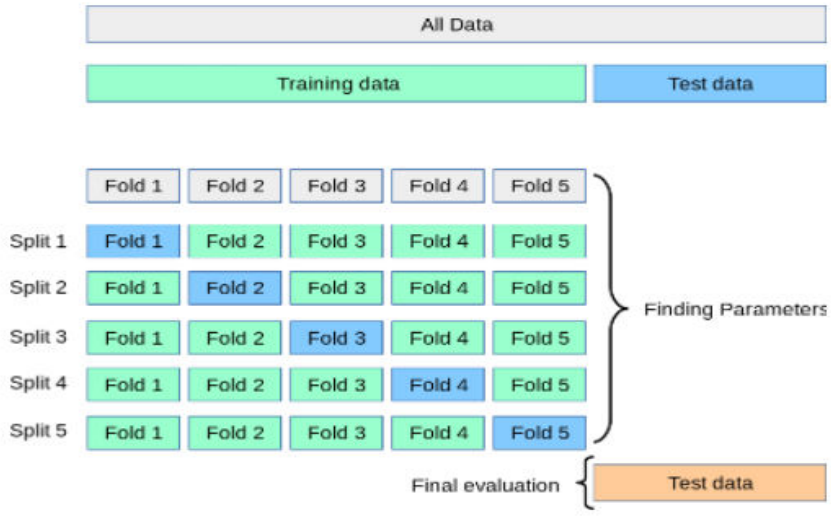

FIGURE :3. 12 5-FOLD CROSS-VALIDATION VISUALIZATION [18]

Figure 3.12 depicts a 5-fold cross validation on the training data. In each iteration, the model is trained on different folds and then the model is evaluated on the held out fold. The performance of each model is done by the averaging and standard deviation calculation and obtains a robust model's performance estimation [17].

3.4.1 Performance Metrics

In achieving the optimal classifier, the performance metric plays a crucial role during model training. Hence, opting for a suitable performance metric is an essential key for discriminating and obtaining the optimal classifier. In this work, different evaluation metrics have been used using the Sklearn library and PyTorch evaluation functions [26].

Accuracy:

This metric measures the ratio of correct predictions over the entire number of instances evaluated.

Precision:

This metric is used to compute the positive instances that are correctly predicted from the total predicted instances from the positive class.

Recall:

Recall metrics are used to measure the fraction of positive instances that are correctly classified

F1-Score:

F1-score metrics are used to measure the harmonic values between precision and recall values.

Matthews Correlation Coefficient:

The Matthews correlation coefficient is used as an evaluation metric in machine learning for measuring the quality of binary and multiclass classification. This evaluation metric is in essence a correlation coefficient value between'-1' and '1'. The value of 1 represents the perfect prediction and 0 represents the average or random prediction,-1 represents the inverse prediction.

Balanced Accuracy:

The balanced accuracy metric is used in binary and multiclass classification problems to deal with imbalanced datasets. It computes the average recall obtained in each class. The best value is 1 and the worst is zero.

3.5 3D CNN

The input to 2D CNN is only restricted to the spatial dimensions of the image. To process the 3D medical images, it is required to process the complete 3D volume of an image. Usually, 3D CNNs can be used for Medical image classification or video processing. However, the general structure of the 3D CNN and 2D CNN networks is similar except the dimensions of the tensors are different. The input to 3D CNN is a 5-dimensional tensor. For instance, tensor shape: [batch size, height, width, depth, channels]. Batch size is commonly known as the hyperparameter of the network which performs the parallel processing of the data by the CNN. The convolutional layer of the network that processes the input volume must also have the 5-dimensions. Normally, the convolutional layer of 3D CNN is composed of 5 dimensions [height, width, depth, in_features, and out_features [24]. The below figure depicts the 3D convolution operation on the 3D input volume [23].

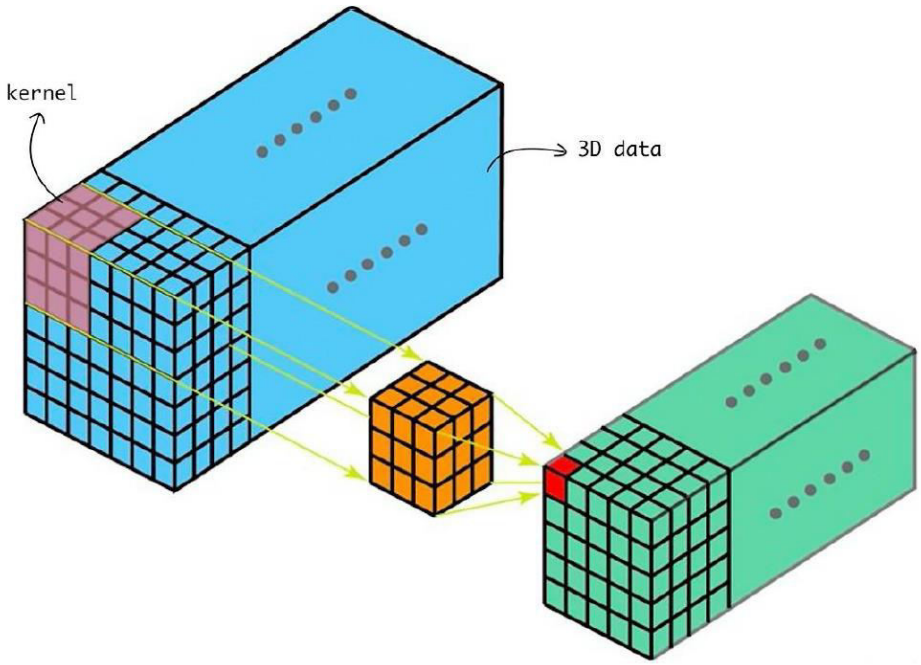

FIGURE :3. 13 3D CONVOLUTION OPERATION ON THE 3D INPUT VOLUME, THE CONVOLUTIONAL KERNEL DEPICTS IN ORANGE AND THE OUTPUT OF THE CONVOLUTION OPERATION DEPICTS IN THE GREEN

3.6 Learning Methodologies

Machine learning methodologies can be separated into supervised, semi-supervised, and unsupervised learning. Supervised learning defines as the data samples being labeled during the training phase, however unsupervised learning means that the data samples are unlabeled. Semi-supervised is defined as a hybrid sort of learning approach where partial data samples are labeled and this affects the final performance of the model. We choose the learning methodology according to the task.

3.6.1 Supervised Learning

Supervised learning deals with two types of problems: regression and classification. Classification can be defined as classifying the input data as a part of a specific category. In the training phase, the machine is shown an input image and it produces an output in the form of a vector of scores, each score belongs to a specific category. The trained network is evaluated on how perfectly the network can identify the unseen data. The regression problem deals with the continuous data and it depends on a lot of variables that need to be investigated. An example of a regression problem is linear regression which studies the relationship between the dependent and independent variable and by giving a specific input it produces a future output based on input. Supervised learning is the best approach for the problem where the dataset is labeled [1].

3.7 Transfer Learning

Deep learning algorithms have been successfully learning the high-level features from a large amount of training data which goes beyond the traditional machine learning methods. A deep learning algorithm allows networks to automatically extract features, whereas traditional machine learning algorithms require them to manually design the features that become an additional load to users. Data dependency is one of the critical problems in the area of deep learning. Deep learning algorithms have a very profound dependence on a large amount of training data compared to traditional machine learning algorithms, because it requires massive amount of training data. However, medical image

data are hard to collect because it requires a lot of professional expertise to label them and the high cost associated with the acquisition of the medical data [20] [21] [22].

The underlying assumption to transfer learning is that every image consists of generic features such as circles, and edges, which describe, for instance, dogs, tables, or human brains as well [20]. As explained in the previous sections, CNN extracts the generic features in the initial layers, and in the subsequent layers, it extracts more detailed features. Transfer learning makes use of this assumption by taking pre-trained CNNs (usually trained on extremely large datasets such as ImageNet, which contain 1.6 billion images of 1000 classes) and transforming them into a customized problem [7][22]. The portability of these generic learned features is a big advantage of deep learning that makes it useful in various domain tasks with small training datasets. Many models are openly available that are trained on ImageNet datasets such as AlexNet [31], VGG [30], GoogleNet [28], and ResNet [29] that can be transformed for different applications. Traditionally, there are two ways to use the transfer learning approach: fixed feature extraction and fine-tuning [7].

In the process of fixed feature extraction, we would remove the fully connected layer from the network pre-trained on ImageNet and preserve the remaining network. The remaining network would consist of convolutional layers, pooling layer, and activation functions, which is commonly referred to as feature extractor. In this case, a machine learning classifier such as SVM, Random Forest, or the modified fully connected layer, can be concatenated on the top of the feature extractor, which results in training limited up to the added classifier on a customized dataset. This type of transfer learning is not very common in deep learning research on medical images because of the difference between ImageNet images and the given medical images [7].

The second process in transfer learning which is commonly used in the medical domain is known as fine-tuning. It does not only replace the fully connected layers with a machine learning classifier of modified fully connected layers but it also allows unfreezing of the convolutional base layers according to the task requirements. All the layers in the convolutional base can be fine-tuned or some layers can be fine-tuned because as we know the upper layers contain more generic features so we freeze those upper layers and retrain the deeper layers which consist of detailed features according to the specific tasks [7].

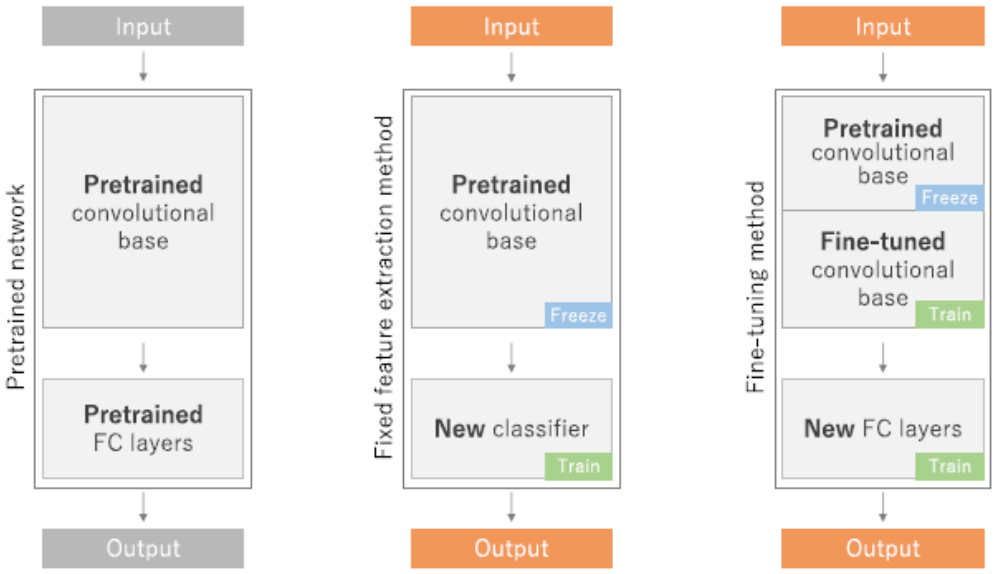

FIGURE :3. 14 DEPICTS THE DIFFERENT METHODS OF TRANSFER LEARNING USED IN THE DEEP LEARNING [7]

3.8 Regression Algorithms

In this section, we discuss the most commonly used regression algorithms.

3.8.1 Linear Models

Linear regression is considered one of the most standard models used for regression problems. It illustrates the linear regression in the following way. The response of linear regression is a linear function of the inputs [70].

$$y(x) = W^t x + € = \sum_j^D w_j x_j + \varepsilon \quad \textbf{(3.6)}$$

Where W^T x is the product multiple of input vector 'x' and the weight vector 'w'. 'ϵ', denotes the residual error of the linear predictions and the ground truth label. Generally, € is assumed as normal or Gaussian distributed, N ≈ (μ, σ^2) (μ defines as the mean and σ^2 define as variance). The model also asserts the conditional probability, so it is rewritten in the following way

$$\rho(y \mid x, \theta) = N(y \mid \mu(x), \sigma^2(x)) \quad \textbf{(3.7)}$$

Let us assume the general case, where 'μ' is a linear function of 'x', so μ=$W^T x$ and the noise is constant $\sigma^2(x) = \sigma^2$. So $\theta = (\omega, \sigma^2)$ are linear regression model parameters [70].

Linear regression models can model non-linear relationships if we substitute 'x' with the non-linear function of the inputs, $\emptyset(x)$

Furthermore, the regression can also be used in classification problems, in this case, it is known as logistic regression. The research [1] explains the logistic regression as follows. Assume the y follows the Bernoulli distribution, y € {0, 1}.

$$\rho(y \mid x, \omega) = Ber(y \mid \mu(x)) \quad \textbf{(3.8)}$$

Where $\mu(x) = E[y|x] = \rho(y = 1 \mid x)$. Moreover, we compute a linear combination of all the features and then we defined it as $0 \leq \mu(x) \leq 1$.

$$\mu(x) = sigm(W^T x) \quad \textbf{(3.9)}$$

Where sigma (γ) is referred to as sigmoid function, which is also defined as logit or logistic function. Which can be defined as

$$sigm(\gamma) = \frac{1}{\exp(-\gamma)} = \frac{e^{\gamma}}{e^{\gamma}} + 1 \quad \textbf{(3.10)}$$

The sigmoid function is also referred to as the squashing function because it maps the input value between the range of [0, 1]. It is important here for the probabilistic understanding of the output. Combining the equations 4.9 and 4.10, we get

$$\rho(y|x, \omega) = Ber(y \mid \text{sigm}\,(W^T x)) \quad \textbf{(3.11)}$$

This is known as logistic regression. If we apply the threshold at the output probability of 0.5, we can introduce the following decision rule

$$y(x) = 1 \Leftrightarrow p(y = 1|x) > 0.5 \quad \textbf{(3.12)}$$

For instance, logistic regression is $\rho(y_i = 1 \mid x_i, \omega)$= sigm $(\omega_0 + w_i x_i)$. Assume now that sigm $(\omega_0 + w_1 x)$ =0.5. We can surely draw a vertical line at $x = x^*$, the line is defined as a decision boundary. Everything that exists on the left side of the line is 0 and everything that exists on the right side of the line is 1 [70]

Chapter 4

Methods

This chapter illustrates all the approaches and methods which we used to achieve the goal of the thesis. The goal of this thesis is to classify the two dementia disease sub-types Alzheimer's disease (AD) and Frontotemporal dementia (FTD). Although we started the implementation by developing transfer learning models for the classification of Alzheimer's disease. We utilized the Resnet-18 architecture with their pre-trained weights. Later, we extracted AD model features for the embedding of FTD model features to classify the FTD disease and their variants in a few-shot learning manner. The method discussed in this chapter will provide us a profound understanding of embedding learning of few-shot learning.

4.1 3D ResNets Architecture

The 3D Resnet architecture was built by Facebook and published in 2018. 3D ResNets architecture was primarily developed to study the temporal relationship in video clips. The motivation derives from the observations that 2D CNN applied to induvial video frames have remained solid performance for the action recognition task [53]. Hence, the downside effect of using these networks is that it requires larger datasets and high computational resources. To compensate for these negative effects and benefited from the power of 3D convolutions, the researchers proposed to use 3D CNNs in the framework of residual learning and explore the advantages of 3D CNNs over 2D CNNs. The R3D architecture is depicted below in table 2. The R3D architecture consists of 5 convolution blocks that are generally responsible for meaningful feature extraction. Each convolution block is stacked of two convolution layers followed by a Batch normalization and the ReLU activation function. The convolution blocks are finally connected with spatial-temporal pooling to reduce the dimensions of the network and followed by a fully connected layer, resulting in classification output. This architecture is considered in our transfer learning model settings.

Layer name	Output size	R3D-18
Conv1	Lx56x56	3x7x7x64,stride 1,2x2
Conv2_x	Lx56x56	$\begin{bmatrix} 3\times3\times3,64 \\ 3\times3\times3,64 \end{bmatrix} \times 2$
Conv3_x	$\frac{l}{2}\times 28 \times 28$	$\begin{bmatrix} 3\times3\times3,128 \\ 3\times3\times3,128 \end{bmatrix} \times 2$
Conv4_x	$\frac{l}{4}\times 14 \times 14$	$\begin{bmatrix} 3\times3\times3,256 \\ 3\times3\times3,256 \end{bmatrix} \times 2$
Con5_x	$\frac{l}{8}\times 7 \times 7$	$\begin{bmatrix} 3\times3\times3,512 \\ 3\times3\times3,512 \end{bmatrix} \times 2$
Spatiotemporal pooling, fc layer with softmax		

TABLE:4. 1 ARCHITECTURE MODEL BLOCKS OF RESNET-18 [56]. THIS ARCHITECTURE IS CONSIDERED IN OUR TRANSFER LEARNING MODELS SETTINGS

The R3D architecture is primarily designed for the classification of video clips by using 3D convolution to extract the temporal features in the video. They considered Kinetics-400, sports-1M kinetic video clips benchmark datasets to train these R3D networks from scratch. The model consists of 18 layers and the network takes an input of clips consisting of L RGB frames with a size of 112 x112. The network uses one spatial downsampling at conv1_x implemented by the stride size of 1 x 2 x 2 and spatiotemporal downsampling at cnv3_x, conv4_x, and conv5_x. The model is initially trained on the kinetics-400 dataset and evaluated on the held-out test set of kinetics-400. Video clips are initially

rescaled to 128 x 171 and then each clip is generated by random cropping of window size of 112 x112. The model is trained on these hyperparameters: epochs=45, batch size = 32 clips per video, learning rate initialized from 0.01 and gradually decreases with a factor of 10 and batch normalization is applied to each convolution layer [56].

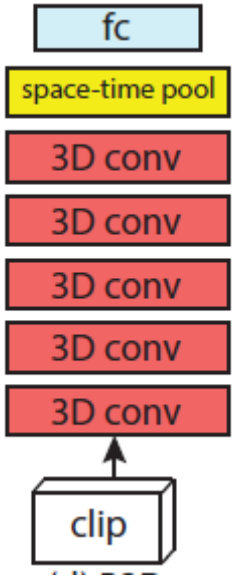

FIGURE 4. 1 NETWORK ARCHITECTURE OF 3D RESNETS [56]

In this thesis, pre-trained 3D ResNet architecture has been used for the classification of Alzheimer's disease and the features generated by the AD model are further being used as prior knowledge for the downstream task to classify FTD disease with fewer training samples. Both methods of transfer learning (feature extraction and fine-tuning) have been used in this work for the classification of Alzheimer's disease and finetuning model features are being used as prior knowledge to perform the downstream task of FTD disease.

4.3 Embedding Learning

The embedding learning method embeds each data sample from X_i € X Ç $\mathbb{R}^d$ to a lower-dimensional space z_i € ƶ ⊆ $\mathbb{R}^m$, which means similar data samples are close together and non-similar data samples are differentiated. In the lower dimensional ƶ, it's possible to develop a smaller hypothesis space Ĥ which only consists of a few training samples. The embedding function is primarily learned from prior knowledge, and also can use task-specific information from D_{train} [69].Embedding learning consisted of important components: a function 'f' which embeds the test data samples from D_{test} to ƶ (b) a function 'g' which embeds training samples from D_{train} to ƶ. Embedding learning are classified into three groups: (i) task-specific embedding model, (ii) task-invariant model, and (iii) hybrid embedding model, which encodes both task-invariant and task-specific knowledge.

In this thesis, our methodology motivation derives from the hybrid embedding model. Learnet [74] and TADAM [73] are the two most popular methods of hybrid embedding learning. The hybrid embedding model learns the embedding function by using generic task-invariant information from prior knowledge by task-specific knowledge from D_{train}.

4.4 Problem formulation

We establish some prefaces for defining our algorithm through which we achieve our thesis goal. The collection of meta-training tasks in our case ADNI dataset is defined as$\{\left(D^{train}, D^{test}\right)\}_{i=1}^{I}$. The notation (D^{train}, D^{test}) termed as training and testing sets of a downstream FTD dataset, which contain few training samples. Training samples of FTD dataset $D^{train} = \{(x_{t,}y_t)\}_{t=1}^{T}$ and testing samples $D^{test} = \{(x_{t,}y_t)\}_{q=1}^{Q}$. The training samples of the FTD dataset are known as Support set and testing samples denote as Query set in the FSL scenario. Both training and testing samples are sampled from the same distribution. In the meta-training task, we develop an AD classification model by using the pre-trained ResNet-18 model to classify Alzheimer's disease and the extraction of model features, which is being used for calculating the embedding function of frontotemporal data samples [68].

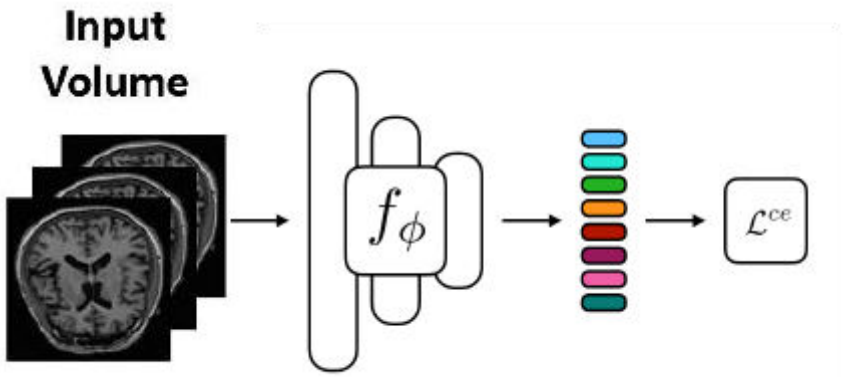

FIGURE 4. 2 IN A META-TRAINING SETTING, WE TRAIN AN ALZHEIMER'S CLASSIFICATION TASK ON THE ADNI TRAINING DATA TO LEARN AN EMBEDDING MODEL. THIS MODEL IS RE-USED AT THE META-TESTING TASK TO EXTRACT AN EMBEDDING FOR OUR SIMPLE LOGISTIC REGRESSION [68].

The task of meta-training is to learn a transferrable embedding model fØ, which is generalizable to any new task. We present that fine-tuning the Alzheimer classification model can generate a powerful embedding for our base learner. For a task (D^{train}, D^{test}) sampled from the FTD dataset distribution. The base learner is trained on D^{train} of FTD dataset (support set). The base learner was initialized as multi-variate logistic regression. The parameters of logistic regression of Θ = {W, b} denotes a weight term and a bias term.

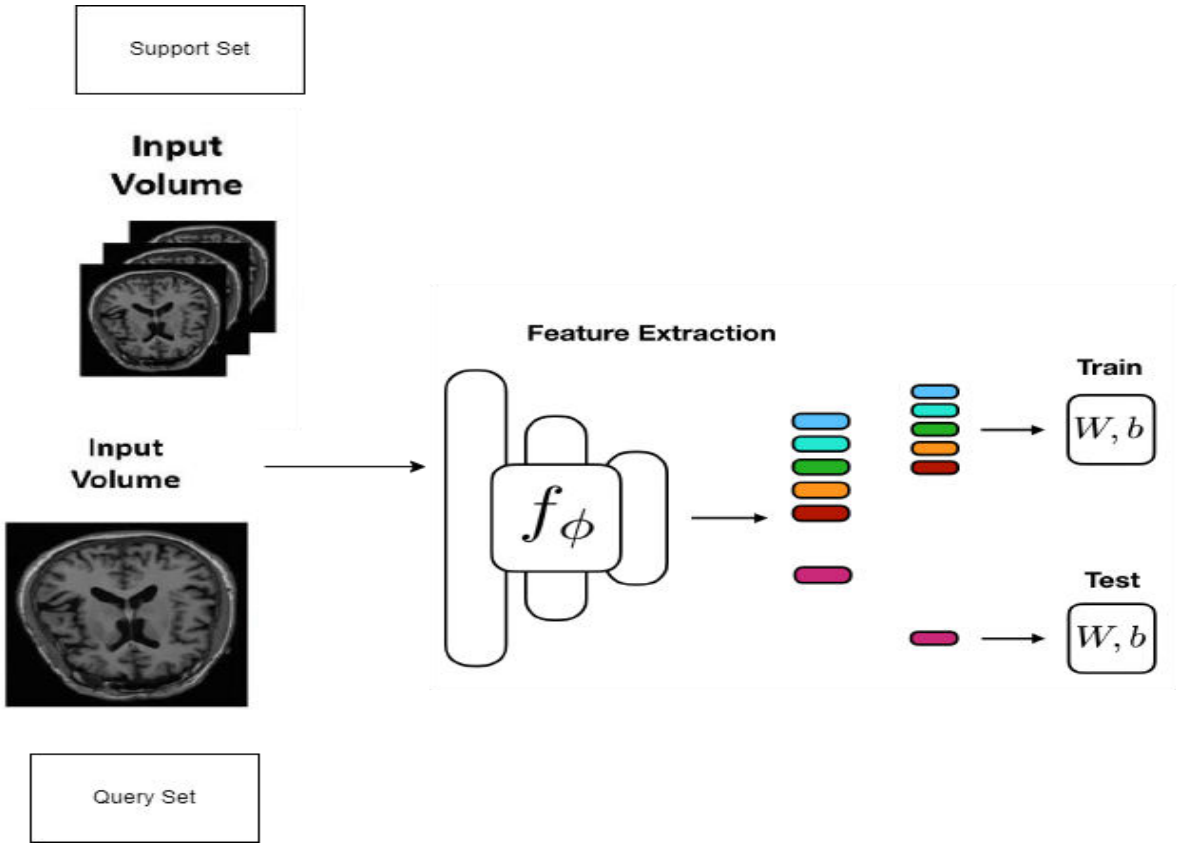

FIGURE 4. 3 SHOWS META-TESTING CASE FOR 3-WAY 1- SHOT CLASSIFICATION, 3 SUPPORT IMAGES, AND ONE QUERY IMAGE ARE TRANSFORMED INTO EMBEDDING USING THE FINE-TUNE AD CLASSIFICATION MODEL. A LINEAR MODEL (LOGISTIC REGRESSION IN OUR CASE) IS TRAINED ON 3 SUPPORT EMBEDDINGS; THE QUERY IMAGES IS TESTED USING THE LOGISTIC REGRESSION[68]

In this thesis, the proposed methodology is that we need optimal feature representations which we use as prior knowledge for the downstream task of FTD disease classification. We learned the optimal feature representations by developing a CNN model based on a large ADNI dataset and utilizing the pre-trained video ResNet-18 model. Moreover, we transfer these optimal feature representations to the model that will be trained with the fewer training samples of the FTD dataset. As a result, we increased the feature size of the downstream task model so that it can achieve the optimal diagnostic accuracy of the FTD disease with fewer training samples.

Chapter 5

Study Data and Implementation

This chapter of the thesis describes the overall implementation steps of the different 3D Convolutional neural networks used to classify FTD and Alzheimer's disease and study the data used in this work. In chapter 4, the methods have been discussed that were used to achieve the goal of the thesis. This chapter explains the training process of each of the CNN models and the classification of FTD disease using the few-shot learning method. The defined tasks of this thesis are briefly explained in this chapter. Initially, we had to find the best methods for the classification of Alzheimer's and FTD disease with fewer training samples. The practical implementation was started by transforming the pre-processed ADNI dataset for the classification of Alzheimer's disease and the feature extraction for the few-shot learning method. Three models were developed for Alzheimer's disease using transfer learning with different network settings. Extracted meaningful features from trained CNNs are transferred to few-shot classification tasks. Furthermore, three different experiments have been performed for the classification of FTD disease. All the developed models for both diseases have been evaluated on Sk-learn metrics and the gold standard evaluation technique in machine learning "Cross-Validation".

5.1 Data collection

The primary goal of this thesis is to develop 3D CNN models that can classify Alzheimer's and FTD diseases with fewer training samples. 3D MRI data is the essential requirement for conducting this study. The decision of choosing this 3D MRI scan data is that the models can learn the brain regions which are affected by these diseases. These affected brain regions are the region of interest (ROI) in classifying these diseases. Finally, for training 3D CNN models that data was acquired from the Alzheimer's Disease Neuroimaging Initiative (ADNI) Database (https://adni.loni.usc.edu/). For the downstream task of FSL, we obtained the data samples of FTD from The Frontotemporal Labor Degeneration Neuroimaging Initiative (https://ida.loni.usc.edu/collaboration/access/appLicense.jsp) Database (NIFTD). In this thesis, we obtained only MRI modality to classify the FTD disease.

5.2 3D MRI Scans for Alzheimer's and Frontotemporal Dementia Disease

In this thesis, the 3D data MRI scans were collected from publicly available datasets from Alzheimer's Disease Neuroimaging Initiative (ADNI) and The Frontotemporal Neuroimaging Initiative (NIFTD). The primary goal of ADNI and NIFTD publicly available databases is to study clinical, imaging, biochemical biomarkers, and genetics that would help us to detect early and the progression of AD and FTD diseases. Due to MRI scans being collected from different scanner platforms, standard pre-processing steps have been used for the ADNI dataset. ADNI-GO/-2 MRI scans were collected from different 3T MRI scanners using scanner-specific T1 3D MPRAGE protocols [71]. In the initial step of preprocessing the 3D MRI scans are automatically segmented into gray matter, white matter, and cerebrospinal fluid partitions of 1.5mm isotropic voxel size by using the segmentation toolbox of VMB8. These gray matter and white matter using the DARTEL algorithm are registered high-dimensionally to an aging specific reference template [71]. To save the total amount of gray matter (GM) before wrapping, final reference template is used to wrap the gray matter segments, and voxel values were modulated for the volumetric changes introduced by the normalization [71]. Lastly, the segmentation and registration accuracy are asserted by the visual inspection of gray matter maps [71]. Fig: 6.1 visualizes the basic preprocessing steps used for preprocessing of ADNI dataset. The Frontotemporal Neuroimaging Initiative (NIFTD) MRI scan is also available in the ADNI database, which also undergoes the same

pre-processing steps before it's being available for public use, Both ADNI and NIFTD datasets have been used in this thesis.

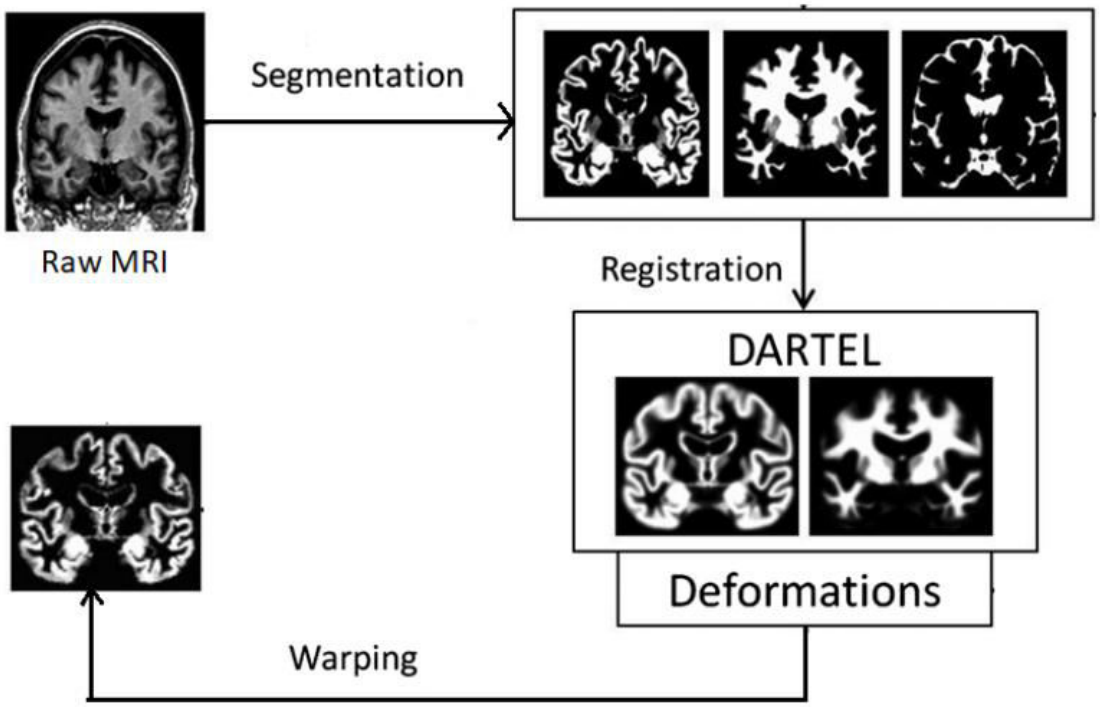

FIGURE 5. 1 BASIC PRE-PROCESSING STEPS HAVE BEEN USED IN ADNI MRI SCANS [71]

ADNI dataset contains 662 MRI scans, which are divided into different categories namely Normal cognitive (NC), late mild cognitive impairment (LMCI), and Alzheimer's disease (AD). Similarly, the NIFTD dataset contains a total of 279 MRI scans which are categorized into Normal cognitive (NC) and frontotemporal dementia (FTD). For training the Alzheimer's disease model, we used a total of 530 MRI scans. For the validation of the trained model, we used 66 data samples while a test set of 67 MRI scans have been used for evaluation of the model. Moreover, in the downstream task of FSL, a smaller number of FTD MRI scans have been utilized for the classification of FTD disease.

ADNI MRI Scan(n=662)	**CN**	**AD**	**LMCI**
662	254	189	220

NIFTD MRI Scans (n=279)	**CN**	**FTD**
279	133	146

TABLE :5. 1 MRI SCANS OF ADNI AND NIFTD DATASETS

Each 3D MRI scan of ADNI and NIFTD databases is a 3D volume of gray scale dimensions of 121 x 145 x 121, where 145 is the number of brain coronal slices of dimension 121 x 121. Figure: 6.3 depicts the MRI scan of AD from the ADNI dataset.

FIGURE 5. 2 3D AD MRI SCAN OF ADNI DATASET

5.3 Preprocessing

The 3D MRI scans used for this work have been preprocessed using the computational Anatomy toolbox (CAT12) and the SPM12 tool has been used for statistical mapping. Segmented gray and white matter images are normalized to the default CAT12 brain template in Montreal Neuroimaging Institute (MNI) reference space using the DARTEL algorithm. This is then resliced into isotropic voxel size of 1.5mm, while expansion and shrinkage of the tissues are adjusted through modulation [72]. The pre-processed image is being used to develop the models and is being transformed for the model inputs.

5.4 3D CNN Models

The primary task of this thesis is to develop 3D CNN models for the multi-classification of AD/LMCI/CN using the ADNI dataset and the features of the best 3D CNN model are transferrable for embedding the FTD features, which classify the FTD disease with fewer training samples. The three 3D CNN models have been developed in this work for classifying the AD and one finalized model feature would be used for embedding learning. All three CNNs have been developed using a pre-trained model of Video ResNet-18 from Pytorch. The three ADNI models are being developed to generate the prior knowledge for the downstream FSL task to classify FTD disease with fewer training samples. The three models are being developed to acquire the optimal classification performance and are also tracked for the classification performance in each network setting.

5.4.1 ADNI CNN Model 1

The first model is being developed without using the pre-trained weights and we use only the pre-trained ResNet-18 architecture. The 3D Video ResNet-18 pre-trained model is composed of four convolution layers and each layer is followed by batch normalization, ReLU activation, and maximum pooling layer. All four convolution layers are connected to the last fully connected layer, which is responsible for the classification. A total of 622 data samples have been used out of which 540 are training samples, 66 validation samples, and 67 samples are reserved for testing the trained model. The model hyperparameters which were used during the training of the model are epochs: 10, Batch size: 10, learning rate: 0.001, optimizer: SGD, and loss function: cross-entropy. Figure 5.3 depicts the architecture of the 3D CNN model without using pre-trained weights.

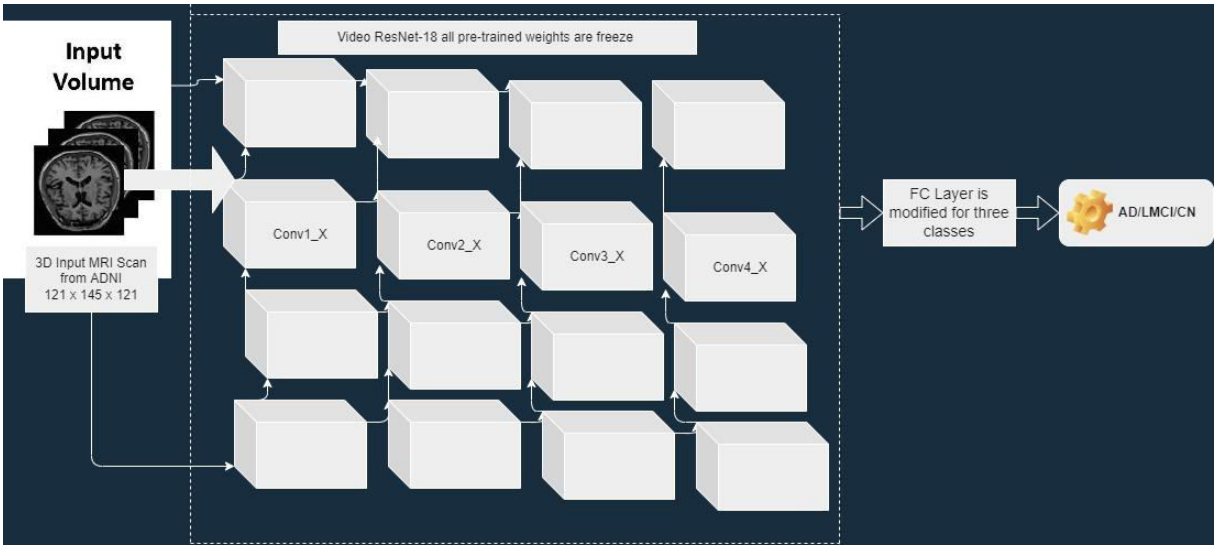

FIGURE 5. 3 ARCHITECTURE OF 3D CNN WITHOUT PRE-TRAINED WEIGHTS

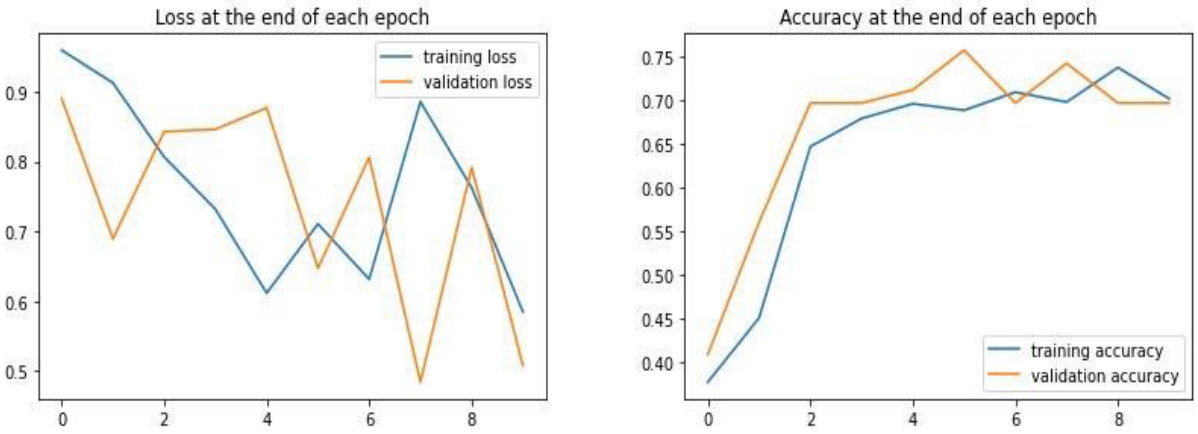

FIGURE 5. 4 LOSS AND ACCURACY CURVES OF THE TRAINED 3D CNN MODEL WITHOUT PRE-TRAINED WEIGHTS

5.4.2 ADNI Model 2

The second 3D CNN model has been developed using the pre-trained weights of the video ResNet-18 model. In this 3D CNN model, the feature extraction method (explained in section 3.7) of transfer learning has been used in which only the classification layer weights are updated and the convolution base weights were frozen. A total of 622 data samples have been used out of which 540 training samples, 66 validation samples, and 67 samples are reserved for testing the trained model. The model hyperparameters which was used during the training of this model are epochs: 10, Batch size: 10, learning rate: 0.001, optimizer: SGD, and loss function: cross-entropy. Figure 5.5 depicts the architecture of 3D CNN mode using pre-trained weights.

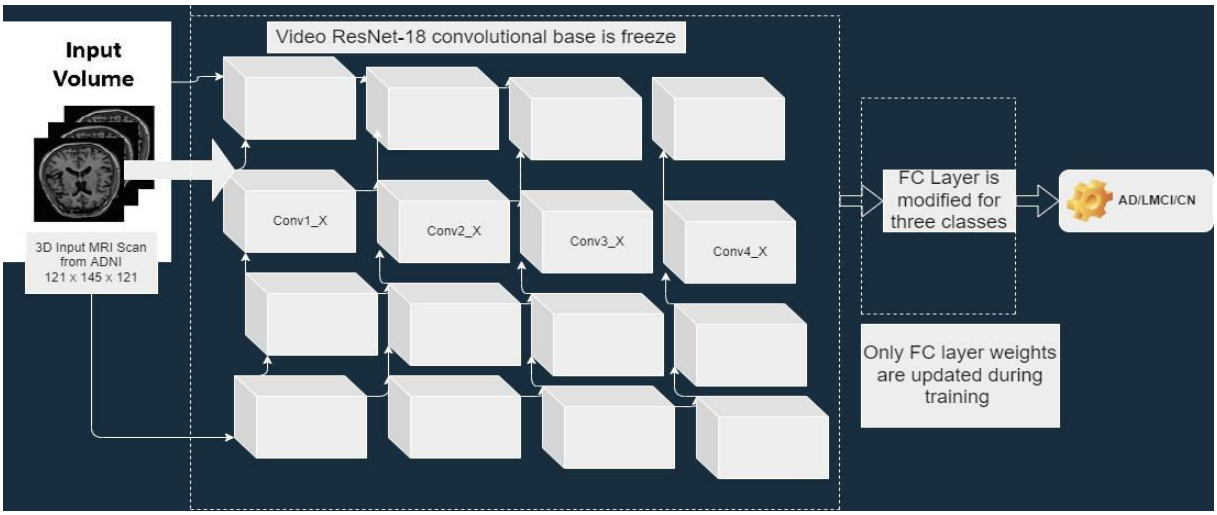

FIGURE 5. 5 ARCHITECTURE OF 3D CNN FEATURE EXTRACTION MODEL USING PRE-TRAINED WEIGHTS OF RESNET-18

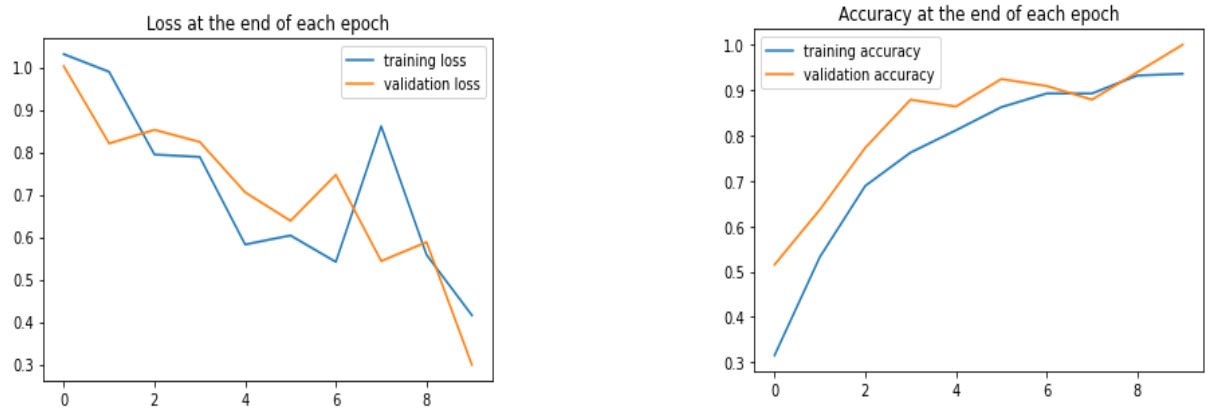

FIGURE 5. 6 LOSS AND ACCURACY CURVES OF THE TRAINED 3D CNN MODEL WITH PRE-TRAINED WEIGHTS OF RESNET-18 MODEL

5.4.3 ADNI Model 3

The third 3D CNN model has been developed using the pre-trained weights of the video ResNet-18 model. In this 3D CNN model, fine-tuning method (discussed in 3.7) of transfer learning has been used in which the second last layer and the fully connected layer of the Video ResNet-18 model weights are updated and the remaining convolutional layer weights were frozen. A total of 622 data samples have been used out of which 540 training samples, 66 validation samples, and 67 samples are reserved for testing the trained model. The model hyperparameters which was used during the training of this model are epochs: 10, Batch size: 10, learning rate: 0.0001, optimizer: Adam, and loss function: cross-entropy. Figure 5.8 depicts the architecture of the 3D fine-tune CNN model using the pre-trained weights of ResNet-18.

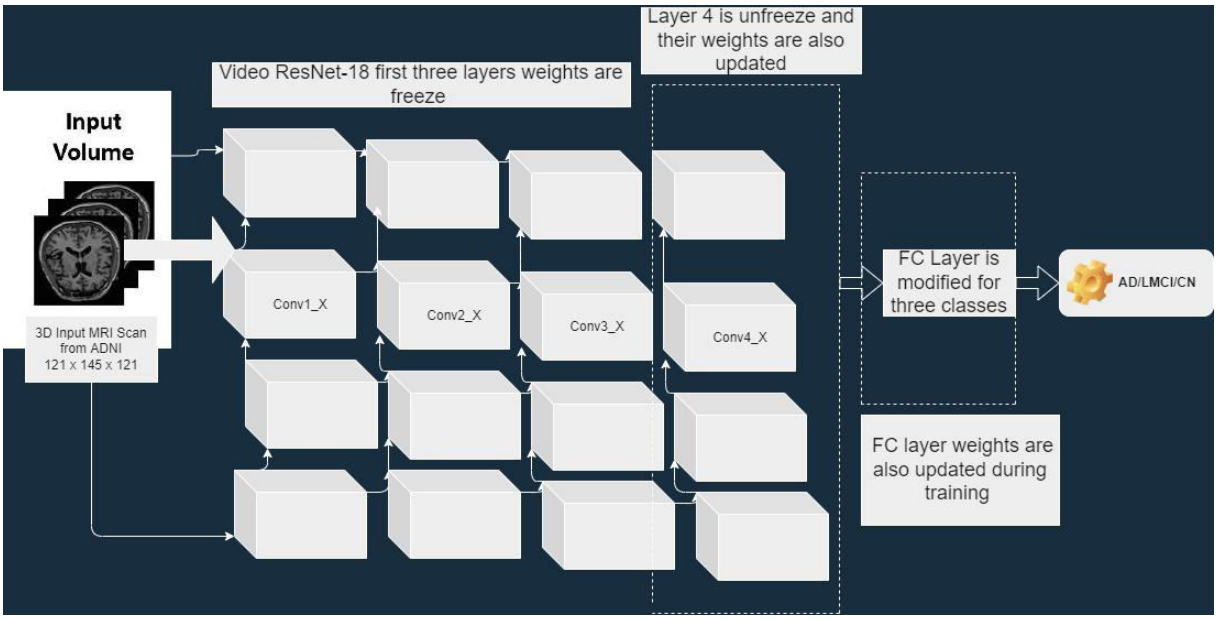

FIGURE 5. 7 ARCHITECTURE OF 3D CNN FINE-TUNE MODEL USING PRE-TRAINED WEIGHTS OF RESNET-18

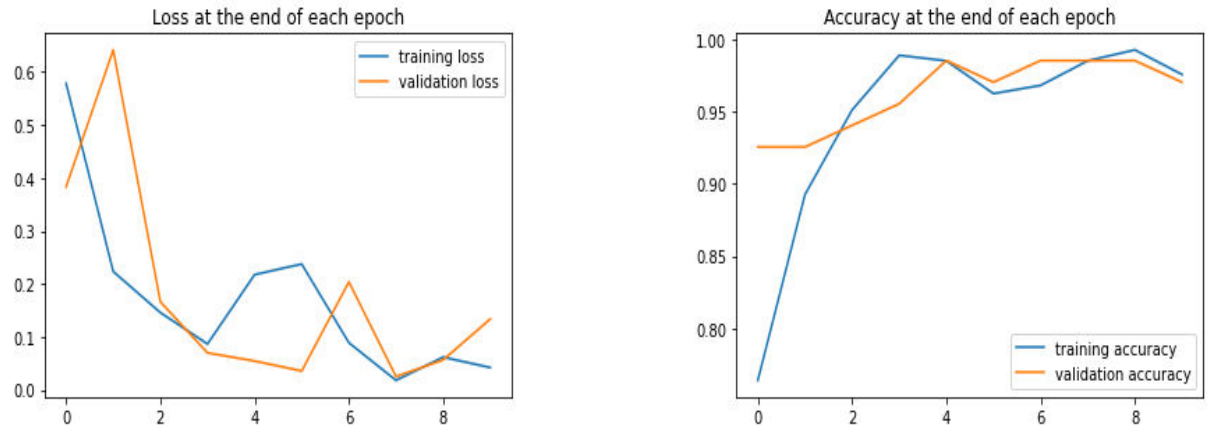

FIGURE 5. 8 LOSS AND ACCURACY CURVES OF THE TRAINED 3D CNN MODEL WITH PRE-TRAINED WEIGHTS

This ADNI3 fine-tune model features will be used in the feature embedding process for the classification of FTD disease. This ADNI3 fine-tune model achieves good classification results, which helps us in classifying the FTD samples with fewer training samples. This model provides a robust transferrable feature that aids in the downstream FTD disease classification. The proposed methodology (discussed in chapter 4) that FSL requires optimal prior knowledge for the meta-testing or downstream task. In this work, ADNI fine-tune model 3 provides the optimal prior knowledge.

5.5 Features Embedding

In this step, we need to embed the features of the ADNI fine-tune model with the NIFTD data samples to perform the task of frontotemporal dementia classification. Then extract the learned features of the ADNI fine-tune model at the average pooling layer stage and don't consider the fully-connected layer, which is responsible for the classification. After embedding the ADNI features with the NIFTD data samples, we trained the logistic regression model for the classification of FTD disease. We trained the logistic regression model with the support set samples, which contained embedding features of NIFTD and ADNI. Likewise, the logistic model is tested on the query set, which contains the embedding features of ADNI and NIFTD data samples. Figure: 5.9 depicts the overall process and the classification of FTD disease by using features embedding.

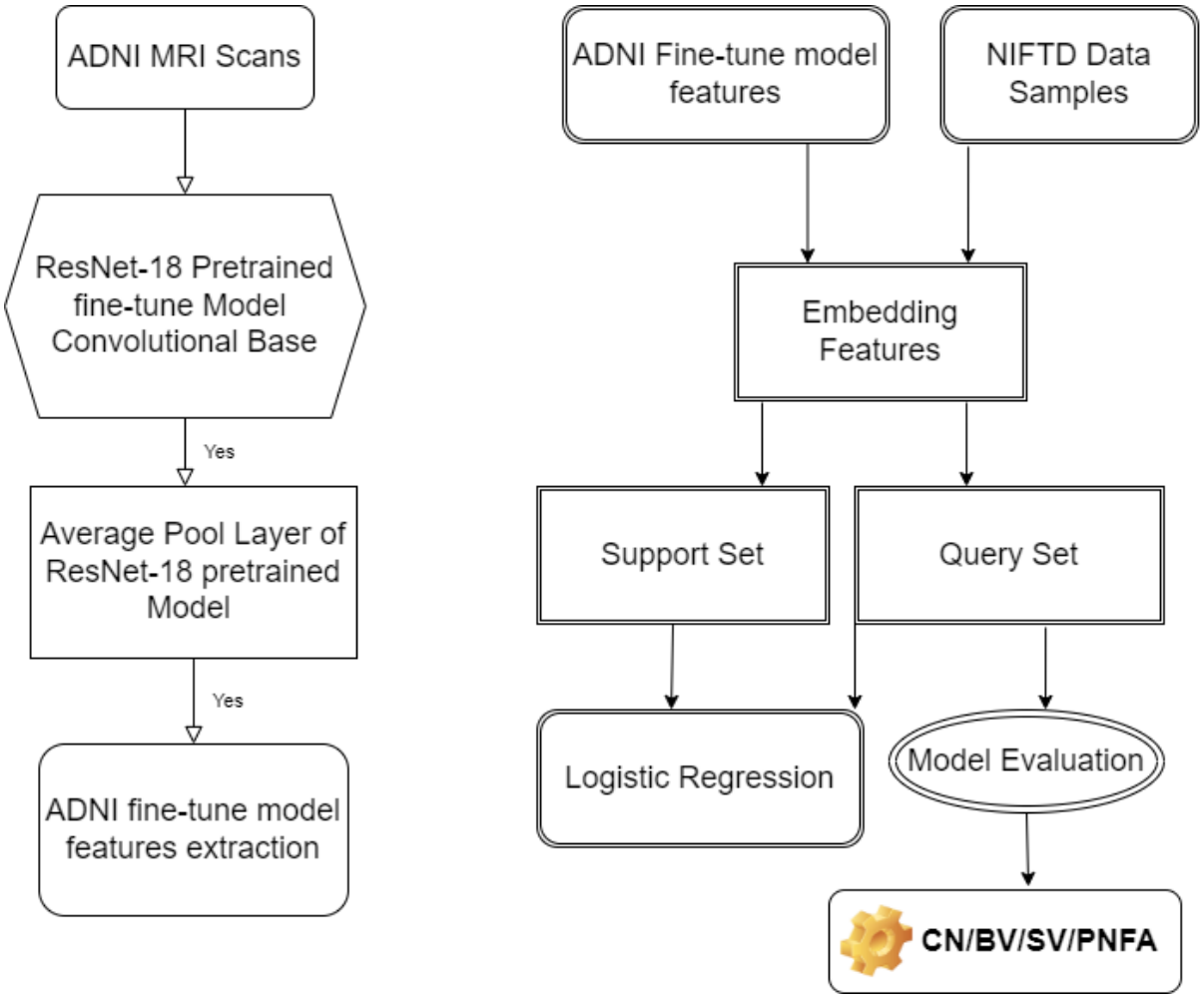

FIGURE 5. 9 OVERALL PROCESS OF FEATURE EMBEDDING OF ADNI FEATURES AND NIFTD DATA SAMPLES

5.6 FSL Experimentations

We perform three FSL experiments to perform multi-classification of FTD tasks using fewer training samples. We arrange the training set with 4 classes of data samples from the NIFTD dataset to train the logistic regression, which is known as the Support set. The test set also contains 4 classes of samples from the NIFTD dataset to evaluate the logistic regression, which is known as the Query set. In this thesis, three experimental settings have been performed to classify the FTD disease and explore the significant performance difference between each experiment setting. The conducted experiments for FSL are baseline binary classification (CN vs FTD), 4-shot 5-way multi-classification, and 4-shot 10-way multi-classification. In FSL, "N-shot" is defined as the number of classes in a dataset and the 'K-way' is described as the number of samples per class. In the following sections, each experiment is defined briefly.

5.6.1 Baseline Multi-classification Model

In the baseline model setting, the trained model is needed to perform the binary classification task (CN vs FTD). We arrange the training set with 2 classes (CN vs FTD) of data samples from the NIFTD dataset to train the logistic regression, which is known as the Support set. The test set also contains 2 classes of samples from the NIFTD dataset to evaluate the logistic regression, which is known as the Query set. In this task, the logistic regression model is trained with the embedded features of ADNI and FTD data samples. The training or support set to train logistic regression consists of 80% (223) NIFTD data samples and the held-out test set or query set consists of 20% (56) NIFTD data samples. The model runs for 1000 iterations, L2 regularization parameter is used to penalize the misclassified samples. The trained model is evaluated using metrics of precision, recall, f1-score, and the Sk-learn evaluation metrics known as accuracy score and phi-coefficient.

NIFTD Data Sample Classes	Precision	Recall	F1-Score	Accuracy score	Phi-coefficient
CN	0.90	0.96	0.93	0.92	0.85
FTD	0.96	0.90	0.93		

TABLE :5. 2 BASELINE BINARY CLASSIFICATION MODEL WITH SK-LEARN EVALUATION METRICS

5.6.2 5-Short 4-Way Multi-classification Model

In the 5-Shot (5 samples per class) 4-way (4 classes: CN, SV, BN, PNFA of NIFTD dataset) multi-classification model, the logistic regression model is trained with only 20 data samples of support set from the NIFTD dataset and the trained model is evaluated using 12 samples of Query set. In this task, the logistic regression model is trained with the embedded features of ADNI and FTD data samples. The model runs for 1000 iterations, L2 regularization parameter is used to penalize the misclassified samples. The trained model is evaluated using metrics of precision, recall, f1-score, and the gold standard evaluation technique known as cross-validation. In this task, the model classifies different variants of FTD disease using fewer training samples per class.

NIFTD Data Samples Classes	Precision	Recall	F1-Score	Accuracy score	Phi-coefficient
CN	0.75	1.00	0.86	0.66	0.56
BV	0.50	0.33	0.40		
SV	1.00	0.67	0.80		
PNFA	0.50	0.67	0.57		

TABLE :5. 3 5-SHOT 4-WAY MULTI-CLASSIFICATION MODEL EVALUATION WITH SK-LEARN EVALUATION METRICS

5.6.3 10-Shot 4-Way Muli-classification Model

In the 10-Shot (10 samples per class) 10-way (4 classes: CN, SV, BN, PNFA of NIFTD dataset) multi-classification model, the logistic regression model is trained with only 40 data samples of support set from the NIFTD dataset, and the trained model is evaluated using 12 samples of Query set. In this task, the logistic regression model is trained with the embedded features of ADNI and FTD data samples. The model runs for 1000 iterations, L2 regularization parameter is used to penalize the misclassified samples. The trained model is evaluated using metrics of precision, recall, f1-score, and the gold standard evaluation technique known as cross-validation. In this task, the model classifies different variants of FTD disease using fewer training samples per class.

NIFTD data sample Classes	Precision	Recall	F1-Score	Accuracy score	Phi-coefficient
CN	1.00	1.00	1.00	0.83	0.78
BV	0.67	0.67	0.67		
SV	1.00	0.67	0.80		
PNFA	0.75	1.00	0.86		

TABLE :5. 4 10-SHOT 4-WAY MULTI-CLASSIFICATION MODEL EVALUATION WITH SK-LEARN EVALUATION METRICS

5.7 Models Developing Environment and Tools

All the implementation has been done on Google Colab (https://colab.research.google.com/), which is a cloud bases environment to execute python codes. Google Colab provides 12 GB of RAM and 64 GB of storage free of charge. Colab provides the facility of free GPU access to fast computation and parallel processing of the code. The model development process utilized the deep learning framework of “Pytorch”. The necessary libraries utilized in this thesis such as NumPy for array computations, nibabel for reading/writing the MRI scans, pandas for arranging the data in data frames, Sk-learn library are being utilized for model evaluation, and seaborn and matplotlib are used for data visualization etc. All the illustrated libraries and tools used in this thesis are mentioned with their versions in the Appendix section A.

Chapter 6

Results and Discussion

In this chapter, the results of the experiments implemented in the previous chapter are presented and illustrated, by taking into account the goal of the thesis, which is to classify frontotemporal dementia using fewer training samples. Different experiment implementations (mentioned in Chapter 5) and methodologies (discussed in Chapter 4) have been adopted to accomplish this thesis goal. Initially, the chapter presents and discusses the ADNI model results, which are acquired by using transfer learning methodologies. Afterward, the chapter discusses the classification results of frontotemporal dementia using the few-shot learning methodology. Below discussed are the results of the implemented three ADNI models in chapter 5. In the later part of the chapter, we discuss the results of the three experimentations using a few-shot learning methodology.

6.1 ADNI Model 1

Chapter 5 discussed the implementation steps like the hyper-parameter settings and the model training part. The ADNI model 1 is being developed using the Video ResNet-18 pre-trained network architecture without the pre-trained weights. The model was evaluated on PyTorch evaluation functions using 67 test data samples of ADNI. The model is also evaluated using 3 fold cross-validation techniques for the consistency of results. ADNI model 1 is developed for the multi-classification task (AD/LMCI/CN).

ADNI test data samples	Test accuracy	Phi-coefficient
67	0.73	0.68

TABLE: 6. 1 ADNI MODEL WITHOUT PERTAINED WEIGHTS AND CROSS-VALIDATION EVALUATION

True Labels	CN	AD	LMCI
CN	25	2	0
AD	9	8	0
LMCI	0	0	23
Predicted labels			

FIGURE 6. 1 CONFUSION MATRIX VISUALIZATION OF ADNI MODEL WITHOUT PRE-TRAINED WEIGHTS

The results of the ADNI model 1 without using the pre-trained weight show that the model misclassifies the number of samples. The CN class misclassified the two samples and predicted them as AD. Whereas, the AD class also misclassified the nine samples and predicted them as CN class. In total, the model misclassified the eleven samples, which is accounted as a significant misclassification in our specific case. Here, we need to obtain the optimal classification performance of the model that helps us in classifying the FTD disease in our downstream task.

6.2 ADNI Model 2

The ADNI model 2 has been developed using the pre-trained weights of Video ResNet-18 and utilized the feature extraction method of transfer learning. The pre-trained model convolutional base weights are frozen and only the last fully connected layer weights are allowed to update. The model was evaluated on PyTorch evaluation functions using 67 test data samples of ADNI. The model is also evaluated using 3 fold cross-validation technique for the consistency of results. ADNI model 2 is developed for the multi-classification task (AD/LMCI/CN).

ADNI test data samples	Test accuracy	Phi-coefficient
67	0.83	0.73

TABLE: 6. 2 ADNI FEATURE EXTRACTION MODEL WITH PERTAINED WEIGHTS AND CROSS-VALIDATION EVALUATION

True Labels	CN	AD	LMCI
CN	26	1	0
AD	5	12	0
LMCI	3	2	18
Predicted labels			

FIGURE 6. 2 CONFUSION MATRIX VISUALIZATION OF ADNI FEATURE EXTRACTION MODEL WITH PRE-TRAINED WEIGHTS

The ADNI feature extraction model utilized the ResNet-18 pre-trained weights, which improves the classification performance of CN and AD classes. However, the classification is still sub-optimal because a total of eleven samples were misclassified. A sample of CN was misclassified and was predicted as AD, while, the five AD samples were misclassified and predicted as CN. The five LMCI samples were misclassified, in which three were predicted as CN and two were predicted as AD. The pre-trained model is more adaptable to CN and AD classes, but not to LMCI that is because only the last layer of the model is trained with the ADNI dataset. Thus, to do more fine-tuning layers of the pre-trained model will significantly improve the classification performance of each class respectively.

6.3 ADNI Model 3

The ADNI model 3 has been developed using the pre-trained weights of Video ResNet-18 and utilized the fine-tuning method of transfer learning. The pre-trained model convolutional layer 4 (second-last layer of the pre-trained model) and the last fully connected layer weights are allowed to update. The model was evaluated on PyTorch evaluation functions using 67 test data samples of ADNI. The model is also evaluated using 3 fold cross-validation technique for the consistency of results. ADNI model 3 is developed for the multi-classification task (AD/LMCI/CN).

ADNI test data samples	Test accuracy	Phi-coefficient
67	0.98	0.97

TABLE: 6. 3 ADNI FINE-TUNE MODEL WITH PERTAINED WEIGHTS AND CROSS-VALIDATION EVALUATION

True Labels	CN	AD	LMCI
CN	26	1	0
AD	0	17	0
LMCI	0	0	23
Predicted Labels			

FIGURE 6. 3 CONFUSION MATRIX VISUALIZATION OF ADNI FINE-TUNE MODEL WITH PRE-TRAINED WEIGHTS

The ADNI fine-tune model utilized the ResNet-18 pre-trained weights, which improves the classification performance of all classes in the ADNI dataset. A sample of CN was misclassified and predicted as AD. The fine-tuning of the pre-trained model is more adaptable to all classes of the dataset. Hence, the model achieves the desired classification performance. Therefore, the downstream task to classify FTD disease will utilize these optimal fine-tune model features to get good classification performance with fewer training samples.

6.4 Comparison of ADNI Models

We set three different ADNI model settings for the multi-classification task to track the model classification performance. All ADNI models provide improved classification performance. The ADNI model 1 provides a good classification performance without using the pre-trained weights of the ResNet-18 pre-trained model. It shows that the networks are learning meaningful features from the ADNI MRI scans dataset. In the ADNI model 2, which is developed using the pre-trained weights of the ResNet-18 model. The model results improve the classification performance compared to model 1, which shows that the pre-trained weights are optimizing the classification performance. Finally, the ADNI model 3, which is a fine-tuned version in which the last convolution layer is also trained and their weights are also updated. The model achieves the optimal classification performance. ADNI models were being developed to generate the prior knowledge to solve the downstream task to classify the FTD disease. All model results are mentioned in the table below:

ADNI Models	Balance Accuracy	3-fold cross-validation mean accuracy
ADNI Model 1	0.73	0.72
ADNI Model 2	0.83	0.82
ADNI Model 3	0.97	0.98

TABLE: 6. 4 RESULTS COMPARISONS OF ADNI MODELS WITH CROSS-VALIDATION GOLD STANDARD EVALUATION TECHNIQUE

6.5 Feature Embedding

To perform the downstream task, to classify the FTD disease with fewer training samples. Specifically, the downstream task is being performed using the embedding learning strategy of the few-shot learning methodology. After generating the prior knowledge for the downstream task using ADNI model development, we finalized that the fine-tuned model would be used as prior knowledge for the downstream task to classify the FTD disease. The features' quality is the most important factor to achieve good classification accuracy on downstream tasks. We embedded the NIFTD data samples with the features of fine-tuned ADNI model. In this approach, the feature size of the training set is increased with the embedding of ADNI fine-tune model features. Although, the support set contains very few training samples, the feature space of the training set is large enough to classify the FTD disease with fewer training samples. Likewise, the Query set also contains fewer data samples to evaluate the trained logistic regression model. Its feature size is also increased due to the embedding of the features of the ADNI fine-tune model, which was developed for the classification of Alzheimer's disease. This embedding learning methodology helps us to increase the feature space by utilizing the prior knowledge generated from ADNI fine-tune model, which helps us in predicting the downstream task of FTD disease with fewer training samples.

6.6 Baseline Experiment

This section of the chapter discusses the result acquired from the binary classification task (CN vs FTD) using the embedding learning methodology. This experiment's implementation steps and evaluation results have already been mentioned in the previous chapter 5. We found that from the table: 5.3 that the model achieves good classification performance. We present the additional results on different evaluation metrics and confusion matrix visualization.

NIFTD Query Set Samples	Accuracy score	Phi-coefficient
56	0.92	0.85

TABLE: 6. 5 BASELINE EXPERIMENT EVALUATION RESULTS USING EMBEDDING LEARNING METHODOLOGY

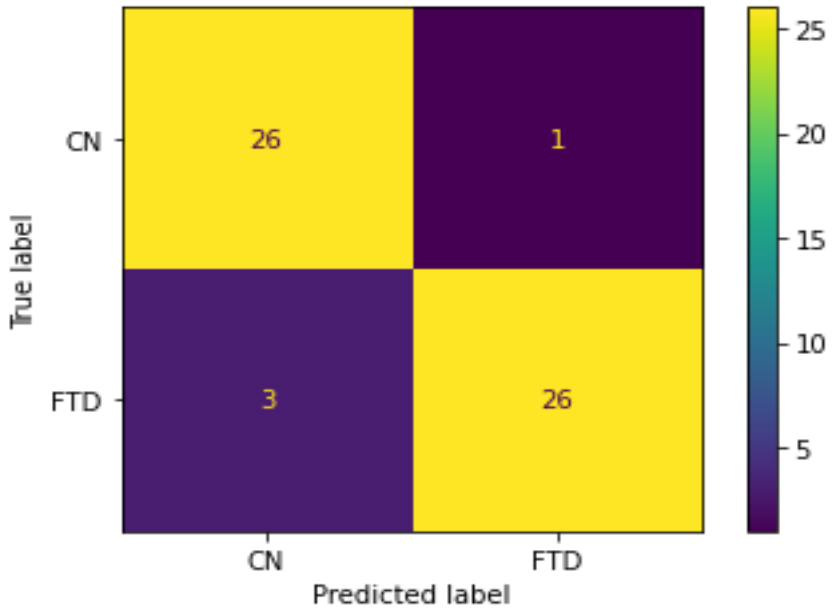

FIGURE 6. 4 CONFUSION MATRIX VISUALIZATION OF BASELINE MODEL USING EMBEDDING LEARNING METHODOLOGY

The baseline experiment achieved a good classification performance with only 4 misclassifications obtained. Although, the baseline experiment utilized all the NIFTD data samples to investigate the power of embedding learning. Afterward, we perform the experiments in the few-shot learning manner with only fewer training samples to train the logistic regression model by utilizing the embedding learning methodology.

6.7 5-Shot 4-Way Multi-classification Model

In this 5-short (5 samples per class) 4-way (CN, BV, SV, PNFA) experiment of FSL, the support set is consisted of 5 samples per class in a total of 20 training samples to train the logistic regression model and the query set consists of only 12 samples (3 samples per class) to evaluate the trained logistic regression model. This whole task setting is responsible to classify FTD disease with fewer training samples. Both support set and query set samples are embedded with prior knowledge of the ADNI fine-tune model. The implementation setting of this task has been explained in the previous chapter 5. We present some additional results on the different evaluation metrics and also in the confusion matrix format.

NIFTD Query Set Samples	Accuracy score	Phi-coefficient
12	0.66	0.56

TABLE: 6. 6 5-SHORT 4-WAY MULTI-CLASSIFICATION EVALUATION RESULTS USING EMBEDDING LEARNING METHODOLOGY

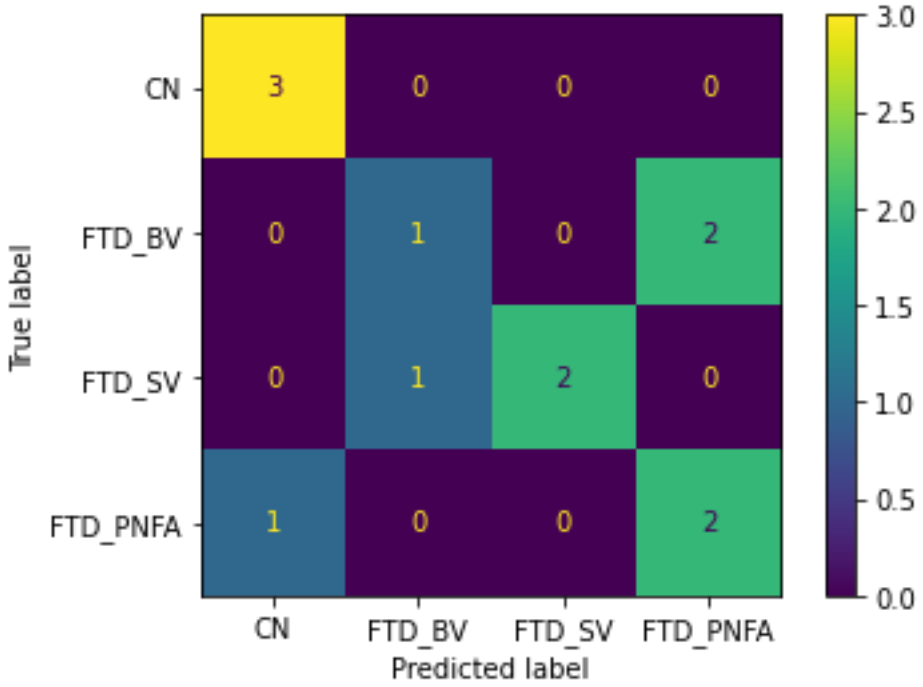

FIGURE 6. 5 CONFUSION MATRIX VISUALIZATION OF 5-SHOT 4-WAY MULTI-CLASSIFICATION MODEL USING EMBEDDING LEARNING METHODOLOGY

The 5-short 4-way multi-classification results show that the model misclassified the samples in BV, SV, and PNFA classes. The misclassification is obtained due to the limited training size of the support set, which was responsible to train the logistic regression model. Apart from misclassification, we achieved a significant classification performance of the model by only using 20 training samples. This significant achievement acquires through the embedding of ADNI fine-tune model features, which helped in classifying FTD data samples with fewer training samples.

6.8 10-Shot 4-Way Multi-Classification Model

In this 10-shot (10 samples per class) 4-way (CN, BV, SV, PNFA) multi-classification task of FSL, the support set consisted of 10 samples per class with a total of 40 samples responsible to train the logistic regression task and the query set consisted of only 12 samples (3 samples per class) which were responsible to evaluate the logistic regression model. This whole task setting was responsible to classify FTD disease with fewer training samples. Both support set and query set samples are embedded with prior knowledge of the ADNI fine-tuned model. The implementation setting of this task has been explained in the previous chapter 5. We present some additional results on the different evaluation metrics and also in the confusion matrix format.

NIFTD Query Set Samples	Accuracy score	Phi-coefficient
12	0.83	0.78

TABLE: 6. 7 10-SHORT 4-WAY MULTI-CLASSIFICATION EVALUATION RESULTS USING EMBEDDING LEARNING METHODOLOGY

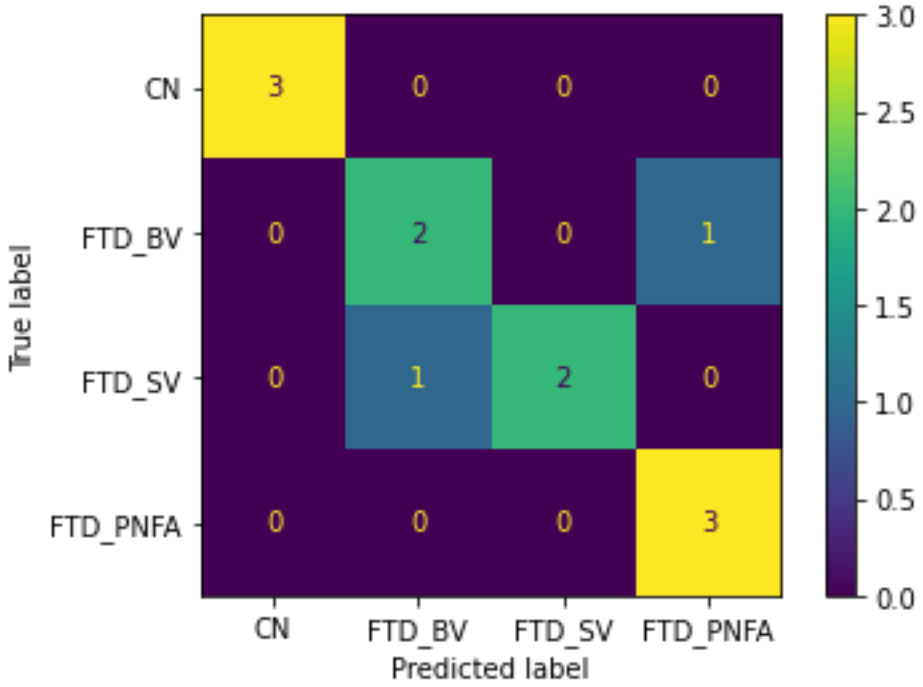

FIGURE 6. 6 CONFUSION MATRIX VISUALIZATION OF 10-SHOT 4-WAY MULTI-CLASSIFICATION MODEL USING EMBEDDING LEARNING METHODOLOGY

The 10-shot 4-way multi-classification results show that the model misclassified the samples in BV and SV classes. The misclassification is obtained due to the limited training size of the support set, which is responsible to train the logistic regression model. Apart from misclassification, the model achieves significant classification performance of the model with only using 40 training samples compared to 5-shot 4-way multi-classification. The findings of this experiment show that whenever we increased the training samples or shots, the model boosts its classification performance. However, the model achieves a good classification performance with fewer training samples and only two misclassifications were obtained. This significant achievement is acquired through the embedding of ADNI fine-tune model features as a prior knowledge, which helps the model to classify FTD data samples with fewer training samples.

6.9 Comparison of FSL Experiment Results

We set three different model settings for the multi-classification task to track the model classification performance. All experiments provide meaningful results to check the robustness of our methodology. The baseline experiment result shows us that our embedding learning methodology can have the power to classify the FTD disease when our dataset size is small. The 5-shot 4-way multi-classification task shows that with the embedding of ADNI fine-tune model prior knowledge with fewer training samples, the model can acquire suitable classification performance. Finally, a experiment of 10-shot 4-way classification result provides the optimal classification performance of FTD disease with fewer training samples.As increasing the shots, the model prediction performance improves which was proved from our obtained FSL experiment results. A comparison of all FSL experiment results is illustrated in the table below.

Experiments	5-fold cross-validation mean accuracy	Standard deviation of the model
Baseline Experiment (Binary Classification)	0.93	0.02
5-Shot 4-Way Multi-classification	0.60	0.19
10-Shot 4-Way Multi-classification	0.75	0.16

TABLE: 6. 8 COMPARISON OF ALL FSL EXPERIMENTS RESULTS USING EMBEDDING LEARNING METHODOLOGY

Chapter 7

7.1 Conclusion

This last chapter of thesis documentation presents the whole findings based on different exploration strategies to achieve the thesis goal, which was to classify frontotemporal dementia with fewer training samples. In the previous chapter 6, each implementation of the model is presented and their results have been discussed in detail. In this chapter, only the prominent results of our implementation are revised briefly.

Adopting the FSL methodology to address the problem of fewer training samples and also considering the limited computational power to process the 3D MRI scans, the transfer learning method is also being used to compensate for the large training time of the model and the limited training data size. To classify the FTD disease, a model-based embedding learning methodology has been finalized. As discussed in (section 1.4), prior knowledge needs to be generated to generalize new tasks with limited supervised information. In the proposed methodology of this thesis, the ADNI dataset is being used to generate the prior knowledge, which performs the downstream task to classify FTD disease with fewer training samples. The three ADNI models have been developed with the help of transfer learning methods. Each ADNI model produces some meaningful results. For instance, model 2 (section 6.3) shows that the adopted pre-trained network of ResNet-18 learns meaningful patterns of ADNI data samples with 10 epochs of model training and achieves the classification performance of ($82\% \pm 1$). After the fine-tuning of ADNI model 3 (section 6.4), more weights can be trained on ADNI data. It shows that a model with 10 epochs of training the model achieves the classification accuracy of ($97\% \pm 1$). The classification accuracy is very important in our case because to achieve the optimal classification of downstream tasks the prior knowledge should also be optimal. The feature extractor which was developed using the ADNI data is optimal, which helps in achieving our goal to classify the FTD disease with fewer training samples.

Generating optimal prior knowledge from the ADNI model 3 (Section 5.4.3) with the help of fine-tuning the pre-trained ResNet-18 model. The three experiments have been set to experience the power of the few-shot learning methodology. The baseline experimentation shows that by adopting the embedding learning methodology, the downstream task model can achieve 93% accuracy with only 223 training samples. The baseline experiment result shows that with the embedding learning strategy of the few-shot learning methodology the model can learn the meaningful features with fewer training samples. The 5-shot 4-way experimentation result provides us the promising results that the downstream task model can achieve the classification accuracy of 63% with only training with 20 samples. The 10-shot 4-way experimentation provides the outperforming results that the downstream task model achieves a 75% classification accuracy with only 40 training samples. Hence, with only fewer training samples the classification accuracy lies in the range of 63% to 75%. As it is evident in our expertimentation results, that increasing the shots the prediction performnace also increase. The phenomomon of FSL is easy to interpret. With increasing training samples or shots, the prediction performance of the model also improves. For instance, 3-shot is easier than 2-shot.

This work achieves the outperforming results with fewer training samples to classify the FTD disease. Thus, few-shot learning methodologies have the power to classify the FTD disease with fewer training samples, which is being proved by our obtained results. In the medical domain, where the acquisition of data is hard and complicated, but now this problem can be tackled through the few-shot learning methodologies. The results of this work have been cross-validated and it shows the proposed methodology is optimal and can be generalizable to other tasks, where data insufficiency problem exists.

7.2 Future Work

Although the present work achieves the overall good results, the limitation of this work is the insufficiency of data. The medical domain normally has limited training data, which could be handled through different methodologies. Even though our present work has observed valuable outcomes but there is still room for improvement. Adopting few-shot learning methodologies based on data would also enhance the classification performance of the downstream task. Another possible direction to learning feature maps with fewer training samples will provide us with the detailed interpretability of the learned model features. This study is heavenly dependent on the quality of 3D images, so the optimal pre-processing steps should be adopted to achieve the optimal results. However, the few-shot learning is the new or the emerging methodology that provides a different number of learning paradigms to resolve problems in the computer vision area such as image retrieval, and video event detection. Although the FSL methodology is new the improvement has been proposed which should be considered in the future work.

References

1. *Ahire JB*. The Artificial Neural Networks Handbook: Part 4, 2018, https://medium.com/@jayeshbahire/the-artificial-neural-networks-handbook-part-4-d2087d1f583e; Status: 16.12.2019
2. *Sze V, Chen Y-H, Yang T-J et al.* Efficient Processing of Deep Neural Networks: A Tutorial and Survey. Proc. IEEE 2017; 105: 2295 – 2329
3. Er.Parveen Kumar, Er.Pooja Sharma: Artificial Neural Networks-A Study.
4. Cs231n.github.io. 2022. CS231n Convolutional Neural Networks for Visual Recognition. [online] Available at: <https://cs231n.github.io/convolutional-networks/> [Accessed 17 April 2022].
5. *Lecun Y, Haffner P, Bottou L, Bengio Y*. Object Recognition with Gradient-Based Learning. In: Forsyth DA (ed). Shape, contour and grouping in computer vision. Berlin [etc.]: Springer, 1999: 319 – 345
6. *C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich*. Going deeper with Convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015: 1 – 9
7. Yamashita, Rikiya; Nishio, Mizuho; Do, Richard Kinh Gian; Togashi, Kaori (2018): Convolutional neural networks: an overview and application in radiology. In *Insights into imaging* 9 (4), pp. 611–629. DOI: 10.1007/s13244-018-0639-9.
8. Fukushima, K. (1980): Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. In *Biological cybernetics* 36 (4), pp. 193–202. DOI: 10.1007/BF00344251.
9. News.microsoft.com. 2022. [online] Available at: <https://news.microsoft.com/wp-content/uploads/prod/sites/93/2020/04/Student-Guide-Module-4-Deep-Learning-and-Neural-Networks.pdf> [Accessed 17 April 2022].
10. Albawi, Saad; Mohammed, Tareq Abed; Al-Zawi, Saad (82017): Understanding of a convolutional neural network. In: 2017 International Conference on Engineering and Technology (ICET). 2017 International Conference on Engineering and Technology (ICET). Antalya, 8/21/2017 - 8/23/2017: IEEE, pp. 1–6.
11. Albawi, Saad; Mohammed, Tareq Abed; Al-Zawi, Saad (82017): Understanding of a convolutional neural network. In: 2017 International Conference on Engineering and Technology (ICET). 2017 International Conference on Engineering and Technology (ICET). Antalya, 8/21/2017 - 8/23/2017: IEEE, pp. 1–6.
12. Ramachandran, Prajit; Zoph, Barret; Le V, Quoc (2017): Searching for Activation Functions. Available online at http://arxiv.org/pdf/1710.05941v2.
13. Nwankpa, Chigozie; Ijomah, Winifred; Gachagan, Anthony; Marshall, Stephen (2018): Activation Functions: Comparison of trends in Practice and Research for Deep Learning. Available online at http://arxiv.org/pdf/1811.03378v1.
14. Kingma, Diederik P.; Ba, Jimmy (2014): Adam: A Method for Stochastic Optimization. Available online at http://arxiv.org/pdf/1412.6980v9.
15. Ruder, Sebastian (2016): An overview of gradient descent optimization algorithms. Available online at http://arxiv.org/pdf/1609.04747v2.
16. Machine Learning From Scratch. 2022. Neural Networks: Feedforward and Backpropagation Explained. [online] Available at: <https://mlfromscratch.com/neural-networks-explained/#/> [Accessed 18 April 2022].

17. *Refaeilzadeh P, Tang L, Liu H*. Cross-Validation. In: LIU L, Özsu MT (eds). Encyclopedia of Database Systems. Boston, MA: Springer US, 2009: 532 – 538
18. scikit-learn. 2022. 3.1. Cross-validation: evaluating estimator performance. [online] Available at: <https://scikit-learn.org/stable/modules/cross_validation.html> [Accessed 19 April 2022].
19. James S. Bergstra; Rémi Bardenet; Yoshua Bengio; Balázs Kégl: Algorithms for Hyper-Parameter Optimization.
20. Tan, Chuanqi; Sun, Fuchun; Kong, Tao; Zhang, Wenchang; Yang, Chao; Liu, Chunfang (2018): A Survey on Deep Transfer Learning. Available online at http://arxiv.org/pdf/1808.01974v1.
21. Yadav, Samir S.; Jadhav, Shivajirao M. (2019): Deep convolutional neural network based medical image classification for disease diagnosis. In *J Big Data* 6 (1). DOI: 10.1186/s40537-019-0276-2.
22. Yosinski, Jason; Clune, Jeff; Bengio, Yoshua; Lipson, Hod: How transferable are features in deep neural networks? In Advances in Neural Information Processing Systems 27. Available online at http://arxiv.org/pdf/1411.1792v1.
23. *Verma S*. Understanding 1D and 3D Convolution Neural Network | Keras, 2019, https://towardsdatascience.com/understanding-1d-and-3d-convolution-neural-network-keras-9d8f76e29610; Status: 20.12.2019
24. Ji, Shuiwang; Yang, Ming; Yu, Kai (2013): 3D convolutional neural networks for human action recognition. In IEEE transactions on pattern analysis and machine intelligence 35 (1), pp. 221–231. DOI: 10.1109/TPAMI.2012.59.
25. Machine Learning From Scratch. 2022. Neural Networks: Feedforward and Backpropagation Explained. [online] Available at: <https://mlfromscratch.com/neural-networks-explained/#/> [Accessed 19 April 2022].
26. M, Hossin; M.N, Sulaiman (2015): A Review on Evaluation Metrics for Data Classification Evaluations. In *IJDKP* 5 (2), pp. 1–11. DOI: 10.5121/ijdkp.2015.5201.
27. scikit-learn. 2022. 3.3. Metrics and scoring: quantifying the quality of predictions. [online] Available at: <https://scikit-learn.org/stable/modules/model_evaluation.html> [Accessed 23 April 2022].
28. Szegedy, Christian; Liu, Wei; Jia, Yangqing; Sermanet, Pierre; Reed, Scott; Anguelov, Dragomir et al. (2014): Going Deeper with Convolutions. Available online at http://arxiv.org/pdf/1409.4842v1.
29. He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian (2015): Deep Residual Learning for Image Recognition. Available online at http://arxiv.org/pdf/1512.03385v1.
30. Simonyan, Karen; Zisserman, Andrew (2014): Very Deep Convolutional Networks for Large-Scale Image Recognition. Available online at http://arxiv.org/pdf/1409.1556v6.
31. Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E. (2017): ImageNet classification with deep convolutional neural networks. In *Commun. ACM* 60 (6), pp. 84–90. DOI: 10.1145/3065386.
32. Russakovsky, Olga; Deng, Jia; Su, Hao; Krause, Jonathan; Satheesh, Sanjeev; Ma, Sean et al. (2015): ImageNet Large Scale Visual Recognition Challenge. In *Int J Comput Vis* 115 (3), pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

33. Christopoulos, Georgios; Graff-Radford, Jonathan; Lopez, Camden L.; Yao, Xiaoxi; Attia, Zachi I.; Rabinstein, Alejandro A. et al. (2020): Artificial Intelligence-Electrocardiography to Predict Incident Atrial Fibrillation: A Population-Based Study. In *Circulation. Arrhythmia and electrophysiology* 13 (12), e009355. DOI: 10.1161/CIRCEP.120.009355.

34. Kang, Guixia; Liu, Kui; Hou, Beibei; Zhang, Ningbo (2017): 3D multi-view convolutional neural networks for lung nodule classification. In *PloS one* 12 (11), e0188290. DOI: 10.1371/journal.pone.0188290.

35. Ardila, Diego; Kiraly, Atilla P.; Bharadwaj, Sujeeth; Choi, Bokyung; Reicher, Joshua J.; Peng, Lily et al. (2019): End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. In *Nature medicine* 25 (6), pp. 954–961. DOI: 10.1038/s41591-019-0447-x.

36. Smith-Bindman, Rebecca; Kwan, Marilyn L.; Marlow, Emily C.; Theis, Mary Kay; Bolch, Wesley; Cheng, Stephanie Y. et al. (2019): Trends in Use of Medical Imaging in US Health Care Systems and in Ontario, Canada, 2000-2016. In *JAMA* 322 (9), pp. 843–856. DOI: 10.1001/jama.2019.11456.

37. Dong, Hao; Yang, Guang; Liu, Fangde; Mo, Yuanhan; Guo, Yike (2017): Automatic Brain Tumor Detection and Segmentation Using U-Net Based Fully Convolutional Networks. Available online at http://arxiv.org/pdf/1705.03820v3.

38. Lian, Chunfeng; Liu, Mingxia; Zhang, Jun; Shen, Dinggang (2020): Hierarchical Fully Convolutional Network for Joint Atrophy Localization and Alzheimer's Disease Diagnosis Using Structural MRI. In *IEEE transactions on pattern analysis and machine intelligence* 42 (4), pp. 880–893. DOI: 10.1109/TPAMI.2018.2889096.

39. Suk, Heung-Il; Lee, Seong-Whan; Shen, Dinggang (2016): Deep sparse multi-task learning for feature selection in Alzheimer's disease diagnosis. In *Brain structure & function* 221 (5), pp. 2569–2587. DOI: 10.1007/s00429-015-1059-y.

40. Lin, Weiming; Tong, Tong; Gao, Qinquan; Di Guo; Du, Xiaofeng; Yang, Yonggui et al. (2018): Convolutional Neural Networks-Based MRI Image Analysis for the Alzheimer's Disease Prediction From Mild Cognitive Impairment. In Frontiers in neuroscience 12, p. 777. DOI: 10.3389/fnins.2018.00777.

41. Meyer, Sebastian; Mueller, Karsten; Stuke, Katharina; Bisenius, Sandrine; Diehl-Schmid, Janine; Jessen, Frank et al. (2017): Predicting behavioral variant frontotemporal dementia with pattern classification in multi-center structural MRI data. In *NeuroImage. Clinical* 14, pp. 656–662. DOI: 10.1016/j.nicl.2017.02.001.

42. Mathkunti, Nivedita Manohar; Rangaswamy, Shanta (2020): Machine Learning Techniques to Identify Dementia. In SN COMPUT. SCI. 1 (3). DOI: 10.1007/s42979-020-0099-4.

43. Meyer, Sebastian; Mueller, Karsten; Stuke, Katharina; Bisenius, Sandrine; Diehl-Schmid, Janine; Jessen, Frank et al. (2017): Predicting behavioral variant frontotemporal dementia with pattern classification in multi-center structural MRI data. In *NeuroImage. Clinical* 14, pp. 656–662. DOI: 10.1016/j.nicl.2017.02.001.

44. National Institute on Aging. 2022. Alzheimer's Disease Fact Sheet. [online] Available at: <https://www.nia.nih.gov/health/alzheimers-disease-fact-sheet> [Accessed 25 April 2022].

45. Wen, Junhao; Thibeau-Sutre, Elina; Diaz-Melo, Mauricio; Samper-González, Jorge; Routier, Alexandre; Bottani, Simona et al. (2020): Convolutional neural networks for classification of Alzheimer's disease: Overview and reproducible evaluation. In *Medical image analysis* 63, p. 101694. DOI: 10.1016/j.media.2020.101694.

46. Da Silva, Iago Richard Rodrigues; Silva, Gabriela dos Santos Lucas e.; Souza, Rodrigo Gomes de; Santana, Maíra Araújo de; Da Silva, Washington Wagner Azevedo; Lima, Manoel Eusébio de et al. (2020): Deep learning for early diagnosis of Alzheimer's disease: a contribution and a brief review. In : Deep Learning for Data Analytics: Elsevier, pp. 63–78.

47. Sethi, Monika; Ahuja, Sachin; Rani, Shalli; Koundal, Deepika; Zaguia, Atef; Enbeyle, Wegayehu (2022): An Exploration: Alzheimer's Disease Classification Based on Convolutional Neural Network. In *BioMed research international* 2022, p. 8739960. DOI: 10.1155/2022/8739960.

48. Zhang, Daoqiang; Wang, Yaping; Zhou, Luping; Yuan, Hong; Shen, Dinggang (2011): Multimodal classification of Alzheimer's disease and mild cognitive impairment. In *NeuroImage* 55 (3), pp. 856–867. DOI: 10.1016/j.neuroimage.2011.01.008.
49. Suk, Heung-Il; Lee, Seong-Whan; Shen, Dinggang (2014): Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis. In NeuroImage 101, pp. 569–582. DOI: 10.1016/j.neuroimage.2014.06.077.

50. Whitwell, Jennifer L.; Josephs, Keith A. (2012): Recent advances in the imaging of frontotemporal dementia. In Current Neurology and Neuroscience Reports 12 (6), pp. 715–723. DOI: 10.1007/s11910-012-0317-0.

51. Bruun, Marie; Koikkalainen, Juha; Rhodius-Meester, Hanneke F. M.; Baroni, Marta; Le Gjerum; van Gils, Mark et al. (2019): Detecting frontotemporal dementia syndromes using MRI biomarkers. In NeuroImage. Clinical 22, p. 101711. DOI: 10.1016/j.nicl.2019.101711.

52. Alkabawi, Elham M.; Hilal, Allaa R.; Basir, Otman A. (52017): Computer-aided classification of multi-types of dementia via convolutional neural networks. In : 2017 IEEE International Symposium on Medical Measurements and Applications (MeMeA). 2017 IEEE International Symposium on Medical Measurements and Applications (MeMeA). Rochester, MN, USA, 5/7/2017 - 5/10/2017: IEEE, pp. 45–50.
53. Ahmed, Md Rishad; Zhang, Yuan; Feng, Zhiquan; Lo, Benny; Inan, Omer T.; Liao, Hongen (2019): Neuroimaging and Machine Learning for Dementia Diagnosis: Recent Advancements and Future Prospects. In IEEE reviews in biomedical engineering 12, pp. 19–33. DOI: 10.1109/RBME.2018.2886237.

54. Kermany, Daniel S.; Goldbaum, Michael; Cai, Wenjia; Valentim, Carolina C. S.; Liang, Huiying; Baxter, Sally L. et al. (2018): Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. In *Cell* 172 (5), 1122-1131.e9. DOI: 10.1016/j.cell.2018.02.010.

55. Yadav, Samir S.; Jadhav, Shivajirao M. (2019): Deep convolutional neural network based medical image classification for disease diagnosis. In *J Big Data* 6 (1). DOI: 10.1186/s40537-019-0276-2.

56. Du Tran; Wang, Heng; Torresani, Lorenzo; Ray, Jamie; LeCun, Yann; Paluri, Manohar (2017): A Closer Look at Spatiotemporal Convolutions for Action Recognition. Available online at http://arxiv.org/pdf/1711.11248v3.
57. Petersen RC, Aisen PS, Beckett LA et al. Alzheimer's Disease Neuroimaging Initiative (ADNI): clinical characterization. Neurology 2010; 74: 201 – 209

58. Yosinski, Jason; Clune, Jeff; Bengio, Yoshua; Lipson, Hod: How transferable are features in deep neural networks? In Advances in Neural Information Processing Systems 27. Available online at http://arxiv.org/pdf/1411.1792v1.

59. Kuperman V. Magnetic resonance imaging: Physical principles and applications. San Diego: Academic Press, 2000

60. Diba, Ali; Fayyaz, Mohsen; Sharma, Vivek; Arzani, M. Mahdi; Yousefzadeh, Rahman; Gall, Juergen; van Gool, Luc (2018): Spatio-Temporal Channel Correlation Networks for Action Classification. Available online at http://arxiv.org/pdf/1806.07754v3.

61. M. Ghazal, Taher; Abbas, Sagheer; Munir, Sundus; A. Khan, M.; Ahmad, Munir; F. Issa, Ghassan et al. (2022): Alzheimer Disease Detection Empowered with Transfer Learning. In *Computers, Materials & Continua* 70 (3), pp. 5005–5019. DOI: 10.32604/cmc.2022.020866.

62. Li, Yi; Haber, Annat; Preuss, Christoph; John, Cai; Uyar, Asli; Yang, Hongtian Stanley et al. (2021): Transfer learning-trained convolutional neural networks identify novel MRI biomarkers of Alzheimer's disease progression. In *Alzheimer's & dementia (Amsterdam, Netherlands)* 13 (1), e12140. DOI: 10.1002/dad2.12140.

63. Hu, Jingjing; Qing, Zhao; Liu, Renyuan; Zhang, Xin; Lv, Pin; Wang, Maoxue et al. (2020): Deep Learning-Based Classification and Voxel-Based Visualization of Frontotemporal Dementia and Alzheimer's Disease. In *Frontiers in neuroscience* 14, p. 626154. DOI: 10.3389/fnins.2020.626154.

64. Luo, Suhuai; Li, Xuechen; Li, Jiaming (2017): Automatic Alzheimer's Disease Recognition from MRI Data Using Deep Learning Method. In *JAMP* 05 (09), pp. 1892–1898. DOI: 10.4236/jamp.2017.59159.

65. Hon, Marcia; Khan, Naimul (2017): Towards Alzheimer's disease Classification through Transfer Learning. Available online at http://arxiv.org/pdf/1711.11117v1.

66. Bontonou, Myriam; Lioi, Giulia; Farrugia, Nicolas; Gripon, Vincent (2020): Few-shot Decoding of Brain Activation Maps. Available online at http://arxiv.org/pdf/2010.12500v3.

67. Tang, Hao; Liu, Xingwei; Sun, Shanlin; Yan, Xiangyi; Xie, Xiaohui (102021): Recurrent Mask Refinement for Few-Shot Medical Image Segmentation. In : 2021 IEEE/CVF International Conference on Computer Vision (ICCV). 2021 IEEE/CVF International Conference on Computer Vision (ICCV). Montreal, QC, Canada, 10/10/2021 - 10/17/2021: IEEE, pp. 3898–3908.

68. Tian, Yonglong; Wang, Yue; Krishnan, Dilip; Tenenbaum, Joshua B.; Isola, Phillip (2020): Rethinking Few-Shot Image Classification: a Good Embedding Is All You Need? Available online at http://arxiv.org/pdf/2003.11539v2.

69. Wang, Yaqing; Yao, Quanming; Kwok, James T.; Ni, Lionel M. (2021): Generalizing from a Few Examples. In ACM Comput. Surv. 53 (3), pp. 1–34. DOI: 10.1145/3386252.

70. Murphy, Kevin P. (2012): Machine learning. A probabilistic perspective. Cambridge MA: MIT Press (Adaptive computation and machine learning series).

71. Grothe, Michel J.; Heinsen, Helmut; Amaro, Edson; Grinberg, Lea T.; Teipel, Stefan J. (2016): Cognitive Correlates of Basal Forebrain Atrophy and Associated Cortical Hypometabolism in

Mild Cognitive Impairment. In *Cerebral cortex (New York, N.Y. : 1991)* 26 (6), pp. 2411–2426. DOI: 10.1093/cercor/bhv062.

72. Dyrba, Martin; Hanzig, Moritz; Altenstein, Slawek; Bader, Sebastian; Ballarini, Tommaso; Brosseron, Frederic et al. (2021): Improving 3D convolutional neural network comprehensibility via interactive visualization of relevance maps: evaluation in Alzheimer's disease. In *Alzheimer's research & therapy* 13 (1), p. 191. DOI: 10.1186/s13195-021-00924-2.
73. Boris Oreshkin; Pau Rodríguez López; Alexandre Lacoste: TADAM: Task dependent adaptive metric for improved few-shot learning.

74. Luca Bertinetto; João F. Henriques; Jack Valmadre; Philip Torr; Andrea Vedaldi: Learning feed-forward one-shot learners.

Appendix A.

Description of used Tools

This appendix asserts the modules and libraries that have been utilized for this work. Section 5.7 explained the training environment and the importance of these tools in this work. Below, we mentioned the used tool or modules that were so essential for this work.

Module Name	Version	Description	Module URL
Pytorch	1.9.1	Deep learning framework	https://pytorch.org/
Sk-learn	1.1.0	Model evaluation	https://scikit-learn.org/stable/
Nibabel	3.2.2	Read/write MRI scans	https://pypi.org/project/nibabel/
Matplotlib	3.5.2	Data visualization	https://matplotlib.org/
Seaborn	0.11.2	Data visualization	https://seaborn.pydata.org/
Numpy	1.22.0	Arrays computation	https://numpy.org/

TABLEA. 1 MOST USED MODULES USED FOR THE VARIOUS MODEL DEVELOPMENTS

Appendix B.

Additional Results

In this appendix, the additional results will be presented. In this work, one CNN model is developed using only two classes of ADNI datasets (CN and AD). The developed model is performing the task of binary classification. Secondly, we present the additional FSL experiment result which is termed as 5-Shot 3-Way multiclassification.

B.1 ADNI CNN Model:

The 3D CNN binary classification model has been developed using the pre-trained weights of the video ResNet-18 model. In this 3D CNN model, the feature extraction method (explained in section 3.7 Transfer Learning) of transfer learning has been used in which only the classification layer weights are updated and the convolution base weights were frozen. A total of 443 data samples have been used out of which 354 training samples, 44 validation samples, and 45 samples are reserved for testing the trained model. The model hyperparameters which was used during the training of this model are epochs: 5, Batch size: 10, learning rate: 0.001, optimizer: SGD, and loss function: cross-entropy. Below depicts the model evaluation results and the confusion matrix visualization of the trained CNN model.

ADNI Test data samples	Test accuracy	Balanced accuracy
45	0.57	0.578

TABLE B. 1 ADNI BINARY CLASSIFICATION MODEL EVALUATION RESULTS

True Labels	CN	AD
CN	16	5
AD	17	7
Predicted labels		

FIGURE B. 1 ADNI BINARY CLASSIFICATION MODEL CONFUSION MATRIX

The 3D CNN binary classification model doesn't achieve the good diagnostic accuracy of Alzheimer's disease due to very few training samples. A total of 22 misclassifications have been obtained in this CNN model. Afterward, we increase the dataset size by considering the LMCI class of the ADNI dataset for training the 3D CNN model to boost the diagnostic performance.

B.2 5-Shot 3-Way Multi-classification

In the 5-Shot (5 samples per class) 3-way (3 classes: CN, SV, BV of NIFTD dataset) multi-classification model, the logistic regression model is trained with only 15 data samples of support set from the NIFTD dataset, and the trained model is evaluated using 9 samples of Query set. In this task, the logistic regression model is trained with the embedded features of ADNI and FTD data samples. The model runs for 1000 iterations, L2 regularization parameter is used to penalize the misclassified samples. The trained model is evaluated using metrics of precision, recall, f1-score, and the gold standard evaluation technique known as cross-validation. In this task, the model classifies different variants of FTD disease using fewer training samples per class. Below depicts the evaluation results of the trained logistic regression model and the confusion matrix visualization.

NIFTD Data Samples Classes	Precision	Recall	F1-Score	5-fold Cross-validation mean accuracy	Standard deviation of the model
CN	0.75	1.00	0.86	0.76	0.23
BV	0.67	0.67	0.67		
SV	1.00	0.67	0.80		

TABLE B. 2 5-SHOT 3-WAY MULTI-CLASSIFICATION EVALUATION OF THE TRAINED LOGISTIC REGRESSION MODEL

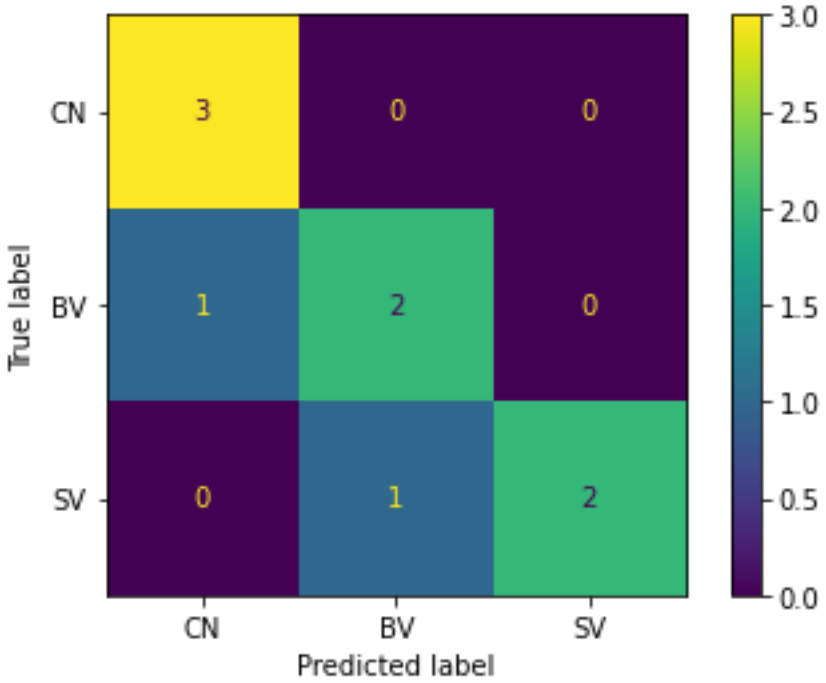

FIGURE B. 2 CONFUSION MATRIX OF THE 5-SHOT 3-WAY MULTI-CLASSIFICATION MODEL

The 5-shot 3-way multi-classification results show that the model misclassified the samples in BV and SV classes. The misclassification is obtained due to the limited training size of the support set, which is responsible to train the logistic regression model. Apart from misclassification, the model achieves significant classification performance of the model with only using 15 training samples or shots. Thus, the model achieves a good classification performance with fewer training samples and only two misclassifications were obtained. This significant achievement is acquired through the embedding of ADNI fine-tune model features as prior knowledge, which helps the model to classify the FTD data samples with fewer training samples.