UNIVERSITY OF CAMERINO

SCHOOL OF SCIENCE AND TECHNOLOGY

Master of Science in Computer Science

Class LM-18

Experimental thesis

FRONTOTEMPORAL DEMENTIA DETECTION USING MACHINE LEARNING TECHNIQUES

Candidates

Keerthi Ravilla Subramanyam

Supervisor

Prof. Francesco Amenta

AMENTA FRANCESCO
2023.03.27 22:46:00
CN=AMENTA FRANCESCO
C=IT
2.5.4.4=AMENTA
2.5.4.42=FRANCESCO
RSA/2048 bits

ACADEMIC YEAR 2021/2022

Abstract

Frontotemporal Dementia (FTD) is a neurocognitive syndrome in which a person's cognitive function deteriorates to the extent that it interferes with their daily activities. FTD is the second most common form of dementia in those younger than 65 years and is expected to increase in prevalence as the population ages. It can be generally diagnosed with a medical history, neuropsychological examinations, neuroimaging, and a detailed neurological examination. Unfortunately, there are no specific treatments for any FTD type.

Treatment is mainly symptomatic with medications that can reduce agitation, irritability, or depression. These treatments should be used to help improve quality of life. FTD gets worse with time and the speed of decline differs from person to person. For better treatment and decline of this dementia it is recommended to diagnose at early stages and make this process more feasible and effective, in this work we adopted machine learning techniques and model training that will be carried out on patient medical history data for early detection of FTD. The project outcomes can be more effective in the early treatment and reduce the risk of getting it worse.

In this study, we propose a machine learning-based approach for FTD detection using magnetic resonance imaging (MRI) data. We use a dataset of MRI scans from patients with FTD and healthy controls and apply various machine learning algorithms such as Support Vector Machines (SVM), Random Forest (RF), and Convolutional Neural Networks (CNN) for feature extraction and classification. We evaluate the performance of each algorithm in terms of accuracy, precision, recall, and F1 score.

The thesis covers a detailed study and understanding of different FTD types and their behaviors. A comprehensive analysis of the available tools and the state of the art of algorithms will be presented and performance estimations are done on this model in order to check whether these effectively predict and diagnose the FTD at early phases. Patient data will be collected from different sources including public libraries like Kaggle, ML repositories. Data preprocessing including of hyper parameter tuning, missing value identification, and replace missed values with the highest probabilistic one to avoid performance bias. This process will be followed

by model training by feeding it with the medical data.

Different supervised and deep learning models will be included to conduct, with cross-validation techniques we evaluate different performance metrics such as accuracy, sensitivity, recall, AUC etc. The performance values of different ML models will be further compared to identify the best model.

Our results show that the CNN-based approach outperforms the other algorithms, achieving an accuracy of 92% and an F1 score of 0.91. We also conduct a feature importance analysis and identify the most relevant brain regions for FTD detection. Our findings suggest that machine learning-based approaches can provide a reliable and accurate tool for FTD detection, and may have significant implications for early diagnosis and treatment of the disease.

Keywords: Dementia, Frontotemporal dementia, Support vector machine, data pre-processing, Random Forest, Convolutional neural networks, Magnetic resonance imaging, Data pre-processing, F1 score.

An investment in knowledge
pays the best interest.

- Benjamin Franklin

Ringraziamenti

I would like to express my heartfelt gratitude to Prof. Francesco Amenta for providing me with invaluable support and guidance during the completion of my thesis. His advice and mentoring were invaluable in guiding me through the project's hurdles and intricacies. His views, expertise, and understanding were crucial in assisting me in making the appropriate decisions and achieving the desired outcomes. I feel privileged to have had the opportunity to collaborate with someone as knowledgeable and dedicated as him. His skill has been an inspiration to me, and I aspire to mimic his attributes in my future pursuits.

I want to express my appreciation and sincere thanks to Dr. Gopi Battineni and Dr.Getu Sagaro for their assistance in helping me comprehend and complete the thesis. I would also like to extend my gratitude and thanks to all the distinguished professors in the department of Computer Science. I am grateful for the opportunities and experiences that this institute has provided me with. The exposure to a diverse range of subjects, the numerous extracurricular activities, and the practical training has equipped me with the skills and knowledge necessary to excel in my future endeavors.

I believe that studying at this institute has not only honed my academic skills but has also shaped me into a well-rounded individual. The institute's commitment to fostering a sense of community and encouraging students to pursue their passions has helped me discover my interests and strengths. In conclusion, I cannot express enough how proud I am to be a part of this institute, and I am truly thankful for the education and experiences it has provided me with. I look forward to continuing my journey here and contributing to the institute's legacy.

Thank you,

Keerthi.R.S

Table of Contents

1 Introduction 9

1.1 Stages of dementia 10

1.2 Dementia types 12

1.3 Diagnosis of FTD 15

1.4 Machine learning 18

1.4.1 Machine learning Workflow 20

1.4.2 Machine learning in healthcare 24

2 Research Objectives 26

3. Related works 27

4 Methodologies 28

4.1. Convolutional Neural Networks 28

4.1.1 Training Convolutional Neural Network 34

4.1.2 Data Preparation for Training and Evaluation 36

4.1.3 Transfer Learning 39

4.1.4 Regression Algorithms 40

4.1.5 3D ResNets Architecture 42

4.1.6 Embedding Learning 44

4.1.7 Problem formulation 45

4.2 Support vector machine 46

4.3 Random Forest 51

4.4 Artificial Neural Networks 52

4.5 Segmentation Tools 54

5 Implementations 57

5.1 CNN Model Implementation 57

5.1.1 Data collection 58

5.1.2 Data Pre-processing 58

5.1.3 3D CNN models 60
5.1.4 Features Embedding 63
5.1.5 FSL Experimentations 63
5.2 SVM and Random Forest Implementation 65
5.2.1 Data collection 66
5.2.2 Data Pre-processing 66
5.2.3 Experiment Analysis 67
5.3 Models Developing Environment and Tools 68
6 Results and discussion 70
6.1 SVM and Random Forest Results 70
6.2 Conventional Neural Network Results 72
6.2.1 ADNI Model 1 Results 72
6.2.2 ADNI Model 2 Results 73
6.2.3 ADNI Model 3 Results 74
6.2.4 Comparison of ADNI Models 75
6.2.5 Feature Embedding Model results 76
7 Conclusion 80
References 82
Appendix A 92
Appendix B 93

List of Figures

fig 1.1 Brain atrophy of frontotemporal lobe in FTD subjects..... 15

fig 1.2 General workflow of machine learning models.....23

fig 4.1 Basic architecture of convolutional layers..... 30

fig 4.2 Convolutional operation on image (4x4).....32

fig 4.3 Max pooling operation on (4x4) image.....33

fig 4.4 Deep neural activation function.....34

fig 4.5 Data preparation and training process of model.....38

fig 4.6 5-fold cross validation visualization.....39

fig 4.7 Transfer learning methods..... 41

fig 4.8 Network architecture of 3D RESNETS.....45

fig 4.9 Meta-testing task for simple logistic regression..... 45

fig 4.10 Data classification using SVM.....48

fig 4.11 Kernels types..... 49

fig 4.12 Influence of C in SVM's.....50

fig 4.13 Illustrates of using Gamma..... 50

fig 5.1 ADNI data preprocessing methods.....60

fig 5.2 3D AD MRI scans of ADNI datasets.....61

fig 5.3 Loss and accuracy curves of trained 3D CNN models.....62

fig 5.4 Loss and accuracy curves of ANI model-2..... 63

fig 5.5 Loss and accuracy curves of ADNI model-3.....64

Fig 5.6 Feature embedding process of ADNI and NIFTD data.....65

1 Introduction

"Dementia" is a descriptive and diagnostic Latin term, used to describe a chronic, largely irreversible syndrome that results in cognitive decline beyond the normal consequences of biological aging[1]. It affects memory, thinking, orientation, comprehension, arithmetic, and learning. According to the World Health Organization (WHO), over 47 million people worldwide suffer from dementia, with that figure anticipated to rise to 82 million by 2030 and 150 million by 2050[2]. Aging is the greatest known risk factor for developing dementia. however, dementia is not a normal part of aging[3]. In rare cases, some people develop dementia in middle age, and this dementia is referred to as young-onset dementia.

Although the cause of dementia is unknown in many cases, neurodegeneration (deterioration of nerve cells that impair decision-making and memory) or the mechanisms of non-neurodegeneration are anticipated to be the underlying cause of dementia[2]. Different types of dementia are associated with specific types of brain cell damage in specific regions of the brain. Neurodegenerative diseases such as frontotemporal dementia cause abnormal protein deposition in the brain[2]. Different protein structures are seen in different types of dementia. For example, beta-amyloid and a protein called tau are associated with Alzheimer's disease, and alpha-synuclein protein is associated with Lewy body dementia. Changes in the blood vessels of the brain can lead to vascular dementia[4].

The most common cause of dementia in the elderly is neurodegeneration. Between 60% and 70% of people with dementia develop Alzheimer's disease[5]. About 20-30% have vascular or mixed causes and Alzheimer's disease. Dementia patients often have communication problems because their memories do not activate the speech-producing mechanisms in their brains. In addition to this problem, patients with dementia of any type have an increased incidence of major depression[6].

Doctors use different diagnostic techniques to determine the exact nature of a patient's dementia[7]. For example, an electroencephalogram (EEG) measures electrical activity in the brain through electrodes on the scalp. Positron emission tomography (PET) uses a radioactive substance that creates images on a computer screen. Doctors also use functional magnetic resonance imaging (fMRI) to check

for changes in blood flow and grey matter density in the brain[7].

1.1 Stages of dementia

Healthcare providers use the Global Deterioration Scale (GDS)[stages gds], a comprehensive tool to assess the seven stages of dementia in elderly patients[8]. This reliable method allows caregivers and health professionals to determine the extent of dementia progression in elderly patients and possible future symptoms. each of the seven stages of dementia[9]. Dementia stage charts can help caregivers track and monitor patients' conditions for stage-related symptoms.

No dementia: stages 1 to 3

Stage 1 – At this stage, there are no signs of dementia. The person is functioning normally and has no signs or symptoms[10].

Stage 2 – Very mild cognitive impairment. At this point, people begin to experience "normal" forgetfulness [8]. This sign is often associated with aging, causing loved ones and professionals to overlook the underlying cause.

Stage 3 - Mild cognitive impairment. At this point, loved ones may begin to notice forgetfulness, difficulty concentrating, and difficulty speaking [9]. This is the final stage in this category before the onset of dementia.

Stage 4: Moderate cognitive decline

Significant personality changes as well as overt, external indicators of cognitive deterioration characterize stage 4 dementia [10]. Dementia is typically not recognized until stage 4 or later. The GDS classifies stage 4 dementia as mild dementia even though the medical term for it is Moderate cognitive dementia [8].

Social withdrawal, emotional irritation, lack of responsiveness, decreased intellectual acuity, trouble with everyday tasks, forgetting of recent events, and denial of symptoms are some of the signs of stage 4 dementia [1].

Stage 5: Moderately severe cognitive decline

In the seven stages of dementia, this stage marks the beginning of what many medical experts refer to as the "mid-stage"[10]. At this stage, a person could no

longer be able to do basic ADLs (activities of daily living), including dressing or bathing, or IADLs (instrumental ADLs), without some help from a caretaker. Although each dementia patient will advance at a different rate, middle-stage dementia frequently lasts between two and four years[11].

Wandering, forgetfulness, confusion, and sundown syndrome are all symptoms of stage five dementia. further loss of mental clarity and problem-solving abilities. Significant memory loss, including recollections of recent events and personal information[12].

Dementia Stage 6: Severe cognitive decline

When dementia reaches stage 6, basic daily activities including eating, using the restroom, and other self-care require caregiver assistance [10]. Seniors who are going through this particularly severe stage of dementia may struggle to regulate their sleep, socialize, or behave appropriately in public.

Some of the common symptoms of stage 6 dementia are urinary or faecal incontinence, aggression and anxiety, mental problems such as paranoia or delusions, difficulty to do ADLs, memory impairment, and inability to identify family members and caregivers [9].

Dementia Stage 7: Very severe cognitive decline

When dementia hits stage 7, which is thought to be the end stage, a person cannot care for themselves [10]. Typically, patients with severe dementia lose all verbal abilities and have very limited movement. Basic bodily functions including breathing, eating, and swallowing are all affected by the symptoms of late-onset dementia [8].

Stage 7 dementia symptoms include difficulty speaking, poor motor coordination, and the inability to walk independently.

Dementia has seven stages which are classified into three progressive phases, early, mid and late stages based on the intensity and indications of the symptoms[7]. The terms mild, moderate, and severe are also occasionally used to describe them.

1.2 Dementia types

Dementia can be categorized into different types, based on the area of the brain affected and the symptoms that the patient exhibits[13]. Each type of dementia affects people differently, although sharing certain early symptoms. About 19 out of every 20 dementia patients falls under one of these four basic types[14] of dementia: Alzheimer's, vascular dementia, Lewy body dementia, and frontotemporal dementia.

1.2.1 Alzheimer's

Alzheimer's disease, named after "Alois Alzheimer," is the most common type of dementia[15]. It has such a high occurrence that it outnumbers stroke, cardiovascular disease, and cancer incidents in those over the age of 60[15]. Although not all causes of Alzheimer's are known[16], experts believe that a small percentage is related to mutations of three genes, which can be passed down from parent to child [19]. While several genes are likely involved in Alzheimer's disease, apolipoprotein E4(APOE) is one of the most crucial genes that raises a person's chance of diagnosis by three to ten times[17].

The two distinguishing factors of Alzheimer's disease, amyloid plaques, and neurofibrillary tangles, set it apart from other dementias[18], [19]. Beta-amyloid protein clumps are known as plaques, whereas tau protein fibrous tangles are known as tangles[20]. It is believed that these aggregates cause damage to healthy neurons and the fibers that connect them.

1.2.2 Vascular dementia

Also known as post-stroke or multi-infarct dementia[21] is the second most common kind of dementia after Alzheimer's. Brain damage that results in unrestricted blood supply to the brain is a major cause of vascular dementia. Brain strokes, untreated blood pressures, and diabetes are the major causes of this brain damage. There are 4 different types of vascular dementia, they are Subcortical vascular dementia, stroke-related dementia, multi-infarct dementia, and mixed dementia[22].

Subcortical dementia is a condition, in which the brain's subcortical areas are damaged[23]. Stroke-related dementia is caused by blocked blood vessels in the

brain as a result of strokes. several smaller strokes causing brain damage result in multi-infarct dementia. While in mixed dementia, more than one underlying type of dementia may be the cause[23].

1.2.3 Lewy body dementia

Lewy bodies dementia is a common type of progressive dementia[24]. it is caused by Lewy bodies named after FH Lewy[25], which are abnormal balloon-like clumps of proteins found in the nerve cells of brains. People with Lewy bodies have plaques and tangles in their brains[25], which are symptoms of Alzheimer's disease and this protein is also associated with Parkinson's disease. Sleep difficulties, Visual hallucinations, movement disorders Fluctuating attention, and depression are the symptoms of this dementia[26].

1.2.4 Frontotemporal dementia

Frontotemporal dementia (FTD), is a relatively uncommon type of dementia (compared to Alzheimer's)[27] that is caused by a neurodegenerative disorder with high clinical, genetic, and pathomorphological diversity[28]. It is the third most prevalent cause of dementia in persons aged 65 and higher. While being the second most common cause of young-onset dementia (aged 45 to 65)[29].

Abnormal protein build-up within neurons gradually damages and destroys brain cells, these destroyed neurons are called "pick bodies" [28]. Which are major causes of frontotemporal dementia. It is mainly a sporadic disease that primarily affects the frontal and temporal lobes of the brain [27] responsible for personality, behaviour, language learning, motivation, abstract thinking, and executive function. Although the exact cause of protein accumulation is still unknown, geneticists have discovered an abnormal gene that is inherited in families associated with FTD and Pick's disease making it as only known risk factor[28].

Causes of FTD:

In some cases, FTD symptoms appear gradually and progress steadily, if not quickly. They differ from person to person, depending on the areas of brain impairment[30]. Apart from the genetic factors involved in 30% of cases of FTD[31], the other causes of FTD are still unknown and are the subject of research

programs. Among neurodegenerative dementias, FTD accounts for about 20% of cases while Parkinson`s disease accounts for approximately 70% of cases [27]. In the majority of cases, FTD occurs in patients younger than for Parkinson`s disease (around 55 years of age for FTD and over 60 years of age for Parkinson's disease).

The most recent epidemiological studies show that FTD is the second most frequent cause of early-onset dementia[15], surpassed only by Alzheimer's disease (AD), with an incidence of about 20-30 cases per 100,000 inhabitants per year. FTD is a common cause of early-onset dementia in people under the age of 65. The overall incidence of FTD ranges from 1 to 17 cases per 100,000 people[32]. In individuals of more than 70 years of age, the range narrows from 1 to 4 cases per 100,000[31].

Types of FTD:

The following disorders are part of the core FTD spectrum (Fig:1.1): behavioral variant FTD (bvFTD), non-fluent/agrammatic variant primary progressive aphasia (nfvPPA), and semantic variant PPA (svPPA)[33]. Frontotemporal dementia with motor neuron disease (FTD-MND), progressive supranuclear palsy syndrome (PSP-S), and corticobasal syndrome are all FTD disorders (CBS)[27].

There are three types of frontotemporal disorders (FTD: Behavioural variant frontotemporal dementia (bvFTD), primary progressive aphasia (PPA), and movement disorders[28]. bvFTD is the most common FTD[33], which involves changes in personality, judgments, and behaviours. People with this disorder might have problems with cognition but their memory will be relatively intact. Problems with the sequence and planning, repeating the same activities again and again, and acting impulsively are the common symptoms of bvFTD[34].

PPA will involve changes in their ability to communicate[35]. Facing difficulties to understand words and speaking, or becoming mute while speaking is the primary symptom of PPA. Based on the kind of symptoms with languages they develop first, PPA can be categorized into three types. They are semantic PPA, Agrammatic PPA, and Logogenic PPA[36]. Here fig 1.1 shows Brain atrophy of frontotemporal lobe in FTD subjects. Structural MRI in FTD syndromes. In bvFTD, right frontal atrophy is characteristic (A), with relative sparing of posterior structures (B). In

svPPA, there is left anterior temporal atrophy (C, D), whereas nfvPPA presents degeneration in the inferior frontal gyrus and adjacent structures (E, F).

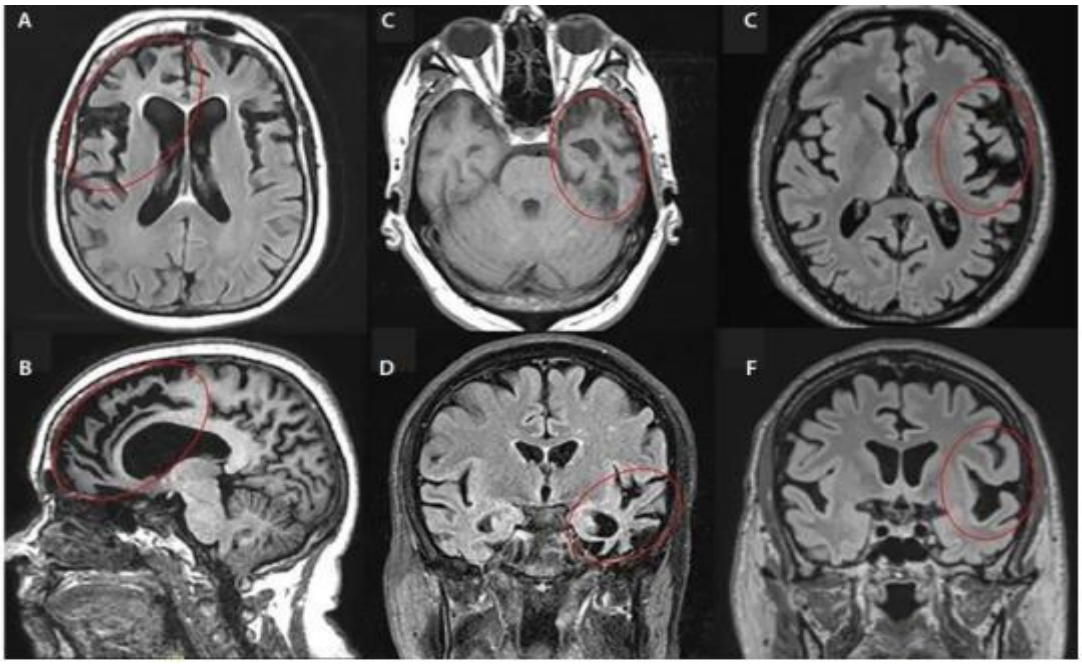


fig 1.1 Brain atrophy of frontotemporal lobe in FTD subjects.

In Semantic PPA in which the person slowly tends to lose the ability to understand single words and is sometimes unable to recognize well-known objects or faces of people[35]. Agrammatic PPA is another type of PPA in which a person tends to omit the linking words between nouns and verbs while speaking[37]. In Logogenic PPA person has trouble finding the correct words during conversations but can understand the words and sentences and has no trouble with grammar.

Two rare neurological movement disorders associated with FTD are corticobasal syndrome and progressive supranuclear palsy[31]. This occurs when the parts of the brain that control movement is affected, which in turn may affect the person's thinking and language abilities.

1.3 Diagnosis of FTD

Compared to other types of dementia, frontotemporal dementia is far less common and typically has different early signs[27]. As a result, FTD can be difficult to diagnose by doctors since its symptoms may not be recognized as dementia.

The majority of FTD-induced behavioural or psychological alterations may initially

be modest. The symptoms of depression, schizophrenia, or obsessive-compulsive disorder might be misinterpreted as bbehavioral changes[38], even when such changes are recognized as medical symptoms. Movement or linguistic issues might potentially go undiagnosed[28]. The FTD diagnosis usually includes:

1.3.1 Blood tests

Blood tests to look for illnesses including thyroid disease, B12 insufficiency, infections like syphilis or HIV, dehydration, or cancer that might resemble FTD[39]. These tests look for particular substances, proteins, hormones, and antibodies. Some of these disorders can be cured, and they are all handled quite differently.

1.3.2 Neurological exam

A thorough assessment of the complete neurological system, including physical and cognitive functions. A first examination normally takes approximately an hour and includes the following[40]:

- obtaining medical previous information.
- Physical examination: assessing walking, balance, coordination, reflexes, strength, vision, and hearing as well as motor function.
- Memory, language, reasoning, planning and organizing skills, visuospatial ability, conduct, and mood are all examined cognitively.[41]

Neurology is a complex field, and there are different specialties even among neurologists. It is important for an individual to be evaluated by a neurologist experienced with FTD and related neurodegenerative conditions.

1.3.3 Neuropsychological testing

Interviews and exams that assess mood and intellect in order to pinpoint particular areas of strength and weakness. Weaknesses frequently represent parts of the brain that are less functional[42]. Tests measure a person's linguistic, basic math, visuospatial, problem-solving, and memory abilities. A neuropsychologist interprets these tests, which are administered over the course of hours [43] They can aid in the identification of particular forms of dementia or brain illnesses and assist distinguish between depression and dementia[42].

For instance, a person with Alzheimer's disease would exhibit major memory losses on tests, but a person with FTD may perform rather well on memory tests but struggle more on language and social skills exams.

1.3.4 Lumbar puncture

Cerebrospinal fluid (CSF), the fluid that surrounds the brain and spinal cord, is collected and studied with this test[44]. Each day, our bodies produce roughly 500ml (about 16 ounces) of CSF, yet there is only room for about 175 ml (6 ounces)[45]. Regular CSF analysis can detect diseases including cancer, inflammatory processes, and uncommon infections that might resemble FTD. A lumbar puncture can also be used to detect the existence of hydrocephalus with normal pressure[45].

Some of the misfolded proteins that are building up in the brain and leading to neurodegenerative diseases[46] like FTD may be assessed as the CSF moves about and bathes the brain. A little needle is inserted during a lumbar puncture operation into the lower region of the back, just below the spinal cord. The process of collecting the fluid just takes a few minutes[44]. After the surgery, patients are allowed one hour of relaxation.

1.3.5 Neuroimaging

These three major categories—structural, functional, and molecular imaging—comprise the neuroimaging techniques used to evaluate neurodegenerative illnesses[47]. Computed tomography (CT) and structural magnetic resonance imaging (MRI), both of which enable the visualization of neuroanatomy[48], are examples of structural imaging techniques that are often employed. The term "functional imaging" refers to a broad range of imaging techniques including positron emission tomography (PET), single-photon emission computed tomography (SPECT), and functional MRI (fMRI).

These techniques measure various parameters including metabolic activity, regional blood flow, or hemodynamic changes while the patient is either at rest or performing a particular task[49]. Furthermore, methods that monitor molecular and cellular activities in live creatures are referred to as molecular imaging (e.g., specific receptors or protein aggregates). In both clinical and research settings,

structural MRI and PET are two of the neuroimaging techniques that are most frequently used to evaluate FTD[50].

Magnetic Resonance Imaging (MRI): A non-invasive method for imaging the brain and other organs that employs magnets and radio waves[51]. For the majority of brain problems, an MRI is preferable over a CT scan because it produces pictures from various angles and offers a thorough view of many brain regions that the CT scan cannot show. By using MRI, it is possible to pinpoint brain atrophy or shrinkage in certain areas that may be indicative of FTD[28]. For several minutes throughout the process, the patient must remain motionless and flat on a table. There is no discomfort or risk from the magnets in the scanner, despite the loud thumping noises it creates[52]. For the purpose of enhancing the photographs, contrast dyes may occasionally be injected into an arm vein.

Positron Emission Tomography (PET): PET scanning provides an image of how the brain functions[53]. An image of the quantity of glucose absorbed by neurons in certain brain areas is captured by a glucose PET scan. This is beneficial since glucose is the primary fuel for neurons, and decreased glucose absorption in a particular brain region indicates that the neurons in that region aren't operating as well. The quantity of amyloid in the brain is seen with an amyloid PET scan[54]. Amyloid is a protein that has been improperly folded that builds up with aging and may be heavily involved in Alzheimer's disease. Thus, an amyloid PET scan that is positive is more likely to indicate Alzheimer's disease than FTD[55].

1.4 Machine learning

Machine learning is an emerging frontier of artificial intelligence. it enables machines to learn from past data or experiences, without being explicitly programmed, in order to make, through statistical analysis and pattern matching, classifications or predictions about the future[56]. But what do we mean by learning? According to Tom Mitchell, professor of Computer Science and Machine Learning at Carnegie Mellon, a computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E[57].

ML tasks are usually described in terms of how the ML system should process an example, that is a collection of features such as pixels in the image, which is part

of a dataset. Some of the most common tasks that can be solved with ML include classifications, regressions, transcriptions, machine translation, synthesis, and sampling.

In order to evaluate the abilities of an ML algorithm, we must design a quantitative measure of its performance: for tasks such as classification and transcription, we often measure the accuracy of the model, so the proportion of examples for which the model produces the correct output, or equivalently, the error rate, so the proportion of examples for which the model produces incorrect output[57]. By what kind of experience they are allowed to have during the learning process, ML algorithms can be broadly categorized into three classes:

Supervised learning: Supervised learning is the most popular paradigm for performing ML operations[58]. It is widely used for data where there is a precise mapping between input-output data: image classification, facial recognition, email spam detection, and so on. The dataset is labelled, meaning that the algorithm identifies the features explicitly and carries out predictions or classification accordingly[59]. The term supervised learning originates from the view of the target being provided by an instructor or teacher who shows the ML system what to do. Some of the algorithms that come under supervised learning are as follows: Linear and Logistic Regression, Random Forest, Support Vector Machines, and Artificial Neural Networks[58].

The three main types of supervised-learning algorithms are regression, classification, and active learning[60]. Active learning is iterative supervised learning, which interactively queries a user to label new data points with the desired outputs. Regression techniques are used when the outputs can have any numerical value within a range, whereas classification methods are used when the outputs are constrained to a small set of values[59]. An email would be the input for a classification system that filters emails, for instance, and the output would be the name of the folder to place the email.

Unsupervised learning: In this method, the data is not explicitly labelled into different classes, that is, there are no labels and no instructor or teacher, so the algorithm must learn to make sense of the data without this guide. The model can learn from the data by finding implicit patterns: densities, structures, similar

segments, and other similar features[61]. Some of the important algorithms that come under unsupervised learning are Clustering, Principal Component Analysis, and Anomaly detection[62]. This unsupervised learning is classified into two different categories of algorithms, they are clustering and association[61].

In clustering, the data items are classified based on the existence or lack of similarities discovered by cluster analysis. In association, the rule of association is used to uncover links between variables in a sizable database using unsupervised learning techniques[63].

Reinforcement learning: It covers more areas of AI which allow machines to interact with their dynamic environment in order to evaluate the ideal behaviour in a specific context and reach their goals. With the help of reward feedback, agents can learn the behaviour and improve it in the longer run, so there is a feedback loop between the learning system and its experiences[61]. Unlike supervised learning, there is no answer key provided to the agent when they have to perform a particular task but he learns from his own experience.

1.4.1 Machine learning Workflow

Machine learning workflows specify the processes that must be taken throughout a certain machine learning implementation[64]. Machine learning workflows differ depending on the project, but four core steps are usually covered. For a general workflow diagram of machine learning refer to fig:1.2

Data collection: Data collection is one of the most important stages of machine learning workflows. The quality of the data you gather defines the potential utility and accuracy of your project during data collection[56]. To gather data, you must first identify your sources and then aggregate data from those sources into a single dataset[61]. This might include streaming data from the Internet of Things sensors, obtaining open-source data sets, or building a data lake from various files, logs, or media.

Data pre-processing: After gathering your data, you must pre-process it. Cleaning, validating, and converting data into a useful dataset is what pre-processing entails[59]. If you collect data from a single source, this may be a simple procedure. However, if you are aggregating data from many sources, you

must ensure that the data formats match, that the data is similarly accurate, and that any potential duplicates are removed.

Building datasets: This phase entails dividing processed data into three datasets: training, validating, and testing: The training set is used to train the algorithm and teach it how to analyze data. The parameters in this collection define model classifications[64]. The validation set is used to measure the model's accuracy. The model parameters are fine-tuned using this dataset. The test set is used to evaluate the models' accuracy and performance. This collection is intended to highlight any flaws or mis trainings in the model[59].

Training and refinement: Once you have your datasets, you can begin training your model. This entails providing your training data to your algorithm so that it may learn proper classification parameters and features. After training, you may use your validation dataset to enhance the model. This may entail changing or removing variables, as well as adjusting model-specific settings (hyperparameters) until an acceptable degree of accuracy is achieved.

hyperparameter tuning: Finding the optimal combination of hyperparameters to enhance the model's performance. It operates by doing several trials within a single training procedure. Every trial entails the full execution of your training application with the values of the selected hyperparameters set within the predetermined bounds[59]. Once this procedure is complete, you will have the set of hyperparameter values that the model needs to perform at its best.

Machine learning evaluations: A model's effectiveness must be evaluated in the early stages of research, and model assessment also aids with model monitoring. Several assessment criteria may be used to determine if your model(s) perform effectively when presented with new data. Accuracy, precision, confusion matrix, log-loss, and AUC are the most often used metrics for evaluating classification performance (area under the ROC curve)[65]. As the ratio of the number of accurate predictions to all guesses, accuracy indicates how frequently the classifier predicts correctly.

Precision is the percentage of projected Positives that really are Positive. When you want to be certain about your forecast, precision is a suitable assessment metric to use[66]. When developing a system to determine whether to lower a

certain account's credit limit, for instance, you must be very certain of your forecast in order to avoid upsetting your customers.

The confusion matrix, often known as the confusion table[67], provides a more thorough description of the proper and improper groupings for each class. When attempting to comprehend the differences across classes, using a confusion matrix might be helpful, especially if one class has significantly more test data than the other or the cost of misclassification could be different for the two groups[57]. Making a false positive or false negative diagnosis of cancer, for instance, has quite different repercussions. Confusion Matrix forms the basis for the other types of metrics.

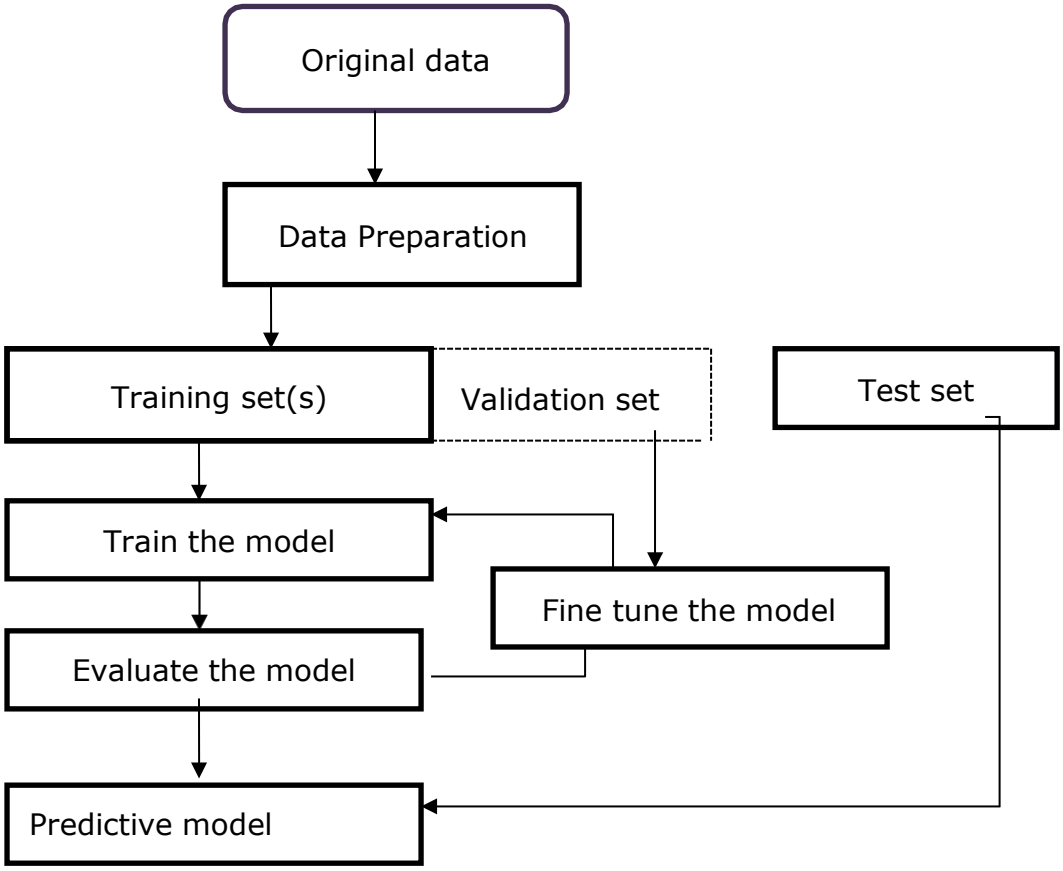


fig 1.2 General workflow of machine learning models

The number of all right predictions divided by the whole dataset size may be used to compute the accuracy for the confusion matrix mentioned above. The accuracy ranges from 0.0 to 1.0, with 1.0 being the greatest[56]. You may alternatively compute it by using 1 - ERR.

Accuracy= TP + TN / TP+TN+FP+FN

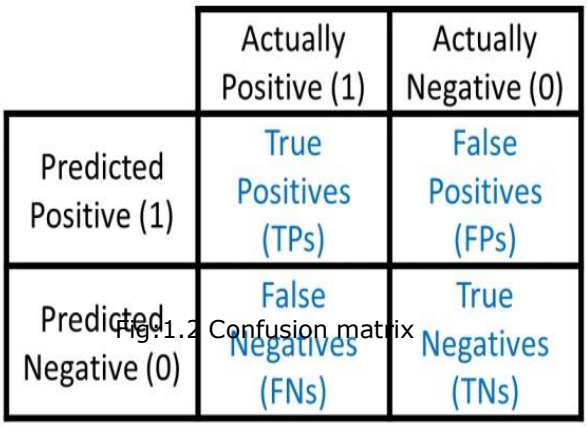


Fig.1.2 Confusion matrix

The number of accurate positive predictions divided by the total number of positives is used to compute sensitivity (SN)[67]. Recall (REC) or true positive rate are other names for it (TPR). The greatest level of sensitivity is 1.0, while the lowest level is 0.0.

Sensitivity= TN/TN+FP

The amount of accurate positive predictions divided by the overall number of positive predictions yields precision (PREC)[66]. It also goes by the name positive predictive value (PPV). In comparison, 0.0 has the poorest accuracy and 1.0 has the highest.

Precision = TP/TP+FP

The number of incorrectly predicted positive outcomes divided by the total number of negatives is the false positive rate (FPR). While the worst false positive rate is 1, the best rate is 0.0. It can alternatively be computed as specificity divided by one[57].

False Positive rate = FP/TN+FP

The number of accurate negative predictions divided by the total number of negatives is used to compute specificity (SP). Additionally known as genuine negative rate (TNR). The ideal specificity is 1.0, whereas the unfavourable value is 0.0[56].

Specificity= TN/TN+FP

These metrics are used to monitor and measure the performance of a model (during training and testing), and don't need to be differentiable.

1.4.2 Machine learning in healthcare

ML technologies are being used by healthcare organizations to track and foresee future epidemic outbreaks throughout the globe [58]. By compiling data from satellites, real-time updates from social media, and other essential information from the web, this computerized system can foresee disease outbreaks. It might be extremely beneficial, especially for developing nations with inadequate healthcare infrastructure [56]. Long lines worry about outrageous costs, a drawn-out and complicated appointment procedure, and difficulty finding the right healthcare provider are some of the root reasons that ML and similar data-driven approaches address [64]. This is due to the fact that ML systems' strengths—vast databases and clever search algorithms—are excellent at problems involving pattern matching or optimization.

By analyzing enormous volumes of data, clinical decision support systems help diagnose conditions, select the next step in therapy, spot possible problems, and enhance the effectiveness of patient care overall [65]. With its rising popularity in recent years, machine learning (ML) is an excellent tool that helps doctors do their jobs more quickly and effectively while lowering the probability that they would make a mistaken diagnosis or recommend an inefficient course of therapy. This is brought on by the growing use of electronic health records and the digitization of a variety of data points, including pictures taken on medical equipment. For a very long time, medical pictures such as X-rays were analog [58]. This has made it more difficult to use technology for general sickness studies, case categorization, and anomaly identification. Fortunately, the industry's transition to digitalization has led to more significant opportunities for these forms of data analysis, including ML.

Another field that expects to gain a lot from ML is the field of research. Clinical trials can cost too much money and take a long time to finish [60]. By applying ML-based predictive analytics to locate potential clinical trial volunteers from numerous data sources, including social media, prior medical visits, and others,

researchers may narrow their pool of potential trial participants. Observing trial participants in real-time is another way to use ML in this circumstance. These technologies can also help scientists choose the best sample size for testing and use electronic records to fix database mistakes. This paper's main goal is to discuss the enormous potential of ML in the healthcare industry.

1.4.3 Applicability of Artificial Intelligence in Medical Imaging

In many areas of healthcare, accurate diagnosis and prognosis are crucial. The application of artificial intelligence (AI) to medical imaging enables automated illness diagnosis, histology, stage, or subtype characterization, and patient classification based on response to therapy or prognosis [67]. Additionally, it enables drawing specific regions in the images, calculating organ volumes, and extracting features from the images, which when used in conjunction with machine learning algorithms, results in the quantification or categorization of image qualities.

Thanks to digitization, inexpensive data storage, and enhanced imaging techniques, an unprecedented amount of digital imaging data has just been available in medicine [70]. This generates a hitherto unheard-of level of interest in AI applications to images, which has accelerated Italian medical physicists (MP's) research efforts. A significant drawback of increased sensitivity is the detection of subtle changes with uncertain significance. An examination of screening mammography, for instance, revealed that while artificial neural networks do not regularly outperform radiologists in the detection of cancer, they do have superior sensitivity for abnormal results, particularly for small lesions [58].

The medical community must foresee the possible ambiguities of this technology at the outset of an AI-assisted diagnostic imaging revolution to enable its efficient and secure integration into clinical practice [61]. Establishing AI's place in clinical medicine requires careful consideration of its potential drawbacks in light of its special talents, and finding the right balance between improved detection and overdiagnosis won't be simple [63]. To improve the quality and interpretability of AI studies, this assessment must consistently incorporate out-of-sample external validation and well-defined cohorts.

MRI head scan analysis is made better by AI techniques. For instance, in the Multimodal Brain Tumor segmentation challenge, Don et al. provided a system with an accuracy of 86% for the detection and segmentation of brain tumors (BRAT) [65]. Input forth an algorithm that may accurately diagnose Alzheimer's disease (AD) by examining MRI head scans up to 90% of the time. Analyzing MRI head scans has been suggested as a method for detecting AD. Frontotemporal dementia (FTD) and various other techniques have been presented for the automatic classification of frontotemporal dementia. Other diseases can also be categorized by utilizing machine learning approaches [69].

2 Research Objectives

We will mainly be focusing on classifying FTD dementia using 3D MRI scans datasets. To work with these medical image segmentations, feature extractions, and classifications of dementia, a convolutional neural networks model, SVM, and Random Forest models have been applied.

The ability of three models to extract useful information from 3D MRI images has allowed us to more accurately diagnose dementia early on and track its progression. Researchers have made progress in their analysis of 3D MRI head images for the purpose of detecting different kinds of dementia. The major goal of this thesis is to identify the machine learning strategy that can offer the best accuracy in detecting FTD.

Despite the great accuracy that CNN networks have been able to attain with 3D MRI data, no articles have shown any encouraging outcomes when using smaller training sets of 3D MRI data. There is now a high need for models that can operate with few training data and produce results that are equivalent to those with many training samples.

The two objectives of this CNN model are:

a) Using the ADNI's data on Alzheimer's disease, create a feature extraction model. Then, evaluate these models using a gold-standard cross-validation approach and on different evaluation metrics.

b) Use the feature extraction model based on Alzheimer's disease data as prior knowledge for classifying Frontotemporal dementia (FTD) in a few-shot learning manner.

3. Related works

The primary issue in medical imaging is the gathering of a large number of training samples such as MRI scans or CT scans, etc. owing to patient privacy issues or clinical rules. MRI head scans, which are a significant source in training deep CNNs, are typically used to classify Alzheimer's disease or subtypes of dementia. Another issue that arises during the categorization of medical pictures is the restricted computing capacity available owing to the processing of 3D MRI scans.

CNNs have achieved decent classification with a large number of training samples; nonetheless, one of the key obstacles in training deep convolution networks is a limited number of training data. The transfer learning approach is used by researchers to address these flaws. Yi et al trained a 3D convolutional neural network (CNN) based on MRI head scans from the ADNI consortium's screening stage using a transfer learning approach to identify picture attributes that signal Alzheimer's disease progression [47]. Marica et al successfully identified Alzheimer's disease using the power of transfer learning while also optimizing the deep network design[52]. They train the architecture with pre-trained weights from huge benchmark datasets of natural photos and get better results than existing deep-learning approaches.

Another study recommends utilizing three imaging modalities of biomarkers, such as MRI, FDG-PET, and CSF, for the categorization of AD/MCI. The mixed modalities strategy outperforms a single modality-based categorization of AD/MCI in this study[62]. Recent research for detecting FTD using MRI biomarkers obtains an accuracy of 83% for distinguishing FTD from other dementia diagnostic groups.

Although researchers have used several approaches to get over the issue of constrained computer capacity and scant training data. To improve the performance of our model, we also use transfer learning, but the study devoted to categorizing FTD used only a very small number of training samples—less than even 10—which is quite sparse. In our research, we looked at this issue using the

recently developed Few-shot learning machine learning technology. Recent studies have used the Few-shot learning method to examine the brain's activation maps.

Recent studies have used the Few-shot learning method to examine the brain's activation maps. They created a neuroimaging benchmark dataset and examined several learning approaches, such as meta-learning and backbone networks. They found from their research that the few-shot approach may efficiently decipher brain signals. Another study for the objective of medical image segmentation using few-shot learning developed a novel network known as a prototype network for a few-shot image categorization.

The baseline study is also taken into account in our effort to classify FTD with few training samples. The embedding learning approaches are used in the baseline study. The backbone idea for few-shot classification tasks is meta-learning algorithms, which focus on learning algorithms that can rapidly adapt to new test time challenges with relatively few data samples. They argued in the preliminary research that excellent learned representations are more powerful for few-shot classification tasks than existing meta-learning techniques.

4 Methodologies

The approaches and techniques we employed to realize the thesis objective are all illustrated in this chapter. This thesis aims to detect frontotemporal dementia (FTD) using MRI scans. Despite the fact that we began the implementation by creating transfer learning models for the diagnosis of Alzheimer's disease. With its pre-trained weights, the Resnet-18 architecture was used. In order to categorize the FTD illness and its variations using few-shot learning, we later extracted AD model characteristics for the embedding of FTD model features. Our grasp of embedding learning and few-shot learning will be much enhanced by the approach covered in this chapter.

4.1. Convolutional Neural Networks

Convolutional Neural Networks (CNN), a class of Artificial Neural Networks (ANN), are regarded as cutting-edge technology for addressing vision challenges in computer vision applications[67]. Each layer in ANN is connected to its equivalent

layers, resulting in an excessive number of parameters in the network, which necessitates a vast amount of memory and processing capacity making it an unfeasible solution in solving vision tasks[68].

For example, a CIFAR-10 dataset picture size of 32 x 32 x 3 pixels necessitates a total of 3072 weights, which is still feasible with ANN design, but these fully-connected structures do not scale to bigger images. Given an image size of 224 x 224 x 3, which equals 150,258 weights for a single fully-connected neuron in the first layer, the number of parameters in the subsequent layers of the network would grow exponentially[69]. With this many parameters, the network would overfit. Another shortcoming of the fully linked architecture is that it entirely disregards picture spatial information.

Convolutional Neural Network design is the solution to the challenges we encountered with fully linked networks. CNN's architecture is identical to that of the human brain [67]. Furthermore, the cell connection pattern is identical to that of the human brain, and the entire organization is inspired by the visual cortex. CNN was created to automatically and adaptively learn spatial hierarchical characteristics ranging from low to high. CNN architecture is made up of several layers, including a convolution layer, a pooling layer, activation functions, and a fully linked layer [68]. These CNN blocks will be addressed briefly in the following sections. Below figure 4.1, explains the basic architecture of convolutional layers followed by a pooling and fully connected layers are that responsible for predicting tumours from the Brain MRI scans.

Convolutional Layer

The convolutional layer is the foundational layer or the central component of a convolutional neural network (CNN)[70]. This layer is in charge of extracting information from an input picture and often comprises a combination of linear and non-linear processes such as convolution functions and activation functions. The feature extraction process utilized by the convolution layer involves applying a small square of inputs known as kernels or filters across the whole input picture[71].

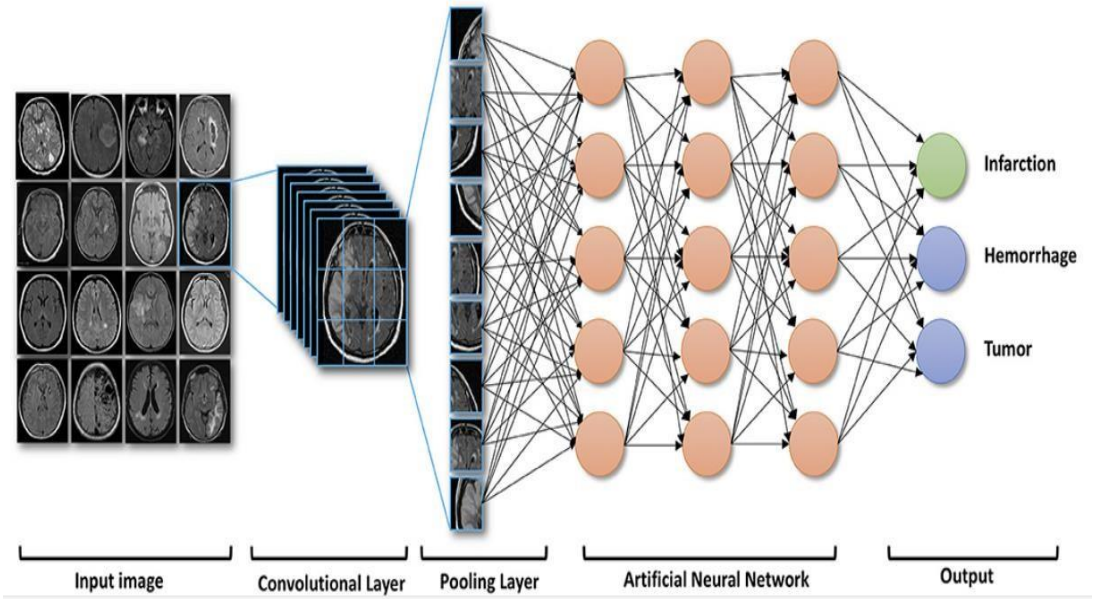


fig 4.1: The basic architecture of convolutional layers

The convolutional procedure is depicted in Figure 4.2. To acquire the depth of the feature maps, this process needs two hyperparameters, namely the size of the kernel and the number of filters. The typical kernel sizes that are taken into account are (3 x 3), (5 x 5) and (7 x 7), and the padding should have a default value of zero in order to preserve the size of the input picture.

The stride is the quantity of pixels that traverse the input image. With stride set to 1, the kernel moves forward by one pixel after each filter operation. When (stride = 2), the kernel moves every time by 2 pixels, and so forth. 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 (($n+2p-f\ s+1$) x ($n+2p-f\ s+1$)). In the case of RGB images, the image and filter size can be modified as: (n x n x nc) and (n x n x $n\ c$).

The quantity of filters employed in the convolution process affects the depth of the output feature. In order to extract the intricate details from the input picture and create the nonlinear representation needed to transfer the input data to the classification result, the convolution function's output propagates through an activation function. During network training, the CNN-linked kernel/filters parameter can be changed.

Pooling layer

The down-sampling process, which shrinks the feature map's dimension while preserving the spatial data, is carried out by the pooling layer[72]. The learnable

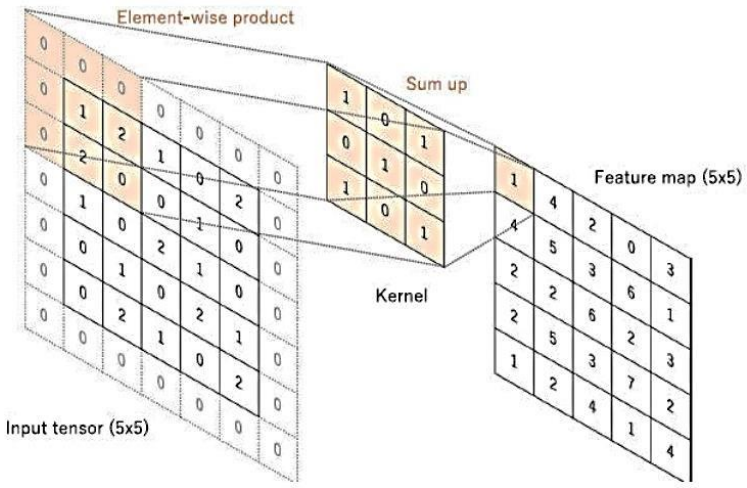


fig 4.2 CONVOLUTION OPERATION IN THE IMAGE USES (3 X 3

parameters for the subsequent convolution layers are likewise reduced due to the reduction in feature map dimension, and this creates a translation invariance to minor shifts and distortions [73]. While the operation of the hyperparameters like stride, padding, and kernel size is identical to that of the convolution process. The max pooling method is the most used kind of pooling technique. The largest value is taken from each of the input patches via max pooling, and all other values are discarded.

The max-pooling function is frequently used with a filter size of (2 x 2) and a stride of 2. There are two further categories of pooling functions: minimal pooling and global average pooling. For example, Fig 4.3 represents the max pooling operation on (4x4) image produces the feature map size of (2x2) with stride =2 and padding=0

Activation Function

The choice of these activation functions has a major impact on task performance, and the activation function plays a crucial role in training deep neural networks. These days, deep neural network training frequently makes use of the ReLU activation function. When the input to the ReLU activation function is positive, gradients easily flow because to its efficacy and simplicity, which makes it easier

to optimize than the sigmoid or tanh functions [74]. The network is able to extract intricate characteristics and patterns from the picture with the aid of the activation function, which converts linear input into nonlinear output.

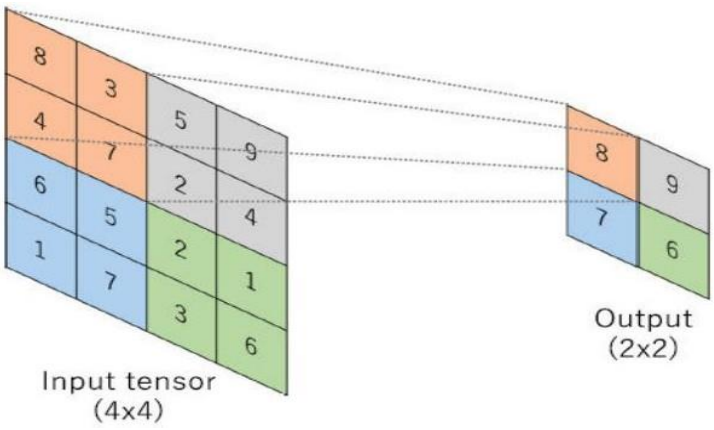


Fig 4.3: MAX POOLING OPERATION ON (4 X 4) IMAGE PRODUCES THE FEATURE MAP SIZE OF (2 X 2) WITH STRIDE=2, AND PADDING = 0

For a network to use the gradient descent approach to minimize loss during backpropagation, the best activation function must be selected (discussed in the following section of this chapter). Numerous more activation function types exist, including sigmoid, softmax, ReLU, LeakyReLU, tanh, etc. figure 4.4 represents the common activation functions used in Deep learning networks.

The sigmoid function, which translates the input value between 0 and 1, is also known as the squash function [75]. The gradient below or above the prescribed value would be close to zero since this activation function is non-linear.

Sigmoid, $\sigma(x)$ = 1 1+e−x

tanh, $\sigma(x)$ = tanh (x)

ReLU, $\sigma(x)$ = max (0, x)

In backpropagation, when the gradient reaches zero, the neuron is rendered ineffective, and the network stops picking up new parameters to learn. The vanishing gradient problem is the term used to describe this issue in deep learning.

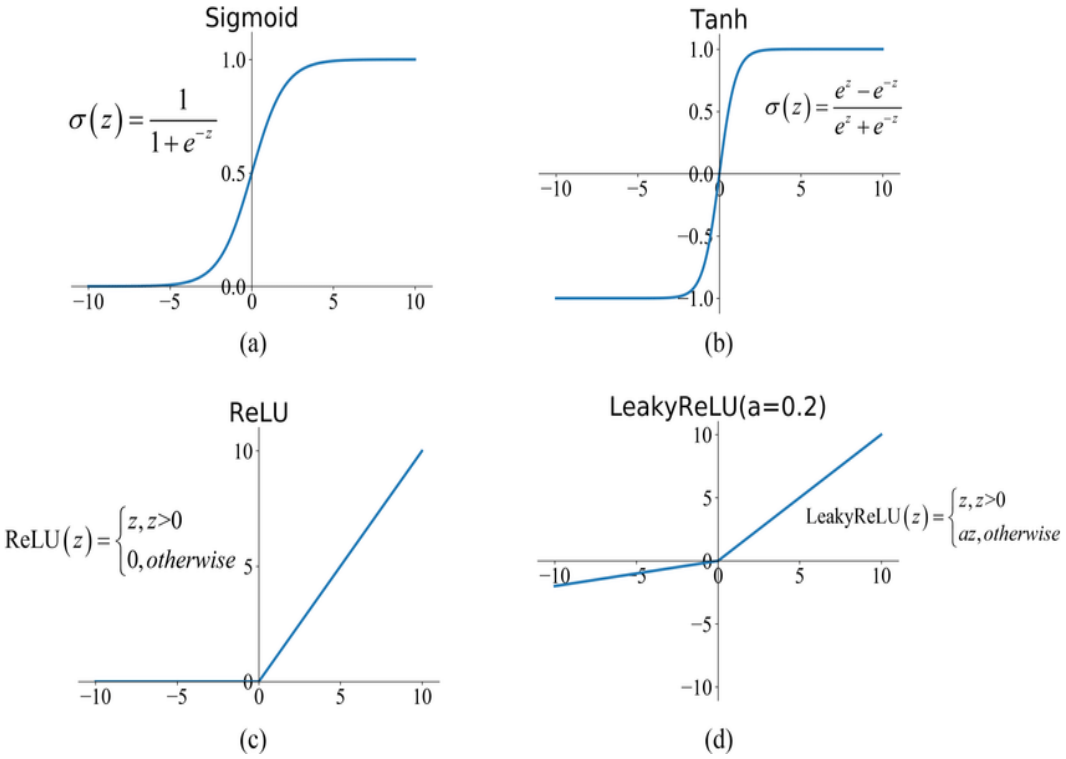


Fig 4.4: COMMON ACTIVATION FUNCTIONS USED IN DEEP NEURAL NETWORKS (A) SIGMOID (B) Tanh (C) ReLU AND (D) LeakyReLU

The softmax function is mostly employed in the last layer of the neural network. The probabilistic values are generated by the softmax function for each categorization class. With each value falling between 0 and 1 and the class probabilities being added together to equal 1, this function normalizes the real values from the last completely connected layer to the desired class probabilities.

Normalization Layer

CNNs use normalization layers to shape each layer's output into a certain range that, in experiments, exhibits quicker convergence. Only when the magnitude of the normalization should be sufficiently great does it become visible in the deep neural network[75]. Deep neural networks use a variety of normalization approaches, including batch normalization, group normalization, weight normalization, and others. Each normalization technique normalizes the variables in a unique way. These normalizing layers, however, made a negligible impact and are no longer in use.

Fully connected Layer

The fully-connected layer (fc), the top layer of the CNN, is where all of the neurons are coupled to all of the neurons in the layer below it. The final output feature map

of the convolution layer is often flattened, which transforms a 1D array of integers that are subsequently linked to one or more fully-connected layers, before the fully-connected layer [73]. Each input is linked to each output by a learnable weight in this situation, which is referred to as thick layers.

4.1.1 Training Convolutional Neural Network

In order to reduce the discrepancy between the projected outputs of the network and the target label, training of CNN entails altering 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. The loss function is used to determine the difference between actual and expected output. The weights are determined using the loss function, and the network is first configured with random learnable parameters (weight and biases) of a given range [76]. Model performance is then determined for the chosen kernel during the forward pass. Backpropagation is a technique where an optimization algorithm updates the learnable parameters to reduce the loss determined in the forward pass.

Loss Function:

The cost function is an alternative name for the loss function. It calculates the discrepancy between the provided ground truth labels and the expected network outputs as determined by the forward pass [74]. Cross-entropy is a frequently used loss function for multi-class classification tasks, while Mean Squared Error (MSE) is utilized for regression to continuous values. The primary objective of the loss function is to update the learnable parameters throughout each training cycle while minimizing the cost function generated during the forward pass [76]. One of the hyperparameters that must be determined while optimizing the network for a particular task is the loss function.

Backpropagation:

Backpropagation is the process of minimizing the cost function during the forward pass computation and modifying the learnable parameters, such as the kernel and weights, to enhance the performance of the model [77]. Using a loss function like MSE or cross entropy, the model's performance might be calculated. Only the gradients and the learnable parameters are updated via backpropagation

throughout each iteration. The Gradient Descent technique may be used to improve the performance of the model.

Gradient descent is the term used to describe how these gradients of the loss function detect the direction in which the loss function has the sharpest rise and update the learnable parameters in the negative direction of the gradient to minimize the loss for desired prediction outputs [78]. The loss function is constructed using the network's entire set of weights, and we must use partial derivatives to change these learnable weights. The learning rate, which is the step size by which the network may improve, needs to be taken into account as a hyperparameter. The range of the learning rate value is (0.1 to 0.0001). Computation of weights is defined as.,

$$\Delta\omega = r\ x\ (\partial C\ \partial\omega 1 + \partial C\ \partial\omega 2\ , \dots\ , + \partial C\ \partial\omega q\ )$$

where the modified weight vector is denoted by $\Delta\omega$. C stands for the model's cost function, 'r' for the learning rate, and 'q' for the total number of weights in the network. A huge dataset's samples cannot be processed in a single iteration due to memory limitations. Therefore, using a tiny subset of the dataset all at once, or in a process known as a mini-batch, it is possible to compute the gradient of the loss function. The technique used for this is Stochastic Gradient Descent (SGD). It is crucial to optimize the neural network using the mini-batch size and the learning rate hyperparameter.

There have been several suggested and extensively used enhanced versions of SGD, including RMSprop, AdaDelta, AdaGrad, AdaMax, NAdam, Adam, etc. Adam is the most often used optimizer due to its reliability and efficacy for a larger range of applications.

Hyperparameters:

A CNN or neural network may be trained to change the learnable parameters, such as the kernel/filters, weight, and biases of the network. The term "parameters" refers to these learnable parameters [61]. A few parameters that regulate the behaviour of the learnable parameters known as hyperparameter were covered in the earlier sections. According to their definition, these hyperparameters are features of the CNN that have an impact on the overall design or operation. These

hyperparameters can be set before the model is trained, and their effects on the model's performance can be substantial [78]. Here are some key hyperparameter definitions:

- Batch size: A small fraction of the dataset that the network is able to process in parallel (Efficient use of the computational resources)
- Epochs are the number of iterations, where each iteration involves sending all of the data via the network.
- Loss Function: This function is used to determine a batch's loss (error).
- Optimizer: The optimizer is used to reduce the loss function's error (optimize the model performance.
- Learning rate: The step size at which the optimizer updates the weights
- Activation function: Function for layer-specific activation that constricts the output feature value to a certain function value range.

To optimize network performance, each CNN or neural network has a unique hyperparameter configuration. Typically, we used grid search, a brute force approach. For each of the hyperparameters that needed to be optimized, we provided a precise value range in this grid search process. In this case, the batch size is [16, 32, 64] and the learning rate is [0.1, 0.001, 0.0001]. After the model has finished being trained, its performance is assessed on each of the hyperparameter values.

The grid search algorithm must successfully finish all three training sessions in this instance. We evaluated the model's performance once the training session was over using the chosen performance measures, and we maintained the hyperparameter settings that produced the best model performance.

4.1.2 Data Preparation for Training and Evaluation

Data preparation is crucial for a model or neural network training. In order to train the model, CNNs or neural networks often need a lot of data, which is classified into three categories: the training set, the validation set, and the test set[76]. Fig 4.5 explains the flow of data preparation and training process of the model.

Cross-Validation is the most effective method for assessing the model performance in machine learning [71]. We randomly partition the dataset into k-

folds without replacement for k-cross validation. The remaining k-folds can be used to train the model while one is kept aside as a test set. K-cross validation is typically used for model tweaking to identify the appropriate hyperparameter values for the greatest generalization performance. K-cross validation is regarded as the reliable and ideal method for assessing the effectiveness of the model.

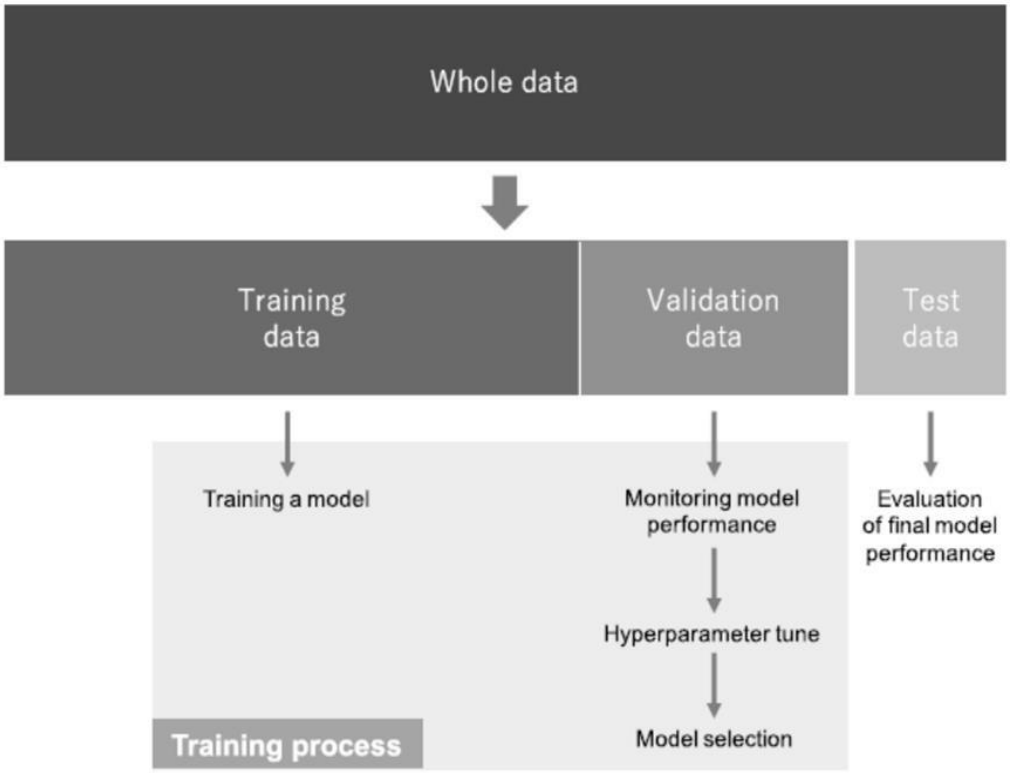


Fig: 4.5: DATA PREPARATION AND TRAINING PROCESS OF THE MODEL

A 5-fold cross-validation on the practice data is shown in Figure 4.6. The model is trained on several folds throughout each iteration before being assessed on the held-out fold. By calculating the average and standard deviation for each model's performance, a reliable model performance estimation is produced.

Performance Metrics:

During model training, the performance metric is extremely important in obtaining the best classifier. Therefore, choosing a good performance metric is a crucial element in differentiating and achieving the best classifier. Utilizing PyTorch evaluation functions and the Sklearn package, many evaluation metrics have been

employed in this study.

Accuracy [71]: The ratio of accurate predictions to the total number of examined occurrences is what this statistic measures.

Precision [73]: From the total predicted instances from the positive class, this measure is used to calculate the positive examples that are accurately predicted.

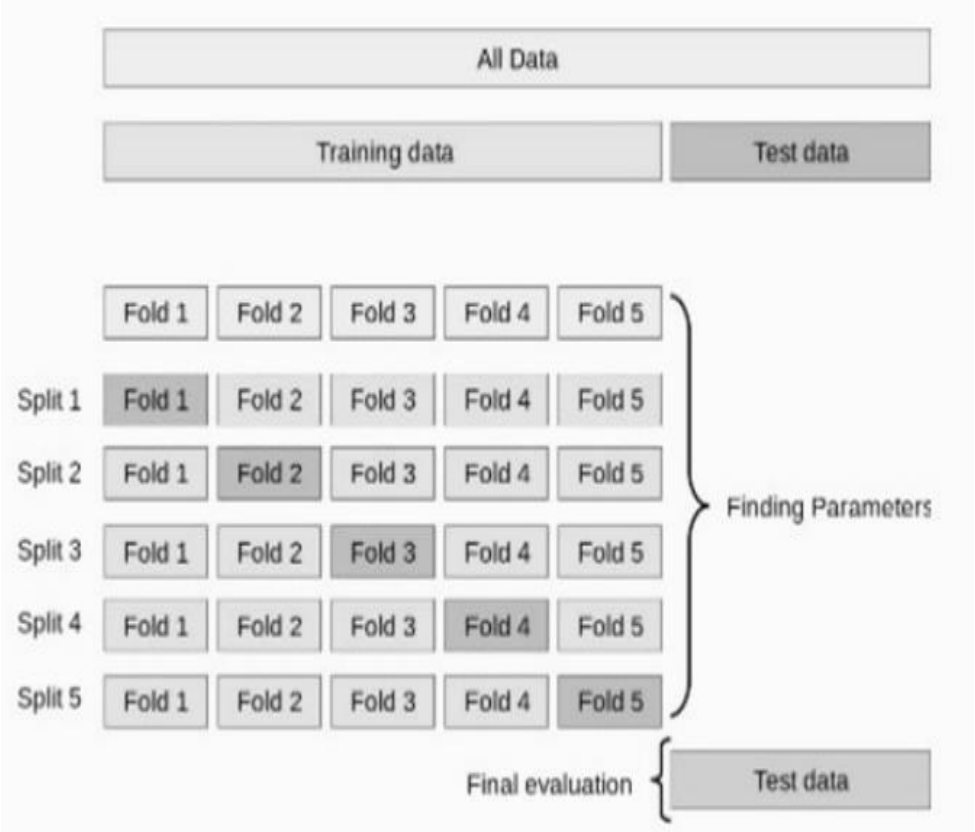


Fig: 4.6: 5-FOLD CROSS-VALIDATION VISUALIZATION

Recall [70]: Measuring the percentage of positive events that are accurately categorized using recall measures.

F1-score Metrics [73]: The harmonic values between accuracy and recall levels are measured using F1-score metrics.

Matthew's correlation coefficient [64]: Machine learning assessment metrics for binary and multiclass classification quality are based on Matthews's correlation coefficient. In its simplest form, this assessment measure is a correlation coefficient value between "-1" and "1." The ideal forecast has a value of 1, whereas the average or random prediction has a value of 0, and the inverse prediction has

a value of -1.

Balanced accuracy metric [64]: In binary and multiclass classification issues, the balanced accuracy measure is applied to deal with unbalanced datasets. It calculates the mean recall achieved across all classes. One is the best number, whereas zero is the worst.

4.1.3 Transfer Learning

Transfer learning (TL) is a machine learning (ML) research subject that is concerned with the storage of information obtained while resolving one problem and its subsequent application to another similar but unrelated problem[76]. The essential premise of transfer learning is that every image has general characteristics like circles and edges that define things like dogs, tables, and human brains[73]. As discussed in earlier sections, CNN extracts general features in the first layers and more specific information in succeeding levels. Transfer learning exploits this assumption by changing pre-trained CNNs (often trained on extremely large datasets like as ImageNet, which has 1.6 billion pictures classified into 1000 classes) into a tailored issue.

Deep learning's portability of these general learned characteristics is a significant benefit that makes it applicable in a variety of domain applications with limited training datasets[70]. Many models that have been trained on ImageNet datasets, such as AlexNet, VGG, GoogleNet, and ResNet, are freely available and may be altered for various purposes. The transfer learning technique has traditionally been used in two ways: fixed feature extraction and fine-tuning[70]. Fig:4.7 explains the different transfer learning methods used in machine learning.

During the fixed feature extraction procedure[76], we would remove the fully connected layer from the ImageNet-trained network and keep the remaining network. The remaining network would be made up of convolutional layers, a pooling layer, and activation functions, often known as feature extraction methods[77]. In this situation, a machine learning classifier such as SVM, Random Forest, or the modified fully connected layer can be concatenated on top of the feature extractor, resulting in training on a customized dataset confined to the new classifier. Because of the differences between ImageNet photos and the provided medical images, this form of transfer learning is uncommon in deep

learning research on medical images.[77]

Fine-tuning is the second procedure in transfer learning that is often used in the medical area[76]. It not only replaces the fully connected layers with a machine learning classifier comprising updated fully connected layers, but it also allows the convolutional base layers to be unfrozen based on the job requirements. All or part of the layers in the convolutional base may be fine-tuned since, as we know, the top layers have more general information, so we freeze those upper levels and retrain the deeper layers, which contain precise features, based on the individual tasks.

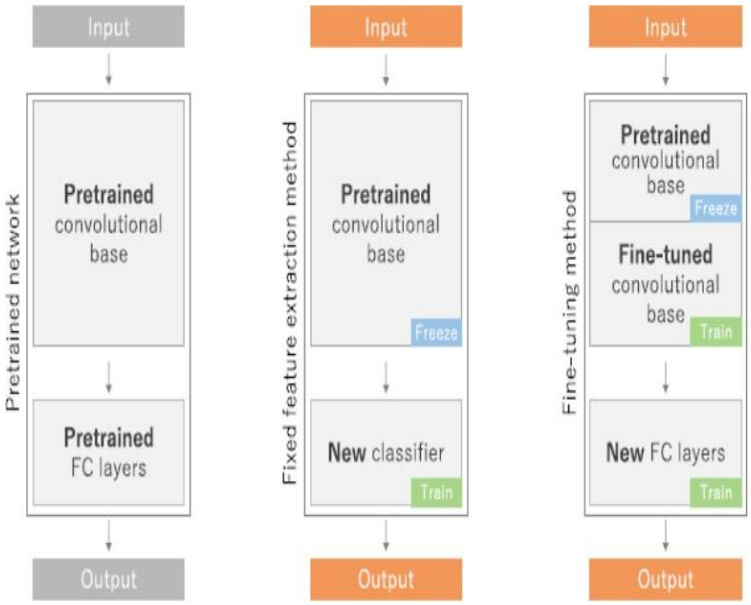


Fig 4.7: DEPICTS THE DIFFERENT METHODS OF TRANSFER LEARNING USED IN THE DEEP LEARNING

4.1.4 Regression Algorithms

Regression is a supervised machine-learning technique that predicts continuous output values based on input[78]. Regression algorithms forecast output values is based on input attributes from the system's data. The standard process involves the algorithm developing a model based on the attributes of training data and then using the model to predict the value of fresh data[58].

Linear regression, ridge regression, neural network regression, Lasso Regression, Decision Tree Regression, Random Forest, KNN Model, and Support Vector

Machines are the most popular regression algorithms[60]. Logistic regression is been used in this thesis.

Linear models:

Linear regression is one of the most commonly used models for regression problems[79]. The linear regression response is a linear function of the inputs. It demonstrates linear regression in the following manner.

$$Y(x) = W^T x + € = \sum_j^n w_j x_{j+g}$$

'ϵ', which stands for the residual error of the linear predictions and the ground truth label, signifies the product multiple of the input vector "x" and the weight vector "w." In most cases, it is believed that € has a normal or gaussian distribution, with N ≈ (μ, σ^2) (μ defined as the mean and σ^2 define as variance). The model is reformulated as follows since it also states the conditional probability [81]:

$$p(y|x,\theta) = N(y\,|\mu(x), \sigma^2(x))$$

Let's assume the standard case, where the linear function of "x" is "μ". So $\mu = W^T x$ and the noise is constant $\sigma^2 = x(\sigma^2)$, $\theta = (\omega, \sigma^2)$ are linear regression model parameters. If we replace 'x' with the nonlinear function of the inputs, $\theta(x)$ then linear regression methods can model nonlinear relationships.

Additionally, logistic regression, as it is known in this context, may be used to solve classification difficulties. The study provides the following justification for logistic regression. Assume that the y has a Bernoulli distribution with values of y € {0, 1}.

$$p(y|x,w) = Ber(y\,|\mu(x)$$

Where μ(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 ≤ μ(x) ≤ 1.

$$\mu(x) = sigm(W^T x).$$

where the sigmoid function, commonly known as the logit or logistic function, is referred to as sigma (γ). It is characterized by

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

The sigmod function, which maps the input value between the range of [0, 1], is also known as the squashing function. For the probabilistic comprehension of the outcome, it is crucial in this case. Putting the equations together, we obtain...

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

This is known as logistic regression. Applying the threshold at an output probability of 0.5 allows us to implement the decision rule shown below.

$$y(x) = 1 \Leftrightarrow p(y = 1|x) > 0.5$$

For instance, logistic regression is $\rho(yi = 1 \mid x_i , \omega)$= sigm ($\omega_i + w_i x_i$). Assume now that sigm ($\omega 0 + w1x$) =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.

4.1.5 3D ResNets Architecture

In 2018, Facebook created the 3D Resnet architecture and released it. In order to analyze the temporal relationship in video clips, the 3D ResNets architecture was created[81]. The inspiration for this project came from observations that 2D CNN when applied to common video frames, continued to perform well on the action identification challenge. The drawback of using these networks is that it necessitates vast datasets and intensive computer resources[71]. The researchers proposed to employ 3D CNNs in the residual learning framework and examine the benefits of 3D CNNs over 2D CNNs in order to counteract these negative effects and profit from the strength of 3D convolutions[82]. Table 4.1, illustrates 3D ResNets architecture that we used in transfer learning model settings.

Five convolutional blocks make up the R3D architecture, and they are typically in charge of significant feature extraction[83]. Every convolution block consists of two convolution layers layered on top of one another, followed by a Batch normalization and a ReLU activation function. A fully connected layer is added after the convolution blocks are eventually coupled using spatial-temporal pooling to decrease the network's dimensions and provide the classification output. In the parameters of our transfer learning model, this architecture is taken into account.

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

Table:4.1: ARCHITECTURE MODEL BLOCKS OF RESNET-18

In order to categorize video clips, the R3D architecture uses 3D convolution to extract the temporal information from the video. To train these R3D networks from scratch, they looked at the benchmark datasets Kinetics-400 and sports-1M kinetic video clips. The model has 18 layers and the network accepts clips as input that is made up of L RGB frames that are 112 by 112 pixels in size. The network employs spatiotemporal down sampling at conv3 x, conv4 x, and conv5 x, as well as one spatial down sampling at conv1 x implemented by the stride size of 1 x 2 x 2.

The kinetics-400 dataset serves as the model's initial training ground, while the held-out test set serves as the model's evaluation ground. Figure 4.8, illustrates the network architecture of 3D ResNets. Video clips are first downscaled to 128 x 171 before being randomly cropped to a window size of 112 × 112. These hyperparameters serve as the model's training data: Each convolution layer is subjected to batch normalization, learning rate is initialized from 0.01 and gradually declines by a factor of 10, and epochs are 45. Batch size is 32 clips per movie.

In this thesis, Alzheimer's disease is classified using a pre-trained 3D ResNet architecture, and the features produced by the AD model are then employed as prior knowledge for the downstream job to diagnose FTD disease with less training data. In this study, both feature extraction and fine-tuning transfer learning techniques were utilized to classify Alzheimer's disease, and fine-tuning model

features were employed as prior knowledge to complete the downstream job of FTD disease.

Fig:4.8: NETWORK ARCHITECTURE OF 3D RESNETS

4.1.6 Embedding Learning

The embedding learning approach embeds each data sample fromXi € X Ç $\mathbb{R}d$ to a lower-dimensional space zi € ⊴ ⊆ $\mathbb{R}m$, resulting in comparable data samples being near together and non-similar data samples being distinguished[84]. A smaller hypothesis space with fewer training samples can be created in the lower dimensional case. Although task-specific data from $D\ train$ can sometimes be used, the embedding function is mostly acquired from past knowledge.

Embedding learning had two key components: (a) a function 'f' that embeds test data samples from $D\ test$ to ⊴ (b) a function 'g' that embeds training samples from $D\ train$ to ⊴[85]. There are three types of embedding learning models: (i) task-specific embedding models, (ii) task-invariant models, and (iii) hybrid embedding models, which encode both task-invariant and task-specific information.

The hybrid embedding model serves as the motivation for our research in this thesis. The hybrid embedding model learns the embedding function by combining generic task-invariant prior knowledge with task-specific knowledge from $D\ train$.

4.1.7 Problem formulation

We construct certain prefaces for developing our algorithm, which will help us reach our thesis aim. In our scenario, the collection of meta-training tasks s{($D\ train$, $D\ test$)}i=1 is the definition of the ADNI dataset. The notation ($D\ train$, $D\ test$) is termed as training and testing sets of a downstream FTD dataset, which contain few training samples. Training samples of FTD dataset $D\ train$ = {(xt,yt)}t=1 T and testing samples $D\ test$ = {(xt,yt)}q=1 Q.

In the FSL scenario, the training samples of the FTD dataset are referred to as the Support set and the testing examples as the Query set. The same distribution is used to sample both the training and testing samples. Using the pre-trained ResNet-18 model to categorize Alzheimer's disease and the extraction of model features, which is utilized to determine the embedding function of frontotemporal data samples, we create an AD classification model in the meta-training job. Figure 4.10 illustrates, meta learning tasks that were used in extracting an embedded features for logistic regressions.

The goal of meta-training is to acquire a generalizable embedding model fØ that can be applied to any new task. In our analysis, we show that optimizing the Alzheimer classification model can produce a potent embedding for our basic learner. with respect to a task (D train, D test) picked from the FTD dataset distribution. On the D train of the FTD dataset, the basic learner is trained (support set). Multivariate logistic regression was used to initialize the base learner. A weighted term and a bias term are the parameters of logistic regression using the ϴ = {W, b}.

The technique put forward in this thesis states that we require the best feature representations, which we then employ as prior knowledge for classifying FTD diseases. By creating a CNN model based on a sizable ADNI dataset and using the pre-trained video ResNet-18 model, we were able to discover the best feature representations. Additionally, we transfer these ideal feature representations to the model that will be trained using the smaller number of FTD dataset training examples. So that the downstream task model may obtain the best diagnostic accuracy for the FTD illness with less training data, we enhanced the feature size of the model.

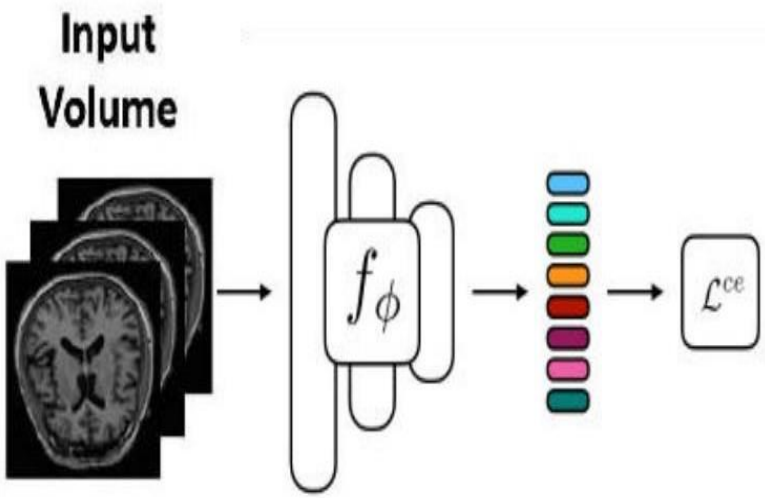


Fig 4.9: META-TESTING TASK TO EXTRACT AN EMBEDDING FEATURES FOR SIMPLE LOGISTIC REGRESSION

4.2 Support vector machine

SVMs are a prominent type of supervised learning algorithm that is used in classification and regression applications[86]. SVMs' fundamental goal is to find a hyperplane with the greatest practicable margin of separation between data points in various classes[87]. The margin is calculated by measuring the distance between the nearest data points in each class and the hyperplane[58]. The hyperplane is chosen to maximize the margin, hence boosting the model's generalizability.

A kernel function is used to convert the input data points into a high-dimensional feature space in SVMs[88]. Because of the intricate non-linear interactions between the input characteristics and the target variable, SVMs can now capture these correlations. The performance of SVMs can be significantly impacted by the choice of the kernel function. Popular kernel functions include linear, polynomial, sigmoid, and radial basis functions (RBF)[89].

Finding the hyperplane that maximizes the margin while adhering to a set of constraints is the goal of SVM optimization. These limitations guarantee that the hyperplane classifies each training data point accurately. Convex optimization techniques, such as quadratic programming, can be used to formulate and solve optimization issues. Compared to other machine learning algorithms, SVMs provide several benefits. Both linear and non-linear correlations between the input characteristics and the target variable may be handled, and they are efficient at

handling high-dimensional data. Below fig:4.10 represents the graphical representation of data classification using SVM model.

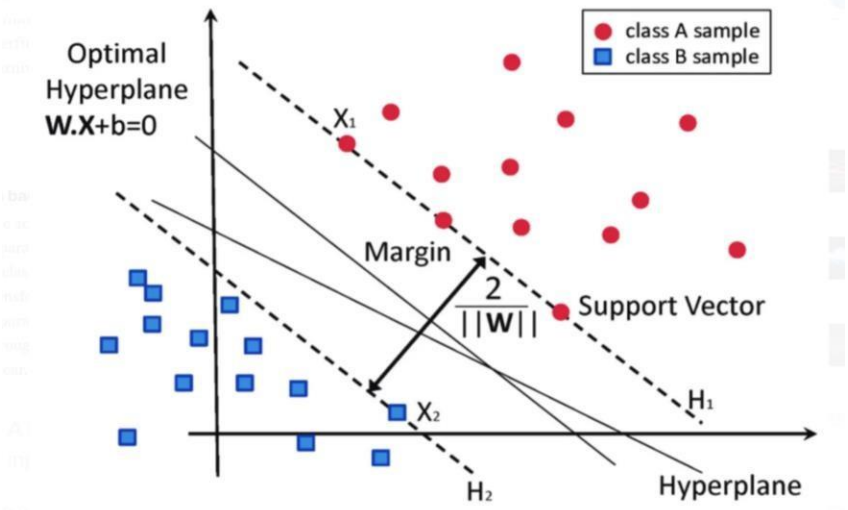


Fig 4.10: DATA CLASSIFICATION USING SVM

MRI data Interpretation using SVM:

Support Vector Machines (SVMs) may be applied to a variety of medical image processing tasks, including the interpretation of data from Magnetic Resonance Imaging (MRI). SVMs may be taught to categorize MRI data into several groups, such as normal and pathological brain tissue, or to forecast clinical outcomes based on MRI properties[88].

SVMs may be employed in the context of MRI data analysis to find subtle patterns and anomalies in MRI images that might not be noticeable to the naked eye. Several MRI sequences, including T1-weighted, T2-weighted, and diffusion-weighted MRI, can provide the input characteristics for SVMs. Many images processing methods, including segmentation, registration, and texture analysis, can be used to extract this information.

The high dimensionality of the data is one of the difficulties when utilizing SVMs for MRI data processing. Millions of voxels might be present in an MRI scan, making it difficult to extract characteristics from such high-dimensional data without overfitting. Principal component analysis (PCA) and linear discriminant analysis (LDA) are two dimensionality reduction approaches that may be used to decrease the number of features and increase the effectiveness of the SVM

algorithm[89].

The restricted availability of labeled data is another difficulty when employing SVMs for MRI data interpretation. The SVM model needs labeled data to be trained, and getting this data from medical professionals can be time and money-consuming[89]. To solve this difficulty, transfer learning techniques may be utilized to exploit pre-trained SVM models on big datasets to enhance the performance of SVMs on small datasets.

When used to analyze MRI data, SVMs can be an effective tool for locating patterns and anomalies that are difficult for human observers to notice. The input features must be carefully chosen, a suitable kernel function and regularization parameter must be selected, and dimensionality reduction and transfer learning techniques must be used to increase the effectiveness and accuracy of the SVM algorithm. All of these steps are necessary to use SVMs for MRI data analysis.

Tuning parameters:

An SVM (Support Vector Machine) model has several tuning parameters that may be adjusted to improve performance. These parameters consist of:

1. Kernal function[90]: The decision border between the two classes is shaped by the kernel function. Linear, polynomial, and radial basis functions are typical kernel functions (RBF). The data and the current issue determine which kernel function should be used. Below fig 4.11, shows the different types of kernels used in svm's.

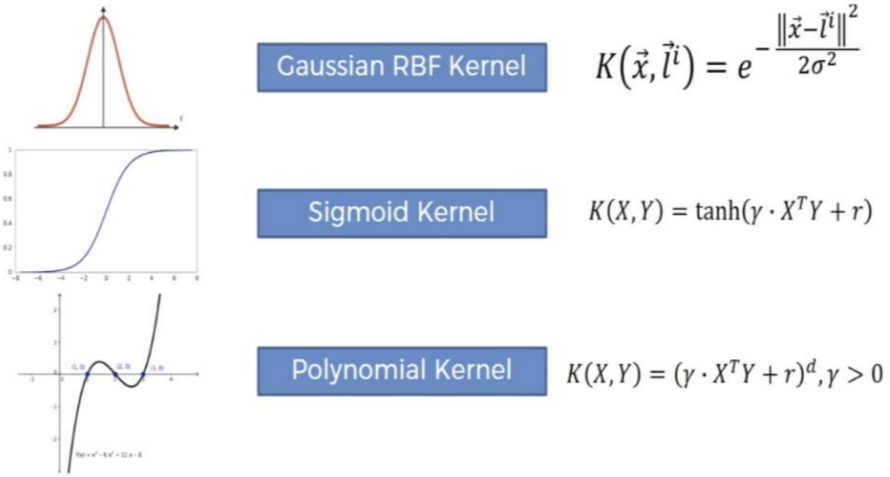


Fig 4.11: TYPES OF KERNALS

2. Regularization parameter(C)[88]: This parameter regulates the trade-off between increasing the margin between the two classes and reducing classification error. Fig 4.12 shows the influence of c in SVM. When C is smaller, the margin is wider and there is greater tolerance for misclassifications, but when C is larger, the margin is narrower and there are fewer misclassifications.

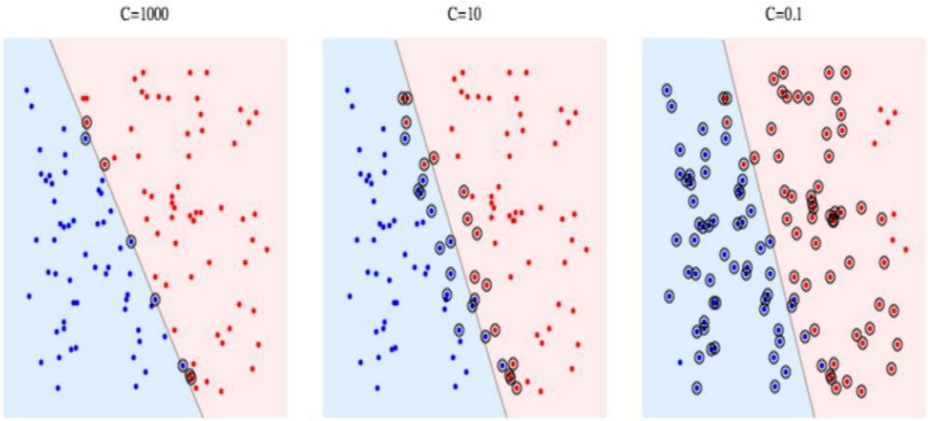


Fig 4.12: INFLUENCE OF C IN SVM'S

3. Kernel coefficient (gamma)[86]: For non-linear kernels like the RBF kernel, the gamma parameter defines the form of the decision border. A higher gamma value results in a more complicated decision border and may cause overfitting, whereas a lower gamma value results in a simpler decision boundary and may cause underfitting. Below fig 4.13 illustrates the gamma usage.

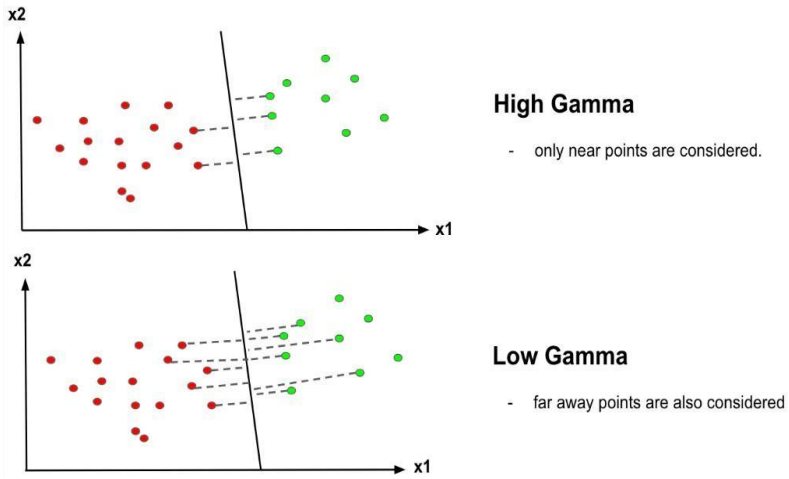


Fig 4.13: IILLUSTRATES OF USING GAMMA

4. Class weights[86]: The class weights may be used to balance each class's contribution to the model. This is helpful when there is an imbalance between the classes, where one class has much more examples than the other.

Using a grid search or a randomized search over a range of parameter values is a popular strategy for tuning these parameters. Using a performance metric like accuracy, precision, recall, or F1 score, the model's performance is assessed on a validation set. The parameter combination with the best performance on the validation set is then chosen.

It is important to remember that these parameters should only be tuned carefully since improperly chosen parameters might cause the model to overfit or underfit. To evaluate the model's performance and make sure it generalizes effectively to new data, cross-validation techniques like k-fold cross-validation can be utilized.

Training SVM Model:

SVM (Support Vector Machines) is a machine learning system that can diagnose Frontotemporal Dementia (FTD) using brain imaging data such as MRI images. SVM is a supervised learning technique that may be trained on labeled data to classify incoming data into one of two categories: FTD or non-FTD.

The SVM algorithm may be trained on brain imaging data from persons with and without FTD in the context of FTD identification. The SVM method may then be used to determine whether new brain imaging data is suggestive of FTD or not. The following steps can be performed to train an SVM model for FTD detection:

1. Data collection and pre-processing [86]: MRI scans are acquired and pre-processed from persons with and without FTD to ensure that they are in an acceptable format for analysis. These might include procedures like skull stripping, normalization, and smoothing.
2. Feature extraction [86]: From the pre-processed MRI images, relevant features are retrieved. Gray matter volume, cortical thickness, and other imaging biomarkers are examples of these characteristics.

3. Data labeling [86]: The clinical diagnoses of the people from whom the MRI scans were taken are used to determine whether or not the MRI scans are suggestive of FTD.
4. Model training [86]: Using a training approach like the Sequential Minimal Optimization (SMO) algorithm, the labeled MRI images are utilized to train an SVM model. The model is optimized by altering hyperparameters such as the kernel function, regularization parameter, and cost function.
5. Model testing and validation [86]: The trained SVM model's performance in identifying FTD is assessed using a different dataset of MRI images. Cross-validation or other statistical techniques can be used to further verify the model's correctness.

4.3 Random Forest

Random forest may be used for both classification and regression problems[91]. It is a member of the family of ensemble learning algorithms, which integrate many models to increase forecasting accuracy and reduce overfitting. The random forest algorithm creates numerous decision trees, each of which is trained on a random portion of the training data and a random subset of the features[92]. With less correlation across the trees, there is less overfitting, and generalization performance is improved.

The algorithm functions as follows[91]:

1. To generate a new training set for each decision tree, a random subset of the training data is sampled with replacement.
2. Each tree is divided into a random subset of features at each node, often using the square root of the total number of features.
3. Using the chosen characteristics, a decision tree is trained on the new training set.
4. A forest of decision trees is created by repeating steps 1-3.
5. Each decision tree in the forest individually predicts the target value during prediction, and the final forecast is formed by taking the majority vote (in classification) or the average (in regression) of the individual tree predictions.

It is possible to utilize MRI pictures to forecast the presence or absence of an

illness or the severity of a disease in the context of MRI data processing. Extracting features from the MRI pictures is the initial step in employing the random forest for MRI data processing[93]. Many methods, including texture analysis, form analysis, and intensity-based analysis, can be used to extract features. The random forest technique may then be applied with these attributes as input.

MRI data with labels indicating the presence or absence of an illness or the severity of a condition may be used to train the random forest algorithm[94]. The algorithm constructs a number of decision trees during training, with each tree being trained using a random portion of the training data and a random subset of the features. All of the decision trees in the forest's forecasts are combined to create the final forecast[93].

The random forest can handle a lot of characteristics, which is typical in MRI analysis, which is one benefit of employing it for MRI data processing. Moreover, the random forest can deal with missing data, which might happen in an MRI study because of errors or partial pictures. The random forest also has the advantage of estimating the relative weights of each feature in the prediction of the target variable [93]. This can aid in determining which MRI findings are most useful for predicting the existence or severity of a disease.

Random forest may be computationally expensive and memory-intensive, especially for big datasets with a lot of features, which is one restriction of employing it for MRI data processing[93]. Moreover, the quality of the feature extraction, which might vary depending on the feature extraction approach selected, affects the quality of the outcomes.

Each decision tree in the forest individually forecasts the target value during prediction, and the final prediction is formed by averaging (in regression) or taking the majority vote (in classification) of the individual tree predictions. Random forest is well-liked because it can handle a lot of information, is resistant to outliers, and can assess the relative contribution of each feature to predicting the target variable.

4.4 Artificial Neural Networks

The structure and operation of the human brain served as the inspiration for

artificial neural networks (ANNs), a form of machine learning method[95]. The neurons that makeup ANNs are linked processing units that cooperate to carry out a certain job. A number of layers are used in an ANN to process data. The output layer generates the final forecast or result, whereas the input layer receives the raw data. One or more hidden layers exist between the input and output layers, doing computations at intermediate levels to change the data into a format that is relevant to the output layer[96].

In an ANN, each neuron gets input from other neurons, processes the information using an activation function, and then generates an output that is sent to further neurons[97]. The weights and biases of the connections between neurons are changed during training so that the ANN can develop accurate prediction abilities. Classification, regression, picture recognition, natural language processing, and many more tasks may be performed with ANNs. They are well-liked because of their capacity to recognize intricate patterns in data, generalize to fresh data, and automatically extract characteristics from unstructured data.

As a group of interconnected processing units known as neurons[97], artificial neural networks (ANNs) analyze incoming data in order to carry out a specified job. An ANN's fundamental operating process may be summed up as follows[96]:

1. Input layer: The input layer is where the raw input data, which may be in the form of numbers, pictures, text, or other sorts of data, is sent.
2. Hidden layers: There may be one or more hidden layers between the input and output layers. These layers carry out intermediary computations to change the input into a format that the output layer can utilize. Many neurons that are coupled to the neurons in the layer above make up each hidden layer.
3. Neuron computation: Each neuron in the ANN takes input from other neurons, processes the information using an activation function, and then generates an output that is sent to further neurons in the network. The activation function decides whether the neuron is activated or not dependent on the input it receives.
4. Weight adjustment: In order for the artificial neural network (ANN) to learn to produce correct predictions, the weights and biases of the connections between neurons are changed during training. The biases describe the

threshold for activation, whilst the weights show the strength of the connections between neurons.

5. Output layer: Based on the input and the weights and biases of the neurons in the hidden layers, the output layer generates the final prediction or output. The result might take the form of a probability distribution, a regression value, or a classification.
6. Error calculation: Calculating the error or loss involves comparing the output of the ANN to the real output. Backpropagation is then utilized to change the weights and biases of the network's neurons based on the mistake.
7. Repeat: Repeat until the ANN performs satisfactorily on the training data: The method is repeated for a number of iterations or epochs.

In general, ANNs learn by changing the biases and weights of the network's neurons in response to input and intended output. The network may then make predictions on fresh, unforeseen data using the learned weights and biases.

4.5 Segmentation Tools

MRI segmentation tools are used to separate or segment certain anatomical components or areas of interest from medical images produced by magnetic resonance imaging (MRI)[98]. These technologies use a range of algorithms and approaches to automatically or partly identify and isolate certain tissues or structures within the MRI data. Here are some common MRI segmentation tools:

FreeSurfer

The Athinoula A. Martinos Institute for Biomedical Imaging's Laboratory for Computational Neuroimaging created FreeSurfer, a widely used software for analyzing structural MRI data[99]. FreeSurfer offers tools for image segmentation, cortical surface reconstruction, and volumetric analysis of brain regions. It is automated and semi-automatic[100].

Both tools for volumetric analysis of various brain areas as well as tools for cortical surface reconstruction are included in FreeSurfer. These tools produce a comprehensive three-dimensional model of the cortex. The foundation of FreeSurfer's cortical surface reconstruction technique is a deformable surface

model, which builds a precise model of the cortical surface using the boundaries between various tissue types in the MRI data. It is possible to see and examine the thickness, curvature, and other characteristics of the recreated cortical surface.

FSL (FMRIB Software Library)

The FMRIB Software Library (FSL) is a software library developed by the FMRIB (Functional MRI of the Brain) group at the University of Oxford that includes image processing and statistical tools for functional, structural, and diffusion MRI brain imaging data[101].

To prepare MRI data for further analysis, FSL offers a variety of tools for image segmentation, registration, and normalization. FSL's segmentation techniques can detect and label several tissue types in MRI data, such as gray matter, white matter, and cerebrospinal fluid[101]. FSL also offers a number of registration and normalization tools for aligning MRI data from various participants or time points to a common space.

These tools can be used to investigate differences in brain function between groups, as well as changes in brain function over time.

SPM (Statistical Parametric Mapping)

Building and evaluating geographically extended statistical processes is known as statistical parametric mapping[102], and it is used to evaluate hypotheses regarding functional imaging data. SPM, a piece of free and open-source software, is an implementation of these concepts.

The analysis of brain imaging data sequences has been incorporated into the SPM software package. The sequences might be a succession of photographs from separate cohorts, or time-series from the same topic. The current version is meant for the study of fMRI, PET, SPECT, EEG and MEG.

ITK-SNAP

ITK-SNAP is a multi-platform, free, open-source software program used to separate out structures in 3D and 4D biological pictures. It was initially created by

student teams at the University of North Carolina under the direction of Guido Gerig (NYU Tanden School of Engineering)[98], who had the idea of an instrument that would be simple to use and have a constrained feature set focused solely on the job of picture segmentation.

ITK-SNAP offers manual delineation, picture navigation, and semi-automatic segmentation utilizing active contour approaches. ITK-SNAP provides a user-friendly interface for segmenting medical pictures manually and partially automatically. Its semi-automatic segmentation capabilities segment anatomical structures using algorithms based on active contour models and level-set techniques. Users can manually alter segmentations using the brush and region-growing tools.

VBM8 (Voxel-based morphometry)

VBM8 is a popular toolset used in neuroscience research for the analysis of structural MRI data (Voxel-based morphometry)[103]. The segmentation module of the VBM8 toolbox uses a combination of tissue probability maps, spatial priors, and spatial normalization to divide MRI images into several tissue classes, including grey matter, white matter, and cerebrospinal fluid[104]. This toolset has been used in this thesis for segmenting the frontal and temporal lobs of the brain from MRI data.

The segmentation module of VBM8 is built on the Unified Segmentation approach, a probabilistic model that divides MRI images based on a combination of tissue intensity, spatial, and contextual data[105]. The segmentation module in VBM8 additionally comes with a number of other functionalities, including:

1. Image inhomogeneity may be corrected using the non-parametric non-uniformity normalization (N3) approach in VBM8, which can also handle image inhomogeneities in MRI data.
2. Automated bias field correction: The VBM8 software has a bias field distortion correction algorithm that can automatically correct the MRI data for bias field distortions.
3. Partial volume estimation: VBM8 has a method for partial volume estimation that can calculate the percentage of various tissue types in each voxel. This approach is valuable for evaluating minute changes in the composition of

tissues.

Overall, the segmentation module in VBM8 provides a robust and flexible tool for segmenting MRI data into different tissue classes, which can be useful for analyzing structural changes in the brain associated with neurological and psychiatric disorders.

5 Implementations

MRI is a sophisticated medical imaging tool that can provide high-resolution brain pictures, and ML algorithms may be used to extract relevant information from these images. Machine learning techniques such as SVM, CNN, ANN, and Random Forest may all be used to analyze MRI data. These machine-learning methods have been proven to be successful in processing MRI data and have been utilized in a range of applications, including disease diagnosis, brain segmentation, and feature selection. And in this chapter, the implementation process of all these Ml techniques is briefly outlined.

5.1 CNN Model Implementation

The multiple 3D Convolutional neural networks used to categorize FTD and Alzheimer's disease and examine the data utilized in this work are described in detail in this part of the thesis. The categorization of FTD disease using the few-shot learning approach and the training procedure for each of the CNN models are both covered in this chapter. With fewer training samples at first, we had to determine the most effective techniques for classifying Alzheimer's and FTD disease.

The actual implementation began with modifying the pre-processed ADNI dataset for the classification of Alzheimer's disease and feature extraction for the few-shot learning approach. Three models for Alzheimer's disease were created using transfer learning and varied network configurations. The extracted relevant features from trained CNNs are used to few-shot classification challenges. Furthermore, three distinct trials were carried out to classify FTD disease. All the created models for both diseases have been tested using Sk-learn metrics and the gold standard assessment approach in machine learning "Cross-Validation".

5.1.1 Data collection

This study's primary prerequisite is 3D MRI data. The choice to include 3D MRI scan data was made so that the models can learn which brain areas are damaged by certain disorders. In identifying these disorders, the damaged brain areas are referred to as the region of interest (ROI). The Alzheimer's Disease Neuroimaging Initiative (ADNI) Database was used to obtain the data used to train 3D CNN models (https://adni.loni.usc.edu/). The Frontotemporal Labour Degeneration Neuroimaging Initiative Database supplied the data samples for the FSL task that was performed downstream (NIFTD). In this study, we only employed MRI to classify the FTD disease.

5.1.2 Data Pre-processing

The 3D MRI scan data used in this thesis was obtained from datasets made accessible to the public by the Frontotemporal Neuroimaging Initiative and the Alzheimer's Disease Neuroimaging Initiative (NIFTD). The main objective of the openly accessible databases for ADNI and NIFTD is to research clinical, imaging, biochemical biomarkers, and genetics that would aid in the early detection and progression of AD and FTD disorders. The ADNI dataset has undergone conventional pre-processing procedures because MRI scans were gathered from several scanner platforms. Using T1 3D MPRAGE techniques tailored to each individual 3T MRI scanner, ADNI-GO/-2 MRI data were obtained.

The segmentation toolbox of VMB8 is used to automatically segment the 3D MRI scans into 1.5mm isotropic voxel-sized partitions of the gray matter, white matter, and cerebrospinal fluid during the first stage of preprocessing. These white matter and gray matter are high-dimensionally registered to an aging-specific reference template using the DARTEL algorithm. Fig: 5.1 shows the basic pre-processing steps that have been carried out on the ADNI dataset.

A final reference template is utilized to wrap the gray matter segments in order to preserve the entire amount of gray matter (GM) before wrapping, and voxel values were modified for the volumetric changes brought on by the normalization. The visual examination of gray matter maps, as a final step, confirms the segmentation and registration accuracy.

Both the ADNI and NIFTD datasets were used in this thesis. The ADNI database

contains the Frontotemporal Neuroimaging Initiative (NIFTD) MRI scan, which likewise goes through the same pre-processing stages. Processing 662 MRI scans from the ADNI dataset are broken down into three categories: normal cognition (NC), late mild cognitive impairment (LMCI), and Alzheimer's disease (AD).

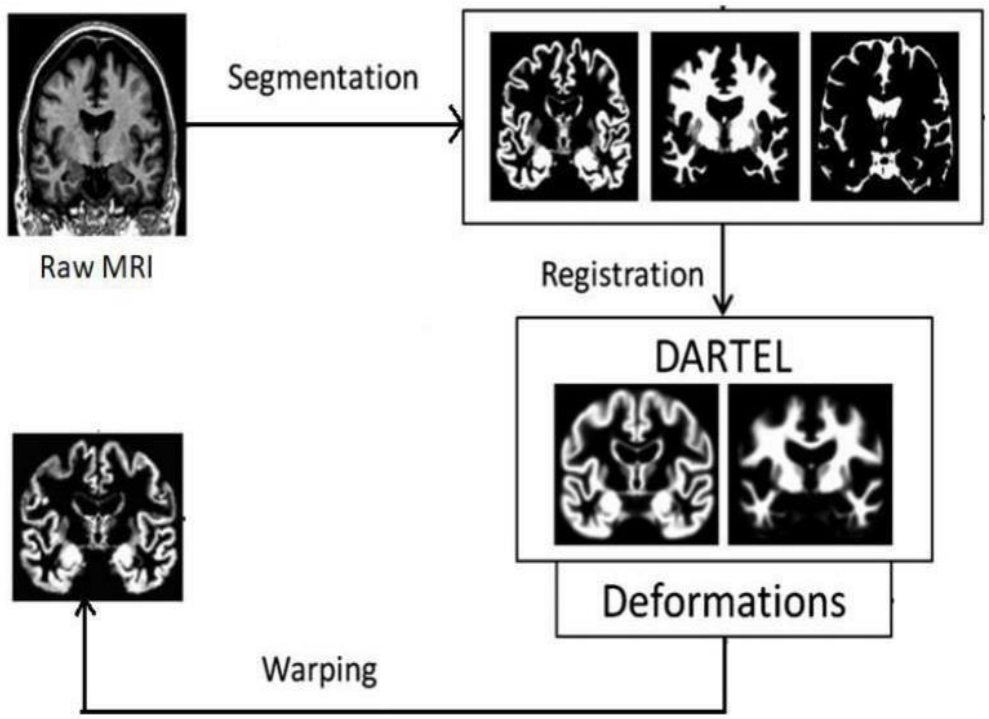


Fig.5.1: Data Pre-processing that has been carried out on ADNI datasets

Similar to this, the NIFTD dataset comprises 279 MRI images in total, which are divided into categories for frontotemporal dementia (FTD) and normal cognition (NC) scans.

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

We used a total of 530 MRI scans to train the Alzheimer's disease model. For the validation of the trained model, we used 66 data samples while a test set of 67 MRI scans have been used for the 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.

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

The computational Anatomy toolbox (CAT12) was used to preprocess the 3D MRI scans for this work, and the SPM12 tool was used for statistical mapping. The DARTEL algorithm is used to normalize segmented gray and white matter pictures to the default CAT12 brain template in Montreal Neuroimaging Institute (MNI) reference space.

This is then resliced into 1.5mm isotropic voxels, with modulation used to regulate tissue expansion and shrinkage. The pre-processed image is utilized to build the models and is converted for model inputs.

Fig.5.2: 3D AD MRI SCAN OF ADNI DATASET

5.1.3 3D CNN models

Three 3D CNN models were built in this study for categorizing AD, and one finished model feature would be utilized for embedding learning. All three CNNs were created using Pytorch's Video ResNet-18 pre-trained model. The three ADNI models are being built in order to create prior knowledge for the downstream FSL job of classifying FTD disease using fewer training data.

ADNI CNN Model 1:

With the exception of the pre-trained ResNet-18 architecture, the initial model is being created without the usage of pre-trained weights. A maximum pooling layer, batch normalization, and ReLU activation are placed after each convolution layer in the 3D Video ResNet-18 pre-trained model. The last completely linked layer, which is in charge of classification, is coupled to each of the four convolution layers. Six hundred twenty-two data samples in total have been utilized, of which 540 are training samples, 66 are validation samples, and 67 are saved for testing the trained model. Epochs: 10, Batch size: 10, Learning rate: 0.001, Optimizer: SGD, and Loss Function: Cross-entropy is the model hyperparameters that were utilized during model training. Loss and accuracy curves of the trained 3D CNN model without pre-trained weights for the ADNI model-1 is shown in fig:5.3.

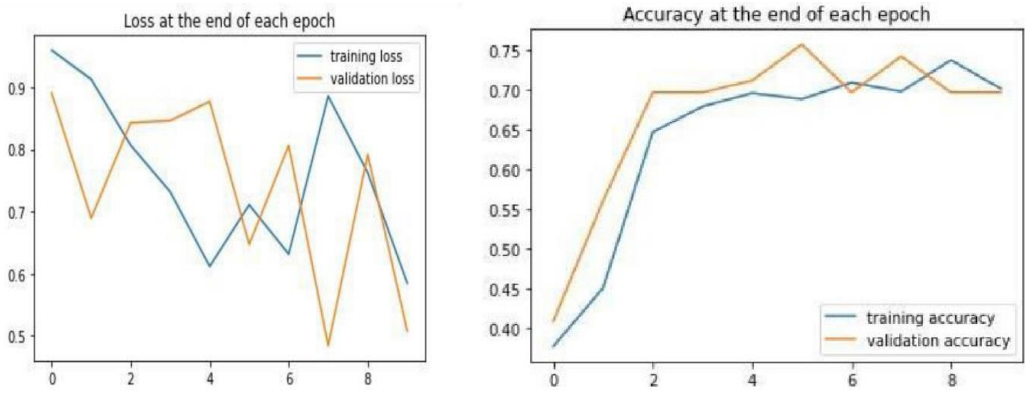


Fig.5.3: LOSS AND ACCURACY CURVES OF THE TRAINED 3D CNN MODEL WITHOUT PRE-TRAINED WEIGHTS

ADNI CNN Model 2:

The pre-trained weights of the video ResNet-18 model were used to create the second 3D CNN model. The feature extraction approach of transfer learning was utilized in this 3D CNN model, with just the classification layer weights changed and the convolution base weights fixed. A total of 622 data samples were used, with 540 being training samples, 66 being validation samples, and 67 being saved for testing the trained model.

The following model hyperparameters were utilized during training: epochs: 10, batch size: 10, learning rate: 0.001, optimizer: SGD, and loss function: cross-entropy. Loss and accuracy curves of the trained 3D CNN model with pre-trained weights of ResNet-18 model for ADNI model-2 is shown in figure:5.4.

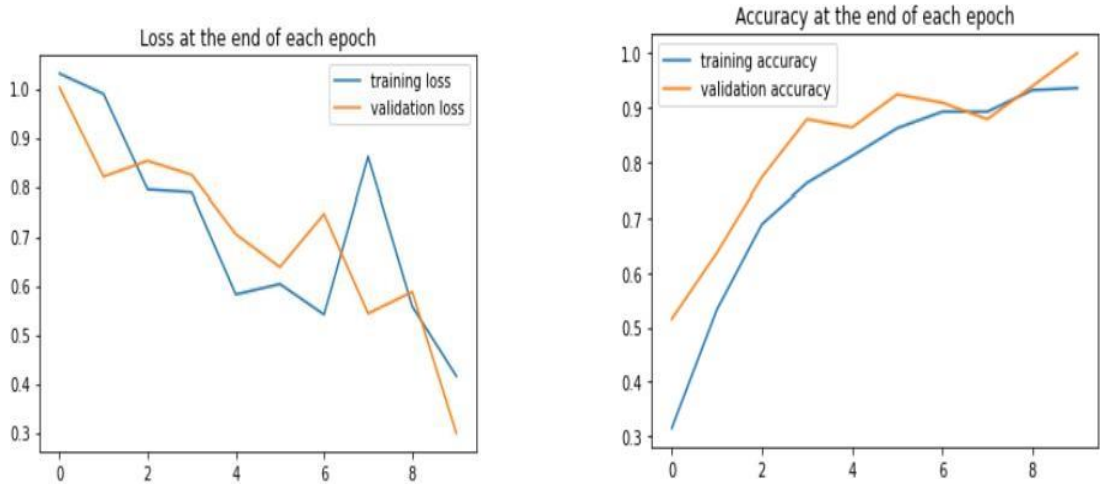


Fig.5.4: LOSS AND ACCURACY CURVES OF THE TRAINED 3D CNN MODEL WITH PRE-TRAINED WEIGHTS FOR ADNI MODEL-2

ADNI Model 3

The pre-trained weights of the video ResNet-18 model were used to create the third 3D CNN model. The remaining convolutional layer weights in this 3D CNN model were locked while the weights of the second-to-last layer and the fully connected layer of the Video ResNet-18 model were changed. This is a fine-tuning approach of transfer learning. A total of 622 data samples were collected, of which 540 were used for training, 66 for validation, and 67 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.

The feature embedding procedure for classifying FTD illness will leverage these ADNI3 fine-tune model features. We can categorize the FTD samples with less training data thanks to the ADNI3 fine-tune model's successful classification outcomes. This model offers a solid transferable feature that helps with the downstream diagnosis of FTD diseases.

For the meta-testing or downstream work, the suggested technique of FSL calls for the best possible previous information. The ADNI fine-tuning model 3 in this study offers the best previous knowledge. Loss and accuracy curves of the trained 3D CNN model with Pre-trained weights for ADNI model-3 is shown in fig:5.5.

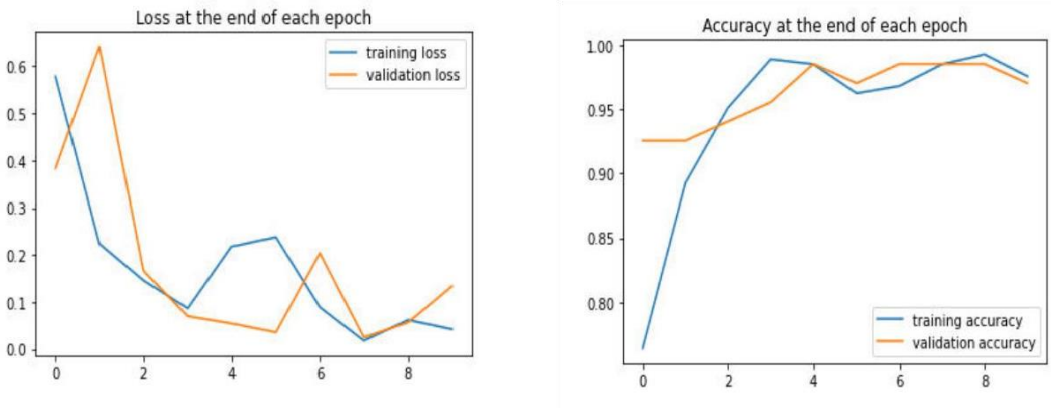


Fig.5.5: LOSS AND ACCURACY CURVES OF THE TRAINED 3D CNN MODEL WITH PRE-TRAINED WEIGHTS FOR ADNI MODEL-3

5.1.4 Features Embedding

The attributes of the ADNI fine-tune model must be incorporated with the NIFTD data samples in order to carry out the task of frontotemporal dementia categorization at this stage. At the average pooling layer stage, learned features from the ADNI fine-tune model are extracted; the classification-related fully connected layer is omitted. We developed the logistic regression model for the diagnosis of FTD sickness after integrating the ADNI features with the NIFTD data sets.

The logistic regression model was trained using samples from our support set that includes NIFTD and ADNI embedding characteristics. The logistic model is further investigated on the query set, which also contains the embedding features of the ADNI and NIFTD data sets. Figure 5.6 demonstrates the whole process of FTD illness classification via features embedding.

5.1.5 FSL Experimentations

To achieve multi-classification of FTD tasks with fewer training samples, we conduct three FSL experiments. To train the logistic regression, also known as the Support set, we arrange the training set with 4 classes of data samples from the NIFTD dataset. The test set, often referred to as the query set, includes 4 classes of samples from the NIFTD dataset for the logistic regression evaluation.

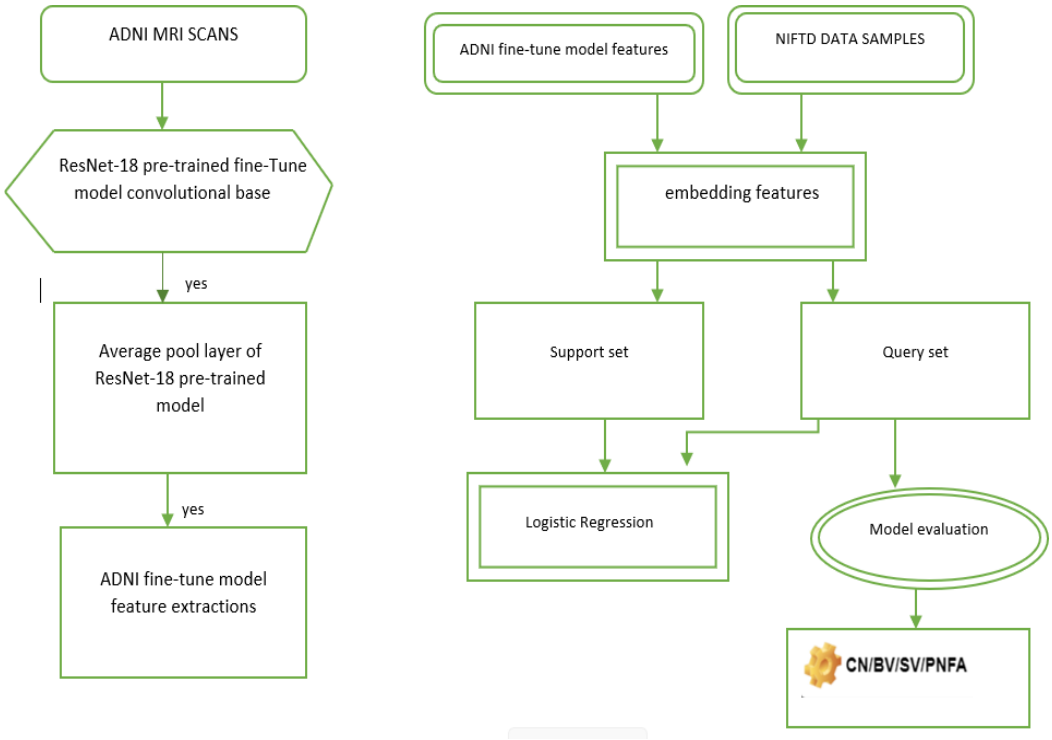


Figure.5.6: OVERALL PROCESS OF FEATURE EMBEDDING OF ADNI FEATURES AND NIFTD DATA SAMPLE

Baseline Multi-classification Model

The trained model is required to carry out the binary classification task in the baseline model setup (CN vs FTD). To train the logistic regression, also known as the upport set, we organize the training set with two classes (CN vs FTD) of data samples from the NIFTD dataset. The query set, also known as the test set, includes two classes of samples from the NIFTD dataset for the logistic regression evaluation. In this assignment, the embedded characteristics of the ADNI and FTD data samples are used to train the logistic regression model. 80% (223) NIFTD data samples make up the training or support set for logistic regression, while 20% NIFTD data samples make up the held-out test set or query set.

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		

10-Shot 4-Way Multi-classification Model

In the NIFTD dataset's 10-Shot (10 samples per class) 10-way (4 classes: CN, SV, BN, and PNFA) multiclassification model, the logistic regression model is trained using just 40 data samples from the support set, and the trained model is assessed using 12 samples from the query set. The embedded characteristics of the data samples from the ADNI and FTD are used to train the logistic regression model in this challenge. The model runs for 1000 iterations, and the samples that were incorrectly categorized are penalized using the L2 regularization parameter. The trained model is assessed using the f1-score, precision, recall, and cross-validation, the gold standard assessment method. Using fewer training examples per class, the model diagnoses several FTD illness types in this test.

NIFTD Data Samples 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		

5.2 SVM and Random Forest Implementation

SVM and random forest are the supervised algorithms used in this portion of the thesis to achieve FTD classification using MRI data. These two strategies have been demonstrated to be particularly well-suited for tasks involving MRI data categorization. The goal of this implementation is to assess the efficacy of these two algorithms and establish an adequate FTD diagnosis in order to provide medical practitioners with a useful tool.

5.2.1 Data collection

The dataset, which was acquired at the Frontotemporal Lobar Degeneration Neuroimaging Initiative (FTLDNI), included MRI data of 279 people, which were split into two classes: 146 FTD patients with the GRN Thr272fs mutation (FTD-GNR+) 1 and 133 normal cognitive, gender-matched healthy volunteers (oHC). The goal of the Frontotemporal Lobar Degeneration Neuroimaging Initiative (FTLDNI), a cooperative research project, is to enhance frontotemporal lobar degeneration diagnosis, treatment, and comprehension (FTLD). The FTLDNI, which unites researchers from several institutions and areas to gather and exchange information on FTLD patients, is funded by the National Institute on Aging (NIA). The effort focuses on using neuroimaging technologies, including as MRI and PET, to conduct studies on structural and functional alterations related to FTD.

5.2.2 Data Pre-processing

The obtained data are then put through a series of pre-processing steps, including Image pre-processing, Intensity normalization, Feature extraction, and Feature scaling, to ensure consistent picture size and format, increase classification accuracy, and provide a better understanding of the data. Images from a 3D dataset display human brains devoid of the skull's framework and encircled by a sizable area of empty voxels. This section may be safely eliminated from each image because it is clearly worthless for classification.

After locating the unnecessary region in each image, the intersection between the images was really removed. Eventually, each image was reduced to only its intersection, resulting in a dataset sample of every size. Finally, in order to ensure the same relevance for all the characteristics (voxels), the dataset has been standardized. Standard Scaler, Robust Scaler, and Min-Max Scaler (MMS) are three distinct scalers that have been examined in our thesis;

Standard Scaler (SS): Assume that X is a matrix with all of the data samples in the columns and all of the features in the rows, and μ and σ are the mean and standard deviation, respectively, of each feature. After removing the mean and scaling all the data to one unit of variance, the standard scaler does the following:

$$X' = \frac{X - \mu}{\sigma}$$

Robust Scaler (RS) : let M be the median observation for each feature inside the dataset, and define p75 and p25 as the respective 75th and 25th percentiles. Then the robust scaler eliminates the median and applies a scaling mechanism dependent on the interquartile range, such as:

$$X' = \frac{X - \mu}{p75 - p25}$$

Min-Max Scaler (MMS): Let $Xmin$ and $Xmax$ represent the lowest and highest values, respectively, for each feature in the dataset. The dataset is then scaled between 0 and 1 by the Min-Max scaler using:

$$X' = \frac{X - Xmin}{Xmax - Xmin}$$

In order to examine the performance effect of a system's dimensionality reduction, Principal Component Analysis (PCA) has also been used. PCA may be used to isolate the most important characteristics (or "components") from the input data and then highlight the main patterns from a vast quantity of data. All of the accuracy numbers were still collected without the use of PCA because it led to more unstable outcomes in our studies.

5.2.3 Experiment Analysis

At the start of our trials, a grid search hyper-parameter optimization is conducted to determine the optimal values for the hyper-parameters of both methods. For SVM, C = 0.1, 1, 10, 100, 1000, and У= 1, 0.1, 0.01, 0.001, 0.0001 were taken into consideration, while linear, polynomial, and sigmoid functions were evaluated as the kernel.

We examined the influence on the algorithmic training process for RF by varying the number of decision trees from 200 to 2000 at 10-step intervals, the maximum depth of the trees from 10 to 109 levels at 11-step intervals, and the minimal number of samples needed to have a decision split (2, 5, 10) and a leaf (1, 2, 4).

The grid-search method is based on a 5-fold cross-validation that employed accuracy for both SVM and RF algorithms as reference metric. Once the optimal

hyper-parameters have been discovered, a Stratified Shuffle Split (SSS) technique with 100 splits has been chosen for validation: it randomly separates data into train and test sets, by keeping the percentage of distribution

of the data between the train and test. By doing this, we enable the same number of patients and controls in the final dataset for each of the 100 split rounds. As a detail, we highlight that our experiments have been implemented using the Python programming language, using the features that the scikit-learn library offers.

5.3 Models Developing Environment and Tools

Google Collab:

Google Colab is a cloud-based platform for running and sharing interactive computing environments in the Jupyter notebook style. It offers 12 GB of RAM and 64 GB of free storage. Without having to install any software on their local PC, users may run and edit Python code in a web browser. With access to several open-source frameworks, such as TensorFlow, PyTorch, and Keras, Google Colab offers a free computational resource (GPU and TPU) to run the code.

The platform is enormously beneficial for machine learning tasks that demand a lot of processing power since it provides free access to GPUs and TPUs (Tensor Processing Units), which may accelerate the computation times. Users may also add new packages and libraries to their environment by using pip or conda.

Pytorch:

The open-source machine learning framework PyTorch is created and maintained by Facebook. PyTorch is an all-inclusive framework for creating deep learning models, a subset of machine learning that's frequently used to tasks like language processing and picture identification. Most machine learning developers can easily pick it up and use it because it was written in Python. In addition to using reverse-mode auto-differentiation, which permits on-the-fly modification of computation graphs, PyTorch is notable for its outstanding support for GPUs. Since it can be quickly tested and prototyped, it is a common option.

In comparison to static networks used in other frameworks, PyTorch's dynamic computational graph enables more flexible and effective processing, making it one

of the platform's distinguishing advantages. This makes it simpler to debug and test out various model designs since PyTorch may modify the computational graph as the data travels through the network.

The relevant libraries used in this thesis are Sk-learn, Seaborn, and metaplot lib for data visualization, as well as NumPy for array calculations, nibabel for reading and writing MRI scans, pandas for organizing the data into data frames, and others.

Keras:

A high-level deep learning API called Keras is developed in Python and is based on TensorFlow, CNTK, or Theano. It is user-friendly, modular, and extendable and was created with the goal of enabling quick experimentation with deep neural networks. Convolutional neural networks (CNNs), recurrent neural networks (RNNs), and other deep neural networks can be built and trained using Keras' straightforward and user-friendly interface.

It provides a variety of pre-built layers and models, as well as tools for visualization and debugging, that may be utilized to create unique structures.The simplicity of usage of Keras is one of its main advantages. Users may concentrate on the network's high-level design by focusing on the library's abstraction of many of the lower-level intricacies of creating and training deep neural networks.

SKlearn

Sklearn is a well-known Python package for machine learning. It includes tools for data pre-processing, model selection, and evaluation, as well as a diverse set of supervised and unsupervised machine learning algorithms.

Scikit-learn is a popular option for data scientists and machine learning practitioners because it is simple to use and effective. It also provides a variety of tools for choosing and assessing models, including cross-validation, hyperparameter tweaking, and metrics for gauging model effectiveness.

Moreover, scikit-learn offers data preparation methods including feature scaling, categorical variable encoding, and missing data management. Moreover, it provides tools for feature selection and dimensionality reduction, enabling users

to extract the most pertinent information from their data.

6 Results and discussion

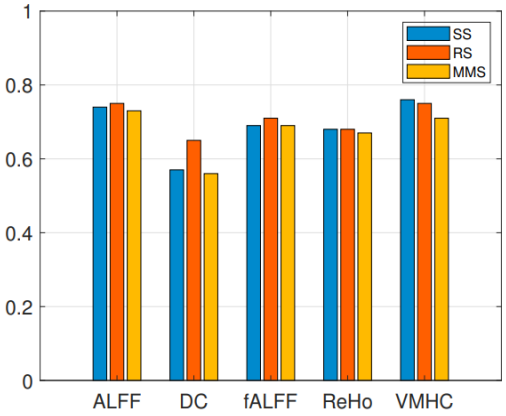
This chapter of thesis, explains the results gained after successfully implementing the ML models on the ADNI and NIFTD datasets.

6.1 SVM and Random Forest Results

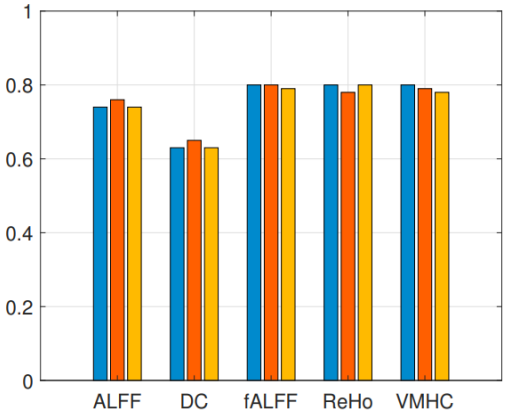
The results of the SVM algorithm's execution offer an intriguing window into the data present in each voxel of fMRI pictures. The accuracy performance for each MRI functional parameter is reported in Fig. 2a, which also displays the outcomes of the various feature standardization techniques we took into consideration for our studies. Fig. 2c displays the corresponding mean and standard deviation accuracy to help the reader better understand the variation across the standardization approaches.

Secondly, with just slight variations in approaches, it is feasible to see consistent findings across the various standardization processes. With a standard deviation of 6%, the average accuracy of the SVM approach on the entire dataset for all functional parameters and scalers is equal to 69%. Looking more closely at the findings produced by the SVM approach, it can be seen that the ALFF and VMHC strategies perform best at 74% of mean accuracy, with highest accuracy being recorded on robust scalers (75% for ALFF) and standard scalers (76% for VMHC).

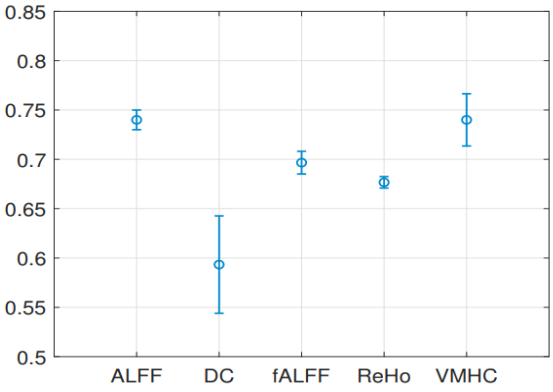
To add further perspective to the data, the classification accuracy obtained on the other MRI functional characteristics did not go over 71%. Intriguingly, the outcome on fALFF (71% with the robust scaler, with an average accuracy across all scalers of 70% and 1% standard deviation) is worse than that observed on ALFF, indicating that the search for the ideal hyperplane to distinguish patients and controls in the dataset appears to be complicated by the exclusion of the typical noise connected to mental states such as hunger from the MR images.



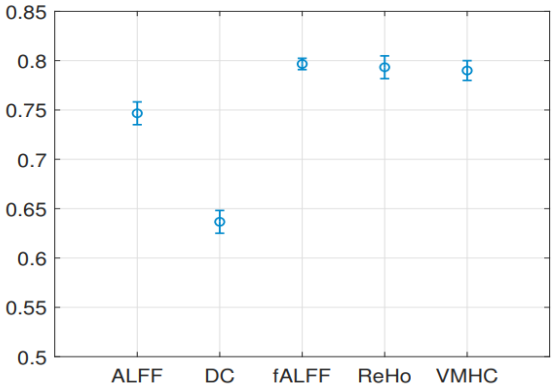
(a) SVM accuracy for each MRI functional parameter for the distinct feature standardization methods.



(b) RF accuracy for each MRI functional parameter for the distinct feature standardization methods.



(c) SVM evaluation for each MRI functional parameter (mean accuracy and standard deviation).



(d) RF evaluation for each MRI functional parameter (mean accuracy and standard deviation).

It's interesting to note that this pattern is seen on each of the three data scalers used. Below, the table shows the accuracy that we gained through this algorithm. After comparing the results on various scalers, it can be shown that the robust methodology beats the standard and min-max procedures, with an average accuracy across all MRI functional parameters of 71% as opposed to 69% and 68%, respectively.

The SVM method performs better when the dataset is developed to account for outliers in the data, therefore this conclusion is unquestionably intriguing. Locating all the voxels in a space with a mean of 0 and a fixed standard deviation in an attempt to create some distortion into the dataset turns out to be less successful than robust scaling. The same holds true for the min-max technique's effort to

scale the data while maintaining their original structure.

	ALFF	DC	fALFF	ReHo	VMHC
Standard scaler	0.740	0.570	0.690	0.680	0.760
	0.740	0.630	**0.800**	**0.800**	**0.800**
Robust Scaler	0.750	0.650	0.710	0.680	0.750
	0.760	0.650	**0.800**	0.780	0.790
Min-Max Scaler	0.730	0.560	0.690	0.670	0.710
	0.740	0.630	0.790	**0.800**	0.780
Mean	0.740	0.593	0.697	0.677	0.740
	0.747	0.637	0.797	0.793	0.790
Standard deviation	0.010	0.049	0.012	0.006	0.026
	0.012	0.012	0.006	0.012	0.010

SVM (top) and RF (bottom) accuracy for each MRI functional parameter for the distinct feature standardization methods. The mean and standard deviation are also reported

6.2 Conventional Neural Network Results

The thesis's objective is to classify frontotemporal dementia using fewer training samples, hence the findings of the experiments carried out in the previous chapter are reported and shown in this chapter. To achieve this thesis objective, many experiment implementation strategies and methodologies have been used. The results of the ADNI model, which were obtained utilizing transfer learning models, are first presented and discussed in the chapter. The chapter next analyses the frontotemporal dementia categorization findings utilizing the few-shot learning approach. The outcomes of the three ADNI models that were used are detailed below. We analyze the outcomes of the three experiments utilizing a few-shot learning approach in the latter section of the chapter.

6.2.1 ADNI Model 1 Results

The Video ResNet-18 pre-trained network architecture is being used to create the ADNI model 1 without the pre-trained weights. 67 test data samples from the ADNI were used to assess the model using PyTorch evaluation routines. 3 fold cross-validation approaches are also used to assess the model's consistency in terms of output. For the multi-classification job (AD/LMCI/CN), the ADNI model 1 is designed.

ADNI test samples	Test accuracy	Phi-coefficient
67	0.73	0.68

Without utilizing the pre-trained weight, the ADNI model 1's results reveal that the model misclassifies a significant portion of the data. The two samples were incorrectly categorized as AD by the CN class.

Furthermore, the AD class incorrectly predicted the CN class for the nine samples. Eleven samples in all were incorrectly identified by the model, which is considered a major misclassification in our particular scenario. Here, we must achieve the model's ideal classification performance in order to classify the FTD disease in our subsequent challenge.

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

6.2.2 ADNI Model 2 Results

The ADNI model 2 was created utilizing the pre-trained weights of Video ResNet-18 and the feature extraction approach of transfer learning. Only the final fully connected layer weights are permitted to change, with the convolutional base weights of the pre-trained model being frozen.

67 test data samples from the ADNI were used to assess the model using PyTorch evaluation routines. The consistency of the findings is further assessed using the 3-fold cross-validation approach on the model. For the multi-classification job (AD/LMCI/CN), the ADNI model 2 is created.

ADNI test samples	Test accuracy	Phi-coefficient

67	0.83	0.73

The ResNet-18 pre-trained weights were included into the ADNI feature extraction model, which enhances the classification performance of the CN and AD classes. A sample of CN was incorrectly predicted to be AD whereas five samples of AD were incorrectly predicted to be CN. Three of the five LMCI samples were incorrectly predicted as CN and two as AD, leading to the misclassification of the five samples. A total of eleven samples were incorrectly identified, therefore the categorization is still not ideal. Because only the last layer of the model is trained using the ADNI dataset, the pre-trained model is more suited to CN and AD classes but not LMCI. Therefore, adding more fine-tuning layers to the pre-trained model will greatly enhance each class's ability to classify data.

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

6.2.3 ADNI Model 3 Results

The pre-trained weights of Video ResNet-18 were used to create the ADNI model 3 along with the transfer learning fine-tuning technique. The last fully connected layer weights as well as the second-to-last layer of the pre-trained model's convolutional layer 4 are permitted to update. 67 test data samples from the ADNI were used to assess the model using PyTorch evaluation routines. The consistency of the findings is further assessed using the 3-fold cross-validation approach on the model. For the multi-classification job (AD/LMCI/CN), ADNI model 3 is created.

ADNI test samples	Test accuracy	Phi-coefficient
67	0.98	0.97

The ResNet-18 pre-trained weights were used in the ADNI fine-tune model, which improved the classification performance of all classes in the ADNI dataset. A CN sample was misclassified and projected to be AD. The fine-tuning of the pre-trained model makes it more flexible to all dataset classifications. As a result, the model attains the necessary categorization performance. As a result, the subsequent task of classifying FTD disease will make use of these ideal fine-tune model features to achieve high classification performance with less training data.

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

6.2.4 Comparison of ADNI Models

To track the model classification performance, we used three distinct ADNI model parameters for the multi-classification task. All of the ADNI models increase classification performance. Without utilizing the pre-trained weights of the ResNet-18 pre-trained model, the ADNI model 1 performs well in classification. It demonstrates that the networks are learning relevant characteristics from the dataset of ADNI MRI images. In the ADNI model 2, which was created using the ResNet18 model's pre-trained weights. The model results outperform model 1 in terms of classification performance, indicating that the pre-trained weights are maximizing classification performance.

Finally, there is the ADNI model 3, which is a fine-tuned version in which the final convolution layer is likewise trained and its weights are adjusted. The model achieves optimum classification performance. ADNI models were being built to

generate prior knowledge for the downstream task of identifying the FTD disease. All model results are listed 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

6.2.5 Feature Embedding Model results

The downstream job is carried out utilizing the embedding learning approach of the few-shot learning methodology. We decided that the fine-tuned model would be utilized as prior information for the downstream task to classify the FTD disease. The most significant component in achieving strong classification accuracy on downstream jobs is the quality of the features. We incorporated the NIFTD data samples with the characteristics of a fine-tuned ADNI model and the feature space of the training set is expanded in this technique by incorporating ADNI fine-tune model features.

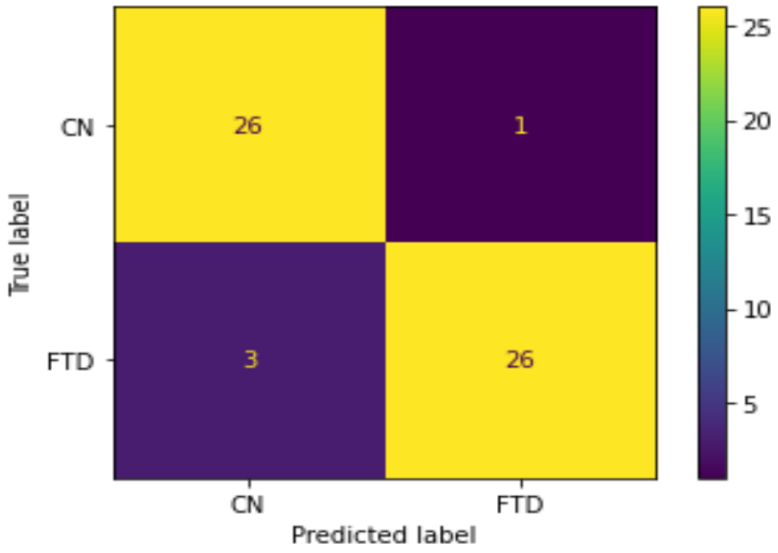
Baseline Experiment

The result of the binary classification problem (CN vs FTD) utilizing the embedding learning approach is discussed in this portion of the chapter. The execution processes and assessment outcomes of this experiment have previously been discussed in earlier chapters. According to the table below, the model achieves good classification performance. We show further findings on several assessment criteria and the depiction of confusion matrices.

Only 4 misclassifications were recorded in the baseline trial, which demonstrated strong classification ability. However, the baseline experiment made use of all of the NIFTD data samples to examine the efficacy of embedding learning.

NIFTD Query set samples	Accuracy score	Phi-coefficient
56	0.92	0.85

In order to train the logistic regression model using the embedding learning approach, we then conduct the experiments in a few-shot learning fashion with somewhat fewer training examples.



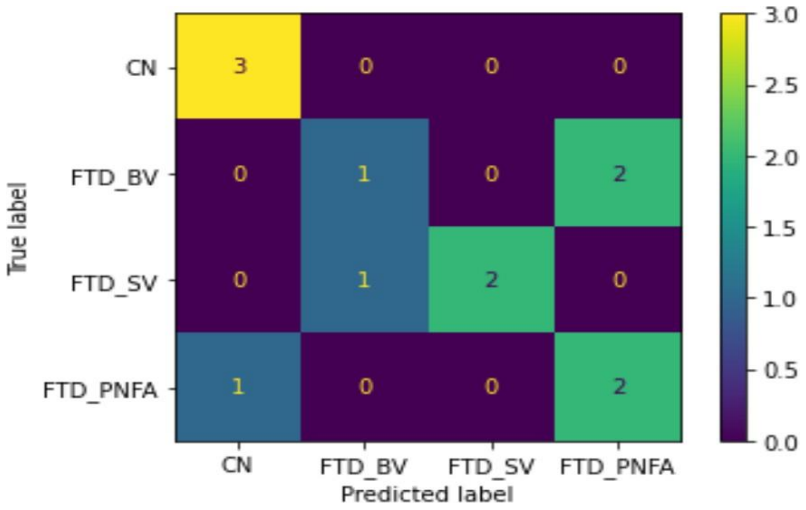
5-Shot 4-Way Multi-classification Model

In this 5-short (5 samples per class), four-way (CN, BV, SV, PNFA) FSL experiment, the logistic regression model is trained using a support set of 5 samples per class in a total of 20 training samples, and the trained model is evaluated using a query set of just 12 samples (3 samples per class). With fewer training samples, this entire task setting is responsible for classifying FTD illness. Samples from the query set and the support set have past information about the ADNI fine-tune model. We also give some more findings in the form of a confusion matrix and on several assessment measures.

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

5-SHORT 4-WAY MULTI-CLASSIFICATION EVALUATION RESULTS USING EMBEDDING LEARNING METHODOLOGY

The results of the 5-short 4-way multi-classification indicate that the model incorrectly assigned the samples to the BV, SV, and PNFA classes. The support set's small training size, which was used to train the logistic regression model, is what led to the misclassification. We just needed 20 training data to produce a substantial classification performance of the model, excluding misclassification. The embedding of ADNI fine-tune model features allowed for the classification of FTD data sets with less training samples, which was a noteworthy accomplishment.



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

10-Shot 4-Way Multi-Classification Model

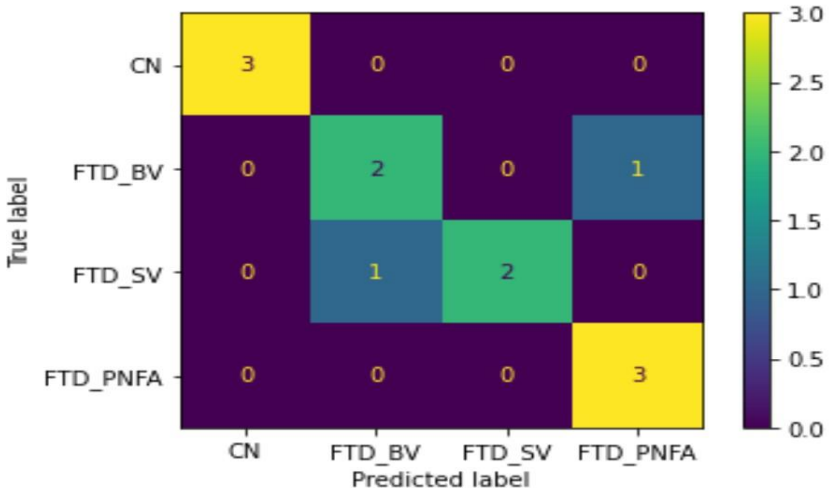
The support set consisted of 10 samples per class with a total of 40 samples responsible for training the logistic regression task and the query set consisted of only 12 samples (3 samples per class) which were responsible for evaluating the logistic regression model in this 10-shot (10 samples per class) 4-way (CN, BV, SV, PNFA) multi-classification task of FSL. This entire job setting was in charge of

classifying FTD illness with less training samples. Both the support set and the query set samples have previous knowledge about the ADNI fine-tuned model. We give some more findings on the various assessment measures as well as in the confusion matrix format.

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

10-SHORT 4-WAY MULTI-CLASSIFICATION EVALUATION RESULTS USING EMBEDDING LEARNING METHODOLOGY

The results of the 10-shot 4-way multi-classification demonstrate that the model misclassified the samples in the BV and SV classes. The misclassification is caused by the support set's small training size, which is used to train the logistic regression model.



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

Apart from misclassification, the model obtains substantial classification performance with only 40 training data when compared to 5-shot 4-way multi-classification. The results of this experiment demonstrate that increasing the number of training samples or shots improves the model's classification performance. Yet, even with fewer training examples, the model obtains high classification performance, with just two misclassifications.

Comparison of FSL Experiment Results

To track model classification performance, we used three distinct model settings for the multi-classification challenge. All tests yield relevant findings, allowing us to test the durability of our methods. The baseline experiment results suggest that our embedding learning technology has the ability to classify the FTD disease even when our dataset size is modest. The 5-shot 4-way multi-classification problem demonstrates that by incorporating ADNI fine-tune model prior knowledge with less training samples, the model may gain adequate classification performance. Ultimately, a 10-shot 4-way classification trial yields the best classification performance for FTD illness with less training data. The model prediction performance improves as the number of shots increases, as evidenced by our FSL experiment results.

Experiments	5-fold cross validation mean accuracy	Standard deviation of the model
Baseline experiment	0.93	0.02
5-shot 4-way multiclassification	0.60	0.19
10-shot 4-way multiclassification	0.75	0.16

COMPARISON OF ALL FSL EXPERIMENTS RESULTS USING EMBEDDING LEARNING METHODOLOGY

7 Conclusion

This last chapter of thesis documentation presents the whole findings based on different exploration strategies to achieve the thesis goal of finding a better approach for precise FTD detection. 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 previous chapters, 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, ADNI model 2 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% ±1). After the fine-tuning of ADNI model 3, 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% ± 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, with the help of finetuning 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 experimentation results, that increasing the shots the prediction performance also increase. The phenomenon 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.

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 highly 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] S. A. Gale, D. Acar, and K. R. Daffner, "Dementia.," *Am J Med*, vol. 131, no. 10, pp. 1161–1169, Oct. 2018, doi: 10.1016/j.amjmed.2018.01.022.

[2] M. P. Aranda *et al.*, "Impact of dementia: Health disparities, population trends, care interventions, and economic costs," *J Am Geriatr Soc*, vol. 69, no. 7, pp. 1774–1783, Jul. 2021, doi: 10.1111/JGS.17345.

[3] V. Hachinski, "Dementia: new vistas and opportunities.," *Neurol Sci*, vol. 40, no. 4, pp. 763–767, Apr. 2019, doi: 10.1007/s10072-019-3714-1.

[4] H. A. Ball *et al.*, "Functional cognitive disorder: dementia's blind spot.," *Brain*, vol. 143, no. 10, pp. 2895–2903, Oct. 2020, doi: 10.1093/brain/awaa224.

[5] S. Tiwari, V. Atluri, A. Kaushik, A. Yndart, and M. Nair, "Alzheimer's disease: Pathogenesis, diagnostics, and therapeutics," *Int J Nanomedicine*, vol. 14, pp. 5541–5554, 2019, doi: 10.2147/IJN.S200490.

[6] S. Tiwari, V. Atluri, A. Kaushik, A. Yndart, and M. Nair, "Alzheimer's disease: Pathogenesis, diagnostics, and therapeutics," *Int J Nanomedicine*, vol. 14, pp. 5541–5554, 2019, doi: 10.2147/IJN.S200490.

[7] L. Raposo Rodríguez, D. J. Tovar Salazar, N. Fernández García, L. Pastor Hernández, and Ó. Fernández Guinea, "Magnetic resonance imaging in dementia.," *Radiologia*, vol. 60, no. 6, pp. 476–484, Nov. 2018, doi: 10.1016/j.rx.2018.04.003.

[8] K. M. Connor and J. R. T. Davidson, "Development of a new Resilience scale: The Connor-Davidson Resilience scale (CD-RISC)," *Depress Anxiety*, vol. 18, no. 2, pp. 76–82, 2003, doi: 10.1002/da.10113.

[9] Malaz Boustani *et al.*, *Screening for Dementia*.

[10] L. Clare *et al.*, "Goal-oriented cognitive rehabilitation for early-stage alzheimer's and related dementias: The GREAT RCT," *Health Technol Assess (Rockv)*, vol. 23, no. 10, pp. 1–244, Mar. 2019, doi: 10.3310/hta23100.

[11] A. Atri, "The Alzheimer's Disease Clinical Spectrum: Diagnosis and Management," *Medical Clinics of North America*, vol. 103, no. 2, pp. 263–293, Mar. 2019, doi: 10.1016/j.mcna.2018.10.009.

[12] Z. Arvanitakis and D. A. Bennett, "What Is Dementia?," *JAMA*, vol. 322, no. 17, p. 1728, Nov. 2019, doi: 10.1001/jama.2019.11653.

[13] B. D. James and D. A. Bennett, "Causes and Patterns of Dementia: An Update in the Era of Redefining Alzheimer's Disease," *Annu Rev Public Health*, vol. 40, pp. 65–84, Apr. 2019, doi: 10.1146/annurev-publhealth-040218-043758.

[14] M. D. Geschwind, "Rapidly progressive dementia," *CONTINUUM Lifelong Learning in Neurology*, vol. 22, no. 2, Dementia, pp. 510–537, Apr. 2016, doi: 10.1212/CON.0000000000000319.

[15] M. N. Rossor, N. C. Fox, C. J. Mummery, J. M. Schott, and J. D. Warren, "The diagnosis of young-onset dementia," *Lancet Neurol*, vol. 9, no. 8, pp.

793–806, Aug. 2010, doi: 10.1016/S1474-4422(10)70159-9.

[16] A. Kumar, J. Sidhu, A. Goyal, J. W. Tsao, and J. Svercauski, "Alzheimer Disease (Nursing)," *StatPearls*, 2021, Accessed: Mar. 01, 2023. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/33760564

[17] M. W. Bondi, E. C. Edmonds, and D. P. Salmon, "Alzheimer's disease: Past, present, and future," *Journal of the International Neuropsychological Society*, vol. 23, no. 9-10 Special Issue, pp. 818–831, Oct. 2017, doi: 10.1017/S135561771700100X.

[18] P. Scheltens *et al.*, "Alzheimer's disease," *The Lancet*, vol. 397, no. 10284, pp. 1577–1590, Apr. 2021, doi: 10.1016/S0140-6736(20)32205-4.

[19] J. L. Cummings, G. Tong, and C. Ballard, "Treatment Combinations for Alzheimer's Disease: Current and Future Pharmacotherapy Options," *Journal of Alzheimer's Disease*, vol. 67, no. 3, pp. 779–794, 2019, doi: 10.3233/JAD-180766.

[20] N. N. Naseri, H. Wang, J. Guo, M. Sharma, and W. Luo, "The complexity of tau in Alzheimer's disease," *Neurosci Lett*, vol. 705, pp. 183–194, Jul. 2019, doi: 10.1016/j.neulet.2019.04.022.

[21] P. B. Gorelick, S. E. Counts, and D. Nyenhuis, "Vascular cognitive impairment and dementia," *Biochim Biophys Acta Mol Basis Dis*, vol. 1862, no. 5, pp. 860–868, May 2016, doi: 10.1016/j.bbadis.2015.12.015.

[22] C. Iadecola *et al.*, "Vascular Cognitive Impairment and Dementia: JACC Scientific Expert Panel," *J Am Coll Cardiol*, vol. 73, no. 25, pp. 3326–3344, Jul. 2019, doi: 10.1016/j.jacc.2019.04.034.

[23] C. Iadecola, "The Pathobiology of Vascular Dementia," *Neuron*, vol. 80, no. 4, pp. 844–866, Nov. 2013, doi: 10.1016/j.neuron.2013.10.008.

[24] Z. Walker, K. L. Possin, B. F. Boeve, and D. Aarsland, "Lewy body dementias," *The Lancet*, vol. 386, no. 10004, pp. 1683–1697, Oct. 2015, doi: 10.1016/S0140-6736(15)00462-6.

[25] T. Yousaf, G. Dervenoulas, P. E. Valkimadi, and M. Politis, "Neuroimaging in Lewy body dementia," *J Neurol*, vol. 266, no. 1, Jan. 2019, doi: 10.1007/s00415-018-8892-x.

[26] S. N. Gomperts, "Lewy body dementias: Dementia with lewy bodies and Parkinson disease dementia," *CONTINUUM Lifelong Learning in Neurology*, vol. 22, no. 2, Dementia, pp. 435–463, Apr. 2016, doi: 10.1212/CON.0000000000000309.

[27] J. Bang, S. Spina, and B. L. Miller, "Frontotemporal dementia," *The Lancet*, vol. 386, no. 10004, pp. 1672–1682, Oct. 2015, doi: 10.1016/S0140-6736(15)00461-4.

[28] K. Younes and B. L. Miller, "Frontotemporal Dementia: Neuropathology, Genetics, Neuroimaging, and Treatments," *Psychiatric Clinics of North America*, vol. 43, no. 2, pp. 331–344, Jun. 2020, doi: 10.1016/j.psc.2020.02.006.

[29] D. K. Kuruppu and B. R. Matthews, "Young-onset dementia," *Semin Neurol*, vol. 33, no. 4, pp. 365–385, 2013, doi: 10.1055/s-0033-1359320.

[30] B. T. Peet, S. Spina, N. Mundada, and R. la Joie, "Neuroimaging in Frontotemporal Dementia: Heterogeneity and Relationships with Underlying Neuropathology," *Neurotherapeutics*, vol. 18, no. 2, pp. 728–752, Apr. 2021, doi: 10.1007/s13311-021-01101-x.

[31] N. T. Bott, A. Radke, M. L. Stephens, and J. H. Kramer, "Frontotemporal dementia: diagnosis, deficits and management," *Neurodegener Dis Manag*, vol. 4, no. 6, pp. 439–454, 2014, doi: 10.2217/nmt.14.34.

[32] B. F. Boeve, A. L. Boxer, F. Kumfor, Y. Pijnenburg, and J. D. Rohrer, "Advances and controversies in frontotemporal dementia: diagnosis, biomarkers, and therapeutic considerations," *Lancet Neurol*, vol. 21, no. 3, pp. 258–272, Mar. 2022, doi: 10.1016/S1474-4422(21)00341-0.

[33] W. W. Seeley, "Behavioral variant frontotemporal dementia," *CONTINUUM Lifelong Learning in Neurology*, vol. 25, no. 1, pp. 76–100, Feb. 2019, doi: 10.1212/CON.0000000000000698.

[34] J. C. Morris and B. F. Boeve, "Behavioral Variant Frontotemporal Dementia," *CONTINUUM Lifelong Learning in Neurology*, vol. 28, no. 3, pp. 702–725, Jun. 2022, doi: 10.1212/CON.0000000000001105.

[35] C. R. Marshall *et al.*, "Primary progressive aphasia: a clinical approach," *J Neurol*, vol. 265, no. 6, pp. 1474–1490, Jun. 2018, doi: 10.1007/s00415-018-8762-6.

[36] M. L. Gorno-Tempini *et al.*, "Classification of primary progressive aphasia and its variants," *Neurology*, vol. 76, no. 11, pp. 1006–1014, Mar. 2011, doi: 10.1212/WNL.0b013e31821103e6.

[37] D. C. Tippett, "Classification of primary progressive aphasia: Challenges and complexities," *F1000Res*, vol. 9, 2020, doi: 10.12688/f1000research.21184.1.

[38] N. T. Bott, A. Radke, M. L. Stephens, and J. H. Kramer, "Frontotemporal dementia: diagnosis, deficits and management," *Neurodegener Dis Manag*, vol. 4, no. 6, pp. 439–454, 2014, doi: 10.2217/nmt.14.34.

[39] "Vitamin B12 and cognitive function: An evidence-based analysis," *Ont Health Technol Assess Ser*, vol. 13, no. 23, pp. 1–45, 2013.

[40] N. Seraji-Bzorgzad, H. Paulson, and J. Heidebrink, "Neurologic examination in the elderly," *Handb Clin Neurol*, vol. 167, pp. 73–88, Jan. 2019, doi: 10.1016/B978-0-12-804766-8.00005-4.

[41] G. W. Paulson, "The neurological examination in dementia.," *Contemp Neurol Ser*, vol. 9, pp. 13–33, 1971.

[42] "Neuropsychological testing dementia - Search Results - PubMed." https://pubmed.ncbi.nlm.nih.gov/?term=Neuropsychological+testing+dementia&filter=pubt.booksdocs&filter=pubt.clinicaltrial&filter=pubt.meta-analysis&filter=pubt.review&filter=pubt.systematicreview (accessed Mar. 01, 2023).

[43] C. D. Morgan and L. E. Baade, "Neuropsychological testing and assessment scales for dementia of the Alzheimer's type," *Psychiatric Clinics of North America*, vol. 20, no. 1, pp. 25–43, 1997, doi: 10.1016/S0193-953X(05)70391-9.

[44] Á. P. Sempere, L. Berenguer-Ruiz, M. Lezcano-Rodas, F. Mira-Berenguer, and M. Waez, "Lumbar puncture: Its indications, contraindications, complications and technique," *Rev Neurol*, vol. 45, no. 7, pp. 433–436, 2007, doi: 10.33588/rn.4507.2007270.

[45] H. Shafeeq, "Lumbar Puncture," *A Medication Guide to Internal Medicine Tests and Procedures*, pp. 179–183, Jan. 2021, doi: 10.1016/B978-0-323-79007-9.00039-8.

[46] C. M. Doherty and R. B. Forbes, "Diagnostic lumbar puncture," *Ulster Medical Journal*, vol. 83, no. 2, pp. 93–102, 2014.

[47] T. Yousaf, G. Dervenoulas, P. E. Valkimadi, and M. Politis, "Neuroimaging in Lewy body dementia," *J Neurol*, vol. 266, no. 1, Jan. 2019, doi: 10.1007/s00415-018-8892-x.

[48] A. M. Staffaroni *et al.*, "Neuroimaging in Dementia," *Semin Neurol*, vol. 37, no. 5, pp. 510–537, Oct. 2017, doi: 10.1055/s-0037-1608808.

[49] S. Mahalingam and M. K. Chen, "Neuroimaging in Dementias," *Semin Neurol*, vol. 39, no. 2, pp. 188–199, 2019, doi: 10.1055/s-0039-1678580.

[50] J. D. Oldan, V. L. Jewells, B. Pieper, and T. Z. Wong, "Complete evaluation of dementia: PET and MRI correlation and diagnosis for the neuroradiologist," *American Journal of Neuroradiology*, vol. 42, no. 6, pp. 998–1007, Jun. 2021, doi: 10.3174/ajnr.A7079.

[51] R. Schmidt, D. Havas, S. Ropele, C. Enzinger, and F. Fazekas, "MRI in Dementia," *Neurol Clin*, vol. 27, no. 1, pp. 221–236, Feb. 2009, doi: 10.1016/j.ncl.2008.09.003.

[52] H. Matsuda, "MRI morphometry in Alzheimer's disease," *Ageing Res Rev*, vol. 30, pp. 17–24, Sep. 2016, doi: 10.1016/j.arr.2016.01.003.

[53] K. Zukotynski *et al.*, "PET/CT of dementia," *American Journal of Roentgenology*, vol. 211, no. 2, pp. 246–259, Aug. 2018, doi: 10.2214/AJR.18.19822.

[54] B. J. Burkett, J. C. Babcock, V. J. Lowe, J. Graff-Radford, R. M. Subramaniam, and D. R. Johnson, "PET Imaging of Dementia: Update 2022," *Clin Nucl Med*, vol. 47, no. 9, pp. 763–773, Sep. 2022, doi: 10.1097/RLU.0000000000004251.

[55] P. Dupont, "A Role of PET/MR Imaging in Dementia?," *Semin Nucl Med*, vol. 51, no. 3, pp. 296–302, May 2021, doi: 10.1053/j.semnuclmed.2021.01.003.

[56] S. Badillo *et al.*, "An Introduction to Machine Learning," *Clin Pharmacol Ther*, vol. 107, no. 4, pp. 871–885, Apr. 2020, doi: 10.1002/cpt.1796.

[57] L. Baecker, R. Garcia-Dias, S. Vieira, C. Scarpazza, and A. Mechelli, "Machine learning for brain age prediction: Introduction to methods and clinical applications," *EBioMedicine*, vol. 72, Oct. 2021, doi: 10.1016/j.ebiom.2021.103600.

[58] L. Lo Vercio *et al.*, "Supervised machine learning tools: A tutorial for clinicians," *J Neural Eng*, vol. 17, no. 6, Nov. 2020, doi: 10.1088/1741-2552/abbff2.

[59] P. S. Reel, S. Reel, E. Pearson, E. Trucco, and E. Jefferson, "Using machine learning approaches for multi-omics data analysis: A review," *Biotechnol Adv*, vol. 49, Jul. 2021, doi: 10.1016/j.biotechadv.2021.107739.

[60] T. Jiang, J. L. Gradus, and A. J. Rosellini, "Supervised Machine Learning: A Brief Primer," *Behav Ther*, vol. 51, no. 5, pp. 675–687, Sep. 2020, doi: 10.1016/j.beth.2020.05.002.

[61] R. Gupta, D. Srivastava, M. Sahu, S. Tiwari, R. K. Ambasta, and P. Kumar,

"Artificial intelligence to deep learning: machine intelligence approach for drug discovery," *Mol Divers*, vol. 25, no. 3, pp. 1315–1360, Aug. 2021, doi: 10.1007/s11030-021-10217-3.

[62] N. Verbeeck, R. M. Caprioli, and R. van de Plas, "Unsupervised machine learning for exploratory data analysis in imaging mass spectrometry," *Mass Spectrom Rev*, vol. 39, no. 3, pp. 245–291, May 2020, doi: 10.1002/mas.21602.

[63] A. Roohi, K. Faust, U. Djuric, and P. Diamandis, "Unsupervised Machine Learning in Pathology: The Next Frontier," *Surg Pathol Clin*, vol. 13, no. 2, pp. 349–358, Jun. 2020, doi: 10.1016/j.path.2020.01.002.

[64] J. A. Roth, M. Battegay, F. Juchler, J. E. Vogt, and A. F. Widmer, "Introduction to Machine Learning in Digital Healthcare Epidemiology," *Infect Control Hosp Epidemiol*, vol. 39, no. 12, pp. 1457–1462, Dec. 2018, doi: 10.1017/ice.2018.265.

[65] J. G. Greener, S. M. Kandathil, L. Moffat, and D. T. Jones, "A guide to machine learning for biologists," *Nat Rev Mol Cell Biol*, vol. 23, no. 1, pp. 40–55, Jan. 2022, doi: 10.1038/s41580-021-00407-0.

[66] R. Y. Choi, A. S. Coyner, J. Kalpathy-Cramer, M. F. Chiang, and J. P. Campbell, "Introduction to Machine Learning, Neural Networks, and Deep Learning.," *Transl Vis Sci Technol*, vol. 9, no. 2, p. 14, Feb. 2020, doi: 10.1167/tvst.9.2.14.

[67] G. Currie, K. E. Hawk, E. Rohren, A. Vial, and R. Klein, "Machine Learning and Deep Learning in Medical Imaging: Intelligent Imaging," *J Med Imaging Radiat Sci*, vol. 50, no. 4, pp. 477–487, Dec. 2019, doi: 10.1016/j.jmir.2019.09.005.

[68] P. Manimegalai, R. Suresh Kumar, P. Valsalan, R. Dhanagopal, P. T. Vasanth Raj, and J. Christhudass, "3D Convolutional Neural Network Framework with Deep Learning for Nuclear Medicine," *Scanning*, vol. 2022, 2022, doi: 10.1155/2022/9640177.

[69] H. Ma, L. Wang, Y. Chen, and L. Tian, "Convolutional neural network-based artificial intelligence for the diagnosis of early esophageal cancer based on endoscopic images: A meta-analysis," *Saudi Journal of Gastroenterology*, vol. 28, no. 5, p. 332, 2022, doi: 10.4103/SJG.SJG_178_22.

[70] K. Yasaka, H. Akai, A. Kunimatsu, S. Kiryu, and O. Abe, "Deep learning with convolutional neural network in radiology," *Jpn J Radiol*, vol. 36, no.

4, pp. 257–272, Apr. 2018, doi: 10.1007/s11604-018-0726-3.

[71] W. Shan, X. Li, H. Yao, and K. Lin, "Convolutional Neural Network-based Virtual Screening," *Curr Med Chem*, vol. 28, no. 10, pp. 2033–2047, May 2020, doi: 10.2174/0929867327666200526142958.

[72] R. Nirthika, S. Manivannan, A. Ramanan, and R. Wang, "Pooling in convolutional neural networks for medical image analysis: a survey and an empirical study.," *Neural Comput Appl*, vol. 34, no. 7, pp. 5321–5347, Apr. 2022, doi: 10.1007/s00521-022-06953-8.

[73] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology.," *Insights Imaging*, vol. 9, no. 4, pp. 611–629, Aug. 2018, doi: 10.1007/s13244-018-0639-9.

[74] J. Dolz, C. Desrosiers, and I. Ben Ayed, "3D fully convolutional networks for subcortical segmentation in MRI: A large-scale study," *Neuroimage*, vol. 170, pp. 456–470, Apr. 2018, doi: 10.1016/j.neuroimage.2017.04.039.

[75] D. R. Sarvamangala and R. V Kulkarni, "Convolutional neural networks in medical image understanding: a survey.," *Evol Intell*, vol. 15, no. 1, pp. 1–22, Mar. 2022, doi: 10.1007/s12065-020-00540-3.

[76] Purwono Purwono, Alfian Ma'arif, Wahyu Rahmaniar, and Haris Imam, "Understanding of Convolutional Neural Network (CNN): A Review," 2023.

[77] H. E. Kim, A. Cosa-Linan, N. Santhanam, M. Jannesari, M. E. Maros, and T. Ganslandt, "Transfer learning for medical image classification: a literature review," *BMC Med Imaging*, vol. 22, no. 1, Dec. 2022, doi: 10.1186/s12880-022-00793-7.

[78] K. Yasaka, H. Akai, A. Kunimatsu, S. Kiryu, and O. Abe, "Deep learning with convolutional neural network in radiology," *Jpn J Radiol*, vol. 36, no. 4, pp. 257–272, Apr. 2018, doi: 10.1007/s11604-018-0726-3.

[79] V. Stumpo *et al.*, "Machine Learning Algorithms in Neuroimaging: An Overview," *Acta Neurochir Suppl (Wien)*, vol. 134, pp. 125–138, 2022, doi: 10.1007/978-3-030-85292-4_17.

[80] M. Lunt, "Introduction to statistical modelling: Linear regression," *Rheumatology (United Kingdom)*, vol. 54, no. 7, pp. 1137–1140, Jul. 2015, doi: 10.1093/rheumatology/ket146.

[81] Irwan Bello *et al.*, "Revisiting resnets: Improved training and scaling strategies".

[82] Ming Zong, Ruili Wang, Xiubo Chen, Zhe Chen, and Yuanhao Gong, "Motion saliency based multi-stream multiplier ResNets for action recognition".

[83] N. Archana and K. Hareesh, "Real-time Human Activity Recognition Using ResNet and 3D Convolutional Neural Networks," in *2nd International Conference on Advances in Computing, Communication, Embedded and Secure Systems (ACCESS), Ernakulam, India.*

[84] Jaeyoung Choi1, Chaeeun Han, Heeyoon Yang, and Yeonkyoung Hong, "Embedding-based Neural Network Models for Book Recommendation in University Libraries".

[85] Ibon Merino, Jon Azpiazu, Anthony Remazeilles, and Basilio Sierra, "3D Convolutional Neural Networks Initialized from Pretrained 2D Convolutional Neural Networks for Classification of Industrial Parts," 2021.

[86] S. Huang, C. A. I. Nianguang, P. Penzuti Pacheco, S. Narandes, Y. Wang, and X. U. Wayne, "Applications of support vector machine (SVM) learning in cancer genomics," *Cancer Genomics Proteomics*, vol. 15, no. 1, pp. 41–51, Jan. 2018, doi: 10.21873/cgp.20063.

[87] Edgar Osuna, Robert M Freund, and Federico Girosi, "Training Support Vector Machines: an Application to Face Detection," 2000.

[88] Laila Khedher, Javier Ramírez, Juan M Gorriz, and Abdelbasset Brahim, "Automatic classification of segmented MRI data combining Independent Component Analysis and Support Vector Machines," in *Innovation in Medecine and Healthcare 2014, At San Sebastian*.

[89] Durgesh Srivastava and Lekha Bhambhu, "Data classification using support vector machine," *J Theor Appl Inf Technol*, 2010.

[90] P. Mahe and J.-P. Vert, "Virtual Screening with Support Vector Machines and Structure Kernels," *Comb Chem High Throughput Screen*, vol. 12, no. 4, pp. 409–423, Apr. 2009, doi: 10.2174/138620709788167926.

[91] A. Sarica, A. Cerasa, and A. Quattrone, "Random forest algorithm for the classification of neuroimaging data in Alzheimer's disease: A systematic review," *Front Aging Neurosci*, vol. 9, no. OCT, Oct. 2017, doi: 10.3389/fnagi.2017.00329.

[92] W. Hong *et al.*, "Usefulness of Random Forest Algorithm in Predicting Severe Acute Pancreatitis," *Front Cell Infect Microbiol*, vol. 12, Jun. 2022, doi: 10.3389/fcimb.2022.893294.

[93] L. Jiang, S. Sun, J. Chen, and Z. Sun, "Random Forest Algorithm-Based

Ultrasonic Image in the Diagnosis of Patients with Dry Eye Syndrome and Its Relationship with Tear Osmotic Pressure," *Comput Math Methods Med*, vol. 2022, 2022, doi: 10.1155/2022/9437468.

[94] S. Han, B. D. Williamson, and Y. Fong, "Improving random forest predictions in small datasets from two-phase sampling designs," *BMC Med Inform Decis Mak*, vol. 21, no. 1, Dec. 2021, doi: 10.1186/s12911-021-01688-3.

[95] Kenji Suzuki, *ARTIFICIAL NEURAL NETWORKS – ARCHITECTURES AND APPLICATIONS*. 2013.

[96] S.-H. Han, K. W. Kim, S. Kim, and Y. C. Youn, "Artificial Neural Network: Understanding the Basic Concepts without Mathematics.," *Dement Neurocogn Disord*, vol. 17, no. 3, pp. 83–89, Sep. 2018, doi: 10.12779/dnd.2018.17.3.83.

[97] R. Yang and Y. Yu, "Artificial Convolutional Neural Network in Object Detection and Semantic Segmentation for Medical Imaging Analysis.," *Front Oncol*, vol. 11, p. 638182, Mar. 2021, doi: 10.3389/fonc.2021.638182.

[98] A. Virzì *et al.*, "Comprehensive Review of 3D Segmentation Software Tools for MRI Usable for Pelvic Surgery Planning," *J Digit Imaging*, vol. 33, no. 1, pp. 99–110, Feb. 2020, doi: 10.1007/s10278-019-00239-7.

[99] B. Fischl, "FreeSurfer," *Neuroimage*, vol. 62, no. 2, pp. 774–781, Aug. 2012, doi: 10.1016/j.neuroimage.2012.01.021.

[100] P. G. Sämann *et al.*, "FreeSurfer-based segmentation of hippocampal subfields: A review of methods and applications, with a novel quality control procedure for ENIGMA studies and other collaborative efforts," *Hum Brain Mapp*, vol. 43, no. 1, pp. 207–233, Jan. 2022, doi: 10.1002/hbm.25326.

[101] Q. Lv *et al.*, "Effect of Acupuncture on Neuroplasticity of Stroke Patients with Motor Dysfunction: A Meta-Analysis of fMRI Studies," *Neural Plast*, vol. 2021, 2021, doi: 10.1155/2021/8841720.

[102] Á. Friás *et al.*, "Technology-Based Psychosocial Interventions for People with Borderline Personality Disorder: A Scoping Review of the Literature," *Psychopathology*, vol. 53, no. 5, pp. 254–263, Dec. 2020, doi: 10.1159/000511349.

[103] S. S. Keller and N. Roberts, "Voxel-based morphometry of temporal lobe

epilepsy: An introduction and review of the literature," *Epilepsia*, vol. 49, no. 5, pp. 741–757, May 2008, doi: 10.1111/j.1528-1167.2007.01485.x.

[104] L. Palaniyappan, N. Maayan, H. Bergman, C. Davenport, C. E. Adams, and K. Soares-Weiser, "Voxel-Based morphometry for separation of schizophrenia from other types of psychosis in first-episode psychosis: Diagnostic test review," *Schizophr Bull*, vol. 42, no. 2, pp. 277–278, Mar. 2016, doi: 10.1093/schbul/sbv189.

[105] M. Goto *et al.*, "Advantages of Using Both Voxel-and Surface-based Morphometry in Cortical Morphology Analysis: A Review of Various Applications," *Magnetic Resonance in Medical Sciences*, vol. 21, no. 1, pp. 41–57, 2022, doi: 10.2463/mrms.rev.2021-0096.

Appendix A

Description of used Tools

This appendix asserts the modules and libraries that have been utilized for this work. Section 5.3 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/
Numpy	1.22.0	Data visualization	https://numpy.org/
Seaborn	0.11.2	Data visualization	https://seaborn.pydata.org/

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

Supplementary material

The code can be archived from: https://github.com/keerthiravilla/FTD-Detection-using-ML-techniques

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 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

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.

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

FIGURE B. 1 ADNI BINARY CLASSIFICATION MODEL CONFUSION MATRIX

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.

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	076	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

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.

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.

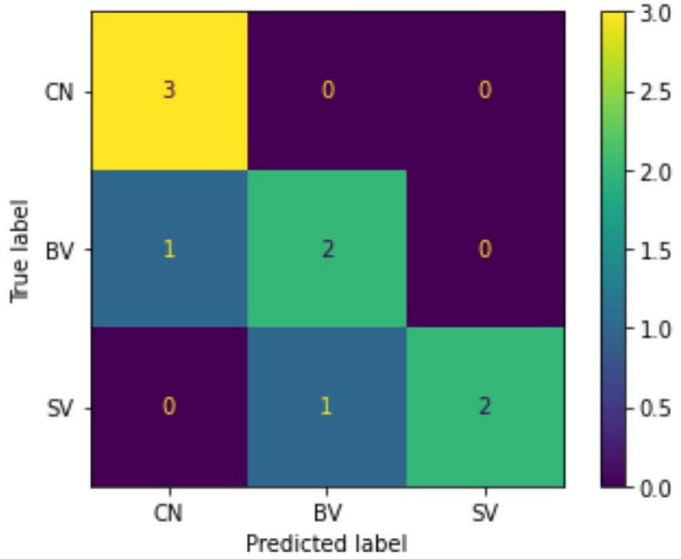


FIGURE B. 2 CONFUSION MATRIX OF THE 5-SHOT 3-WAY MULTI-CLASSIFICATION MODEL

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.