

**REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING**



**DEEP LEARNING APPROACHES FOR ALZHEIMER'S
DISEASE DIAGNOSIS**

18011094 – Toygar Tanyel

SENIOR PROJECT

Advisor
Dr. Ahmet ELBİR

Jun 2023

ACKNOWLEDGEMENTS

I would like to acknowledge and give my warmest thanks to my advisor Dr. Ahmet ELBİR who made this work possible. His guidance and advice carried me through all the stages of writing my project.

Toygar Tanyel

TABLE OF CONTENTS

LIST OF ABBREVIATIONS	v
LIST OF FIGURES	vi
LIST OF TABLES	viii
ABSTRACT	ix
ÖZET	x
1 Introduction to Alzheimer's Disease	1
2 Related Work	4
3 Feasibility Report	6
3.1 Technical Feasibility	6
3.1.1 Software Feasibility	6
3.1.2 Hardware Feasibility	7
3.2 Social Feasibility	8
3.3 Management Feasibility	8
3.4 Legal Feasibility	8
3.5 Economic Feasibility	9
3.6 Time Feasibility	9
4 System Analysis	10
4.1 Use Case Diagram	10
5 System Design	12
5.1 Overview of the Dataset: ADNI	12
5.2 Overview of the Dataset: Kaggle	14
5.3 Data Handling & Preprocessing	15
5.3.1 Flow of Coding (Intuitive)	16
5.3.2 Flow of Coding (Logic-based)	19
5.4 Deep Learning Approaches	20
5.4.1 Traditional Deep Learning: 2D Slice-based	21

5.4.2	3D patch-level CNN	26
5.4.3	ROI-based CNN	27
5.4.4	3D subject-level CNN	28
5.4.5	Vision Transformers	29
6	Implementation	31
6.1	Standardized Implementation: 2D Slice-based	31
6.2	Simulating Wrong Implementations	34
7	Performance Analysis	35
7.1	Standardized Results: 2D Slice-based	35
7.2	Simulating Wrong Analysis	40
8	Discussion	44
8.1	Why is there a need to set particular standards?	44
8.2	Can we extract useful information from wrong approach?	45
8.3	Study limitations	45
9	Conclusion	47
References		48
Curriculum Vitae		52

LIST OF ABBREVIATIONS

ADNI	Alzheimer's Disease Neuroimaging Initiative
AD	Alzheimer's Disease
CN	Cognitively Normal
MCI	Mild Cognitive Impairment
EMCI	Early Mild Cognitive Impairment
LMCI	Late Mild Cognitive Impairment
MRI	Magnetic resonance imaging
CT	Computed Tomography
PET-CT	Positron Emission Tomography and Computed Tomography
ROI	Region of Interest
VOI	Volume of Interest
CNN	Convolutional Neural Network
ViT	Vision Transformers
NLP	Natural Language Processing
VGG	Visual Geometry Group
ResNet	Residual Network
DNN	Deep Neural Network
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

LIST OF FIGURES

Figure 3.1	Gantt Diagram	9
Figure 4.1	Use Case Diagram	11
Figure 5.1	ADNI raw slices.	13
Figure 5.2	Random samples from Kaggle augmented dataset (training set in our wrong simulation study).	14
Figure 5.3	Random samples from Kaggle baseline dataset (test set in our wrong simulation study)	15
Figure 5.4	Sample cmd command file for FreeSurfer.	16
Figure 5.5	a) Automated slice extraction into particular pathway (code piece 2), and b) slices after removing empty pixels (code piece 3)	17
Figure 5.6	Intuitive function for extracting slices from 3D images.	17
Figure 5.7	Skull-Stripping outcomes of the data.	18
Figure 5.8	Logic-based function for extracting slices from 3D images.	19
Figure 5.9	Variety of 3D images through different MRI protocols can be seen.	20
Figure 5.10	a) Automated slice extraction into particular pathway (code piece 2), and b) slices after removing empty pixels and resizing to square with padding (code piece 3).	20
Figure 5.11	VGG Network.	21
Figure 5.12	Fundamental approach of residual networks, and ResNet comparison with plain and VGG19 networks.	22
Figure 5.13	Inception module	23
Figure 5.14	GoogLeNet architecture.	24
Figure 5.15	Xception architecture.	24
Figure 5.16	MobileNet depth-wise and point-wise convolution architecture.	25
Figure 5.17	Visualization of DenseNet architecture.	26
Figure 5.18	ViT, classification model overview [37].	30
Figure 6.1	Procedure 1. Total number of slices.	32
Figure 6.2	Procedure 2. Total number of slices.	32
Figure 6.3	Procedure 3. Total number of slices.	33

Figure 6.4	Procedure 4: Creating custom network and using as feature extractor.	34
Figure 7.1	Procedure 1: Classification of AD, MCI and CN using VGG19.	35
Figure 7.2	Procedure 1: Classification of AD, MCI and CN using InceptionV3.	35
Figure 7.3	Procedure 2: Classification of AD and CN using VGG19.	36
Figure 7.4	Procedure 2: Classification of AD and CN using InceptionV3.	36
Figure 7.5	Procedure 3: Classification of AD and CN using VGG19.	36
Figure 7.6	Procedure 3: Classification of AD and CN using InceptionV3.	36
Figure 7.7	Procedure 4: Classification of AD, MCI and CN using VGG19.	37
Figure 7.8	Procedure 4: Classification of AD, MCI and CN using InceptionV3.	37
Figure 7.9	Procedure 5: Classification of AD and CN using VGG19.	37
Figure 7.10	Procedure 5: Classification of AD and CN using InceptionV3.	37
Figure 7.11	Procedure 6: Classification of AD and CN using VGG19.	38
Figure 7.12	Procedure 6: Classification of AD and CN using InceptionV3.	38
Figure 7.13	Procedure 7: Classification of AD and CN using VGG19.	38
Figure 7.14	Procedure 7: Classification of AD and CN using InceptionV3.	38
Figure 7.15	Procedure 8: Classification of AD and CN using VGG19.	39
Figure 7.16	Procedure 8: Classification of AD and CN using InceptionV3.	39
Figure 7.17	Procedure 9: Classification of AD and CN using custom network and random forest.	39
Figure 7.18	VGG16	40
Figure 7.19	VGG19	41
Figure 7.20	ResNet50	41
Figure 7.21	ResNet101	41
Figure 7.22	Xception	41
Figure 7.23	MobileNet	42
Figure 7.24	MobileNetV2	42
Figure 7.25	DenseNet121	42
Figure 7.26	DenseNet169	42
Figure 7.27	InceptionV3	43

LIST OF TABLES

Table 7.1 Total traning time spend for different deep learning models. . . 40

ABSTRACT

Deep Learning Approaches For Alzheimer's Disease Diagnosis

Toygar Tanyel

Department of Computer Engineering
Senior Project

Advisor: Dr. Ahmet ELBİR

The well-known form of dementia, Alzheimer's disease (AD), is a serious concern in modern healthcare. A loss of cognitive function is the hallmark of AD, a degenerative brain condition that is irreversible and for which there is no known treatment. Recent research efforts in machine learning and deep learning offer groundbreaking experimental solutions using radiomics data and computer vision applications for medical diagnosis, prognosis and treatment planning. However, the phenomenon of explainability requirement of medical data remains an important problem that needs to be overcome. Since 2014, deep learning methods have started to receive significant consideration in research on AD diagnosis, and the volume of publications published is rapidly increasing. The rise in the number of publications also revealed many technical implementation complications and uncertainties. Therefore, approving current literature and providing certain standards gain more importance in order to reduce the technical mistakes made. In this study, the ambiguity involved, have been tried to be explained as clearly as possible with an end-to-end approach from raw 3D MR images that provided by Alzheimer's Disease Neuroimaging Initiative (ADNI) to training and interpretation of results.

Keywords: Alzheimer's disease, deep learning, image classification, medical image processing, end-to-end pipeline, MRI.

ÖZET

Alzheimer Hastalığının Teşhisinin İçin Derin Öğrenme Yaklaşımları

Toygar Tanyel

Bilgisayar Mühendisliği Bölümü

Bitirme Projesi

Danışman: Dr. Ahmet ELBİR

Demansın iyi bilinen bir türü olan Alzheimer hastalığı (AD), modern sağlık dünyasında ciddi bir endişe kaynağı olarak varlığını sürdürmektedir. Bilişsel işlev kaybı, geri dönüşü olmayan ve bilinen bir tedavisi olmayan dejeneratif bir beyin durumu olan AD'nin ayırt edici özelliğidir. Derin öğrenme ve makine öğrenmesindeki son araştırmalar, tıbbi teşhis, прогноз ve tedavi planlaması için bilgisayarlı görü yöntemlerini ve radyomik verilerini kullanarak çığır açan deneySEL çözümler sunmaktadır. Bununla birlikte, tıbbi verilerin açıklanabilirlik gerekliliği olgusu, aşılması gereken önemli bir sorun olmaya devam etmektedir. 2014'ten bu yana, derin öğrenme yöntemleri AD teşhisini araştırmalarında önemli ölçüde dikkate alınmaya başlandı ve yayınlanan yayınların sayısı da hızla artmaktadır. Bu artış, birçok teknik uygulama komplikasyonunu ve belirsizliğini de ortaya çıkarmaktadır. Bu nedenle, yapılan teknik hataları azaltmak için mevcut literatürün onaylanması ve belirli standartların sağlanması daha fazla önem kazanmaktadır. Bu çalışmada Alzheimer's Disease Neuroimaging Initiative (ADNI) tarafından sağlanan ham 3 boyutlu MR görüntülerinden eğitim ve sonuçların yorumlanmasına kadar uçtan uca bir yaklaşımla söz konusu belirsizlikler mümkün olduğunda açık bir şekilde aktarılmasına çalışılmıştır.

Anahtar Kelimeler: Alzheimer hastalığı, derin öğrenme, görüntü sınıflandırma, görüntü veri işleme, uçtan uca model, MRG.

1

Introduction to Alzheimer's Disease

Numerous people throughout the world are afflicted by Alzheimer's disease, a progressive neurological condition that results in cognitive decline and loss of cognitive function. According to ADI statistics, currently more than 50 million people suffering from Alzheimer's disease, and the number is expected to increase annually. The disease manifests as a variety of cognitive problems and can develop till death, since brain cells are destroyed [1]. The symptoms of AD include difficulty in learning new things called amnesia, problems with remembering and mobility, followed by severe speech, literacy, and short/long-term memory impairment. Finally, the patient becomes bedridden and eventually dies since it is difficult for them to accomplish even the most fundamental tasks [2–4]. Uncertainty surrounds the stages of the disease's onset, and different stages of the illness are investigated. Cognitively normal (CN) and moderate cognitive impairment (MCI) are the first phases of progression, followed by AD. Advanced age, family history, and the existence of the ApoE4 gene in the genome are three of the most significant variables affecting AD [5]. Additionally, not all people with mild cognitive impairment develop AD. MCI is a stage of transition, and it has been seen that people with MCI successfully carry out their daily tasks and retain a range of cognitive abilities after being diagnosed. It is estimated that 30% to 40% of patients progress to the next stage within 5 years. After an 18-month observation period, patients were separated into MCI converts and non-converters [6]. Classification distinguishes if disease develops into AD within 18 months. In ADNI database, MCI is labeled as EMCI for not developed MCI patients for a long interval, whereas LMCI represents diagnosed MCI, however, became an AD patient in time. A variety of therapeutic approaches reduce the disease's course and preserve the continuity of brain connections. For cases that result in death, early detection of the condition and understanding the point at which these treatments will begin to work may change the outcome [7]. Analyses and imaging procedures used on the subject can detect the existence of AD, and techniques such the amyloid deposition test can be used as a baseline for an early diagnosis [8]. This approach requires that the disease be in an advanced stage and that the buildup be apparent, which

is achievable in advanced stages [9]. Therefore it must be detected by different clinical, cognitive, and biological methods. Since the detection of amyloid deposition cannot be observed in the early stages, the neuronal loss can be diagnosed by imaging methods, as well as clinical symptoms, neurological tests, clinical dementia grading score, mental state grading score, and various genetic protein markers. However, these diagnostic techniques are challenging, expensive, and time-consuming, and they do not produce the desired outcome quickly [10]. Neurologic and psychological status reports are also examined at in addition to the tests described in the diagnosis section. The subject's medical history, diet, medication use, and prior ailments are all taken into consideration. In order to monitor and diagnose the disease, several imaging techniques, including computerized tomography (CT), magnetic resonance imaging (MR), and positron emission tomography computerized tomography (PET-CT), are used to study the anatomy of the brain and the effects of the disease [11]. In examinations, the structural changes of important areas of the brain such as the temporal lobe, hippocampal area, and frontal lobe are examined. Information about the anatomical structure of the brain, the entire shape of the brain, and information about the vascular structure can all be found in MR scans. This information also aids in the distinction between Alzheimer's disease and other diseases that cause similar symptoms in patients but have different patient changes, such as paralysis. The reduction in size and shrinking of the brain, particularly the loss in folds, are visible on MR images of Alzheimer's patients' brains, which facilitates deep learning in computer vision studies.

Recent research, particularly in the field of neuroimaging, has made possible to collect images with higher resolution and precise information due to the use of high-contrast MR images, which aid in better imaging-based diagnosis. In addition to the subject data gathered, many computer-aided diagnostic tools also enable the classification of AD using subject brain scans. These image characteristics can be derived from numbers such as the ROI of the pertinent subject regions and the VOI of the pertinent volume regions. Making such decisions also heavily relies on input from the hippocampus region, changes in the images, and gray matter information [12]. With the aid of such information, the subject can be categorised as either having AD or not. Additionally, a number of studies have shown in the literature that utilizing MRI data with machine learning, deep learning, and image processing techniques, forecast is feasible to figure out which stage of AD the subject might be in. The research that has been done so far has made substantial advancements in the fields of deep learning and image processing. On medical images, a variety of deep learning techniques are used, and productive results are obtained. However, the increase in publications brings with it a significant amount of technical implementation uncertainties. As a result, the

importance of papers that adhere to a set of standards is growing in order to decrease the frequency of technical errors. In this study, ADNI dataset is utilized for diagnose Alzheimer's disease from 3D MRI.

The rest of this study is organized as follows: Section 2 outlines related works, Section 3 presents feasibility report, Section 4, 5 provides system analysis and design, respectively, Section 6 shows how the project is implemented, Section 7 examines the performance of the results and Section 8, 9 includes discussion, and conclusion.

2 Related Work

Deep learning approaches have been applied in several studies for the early detection and classification of Alzheimer’s disease stages. The approaches involve the use of artificial neural networks to analyze various types of data, such as imaging data, genetic data, and cognitive test scores, to identify patterns and features that may be indicative of the disease.

In [13], the authors employed both autoencoders and 3D CNN models. They created an algorithm that uses MR brain images to identify AD patients. The key technique here is 3D convolution, which is recommended since it performs better than 2D convolution. Layers in this network are pre-trained, but not precisely fine tuned. It is anticipated that the development of fine tuned model would lead to an increase in performance. In another study [14], Sarraf and Tofighi used LeNet-5 to separate AD and normal patient. In [15], Hosseini-Asl et al. detected AD using a deeply supervised adaptive 3D-CNN classifier. Three layered 3D Convolutional Autoencoder (3D-CAE) networks were trained using the CAD-Dementia dataset without any prior preparation for skull stripping. Performance evaluation made use of ten-fold cross-validation. Korolev et al. demonstrated that an equivalent performance is possible [16]. However, 3D CNN applications on 3D images were also bigger than anticipated, and due to the length and complexity of their networks, 3D structural MRI brain scans could not produce satisfactory results. Wang et al. conducted another investigation with an 8-layer CNN [17]. Of these eight layers, two are completely linked classification layers and the other six are for feature extraction. The results showed that LReLU and maximum pooling techniques provided excellent performance, however their results are debatable. Khvostikov et al. distinguished the disease using a CNN model built on a 3D inception [18]. On the other side, this technique was developed utilizing DTI and SMRI and was based on the hippocampus Region of Interests. When the model’s performance was evaluated and applied to the AlexNet architecture, it was discovered that 3D inception performed better. In [19], Sahumbaiyev et al. utilized MRI to study the HadNet architecture for Alzheimer’s disease. For improved estimate,

the data was standardized using the SPM toolbox and skull stripped. It was anticipated that as HadNet evolved, the specificity and sensitivity levels would rise. In [20], Ruiz et al. employed a densely linked 3D CNN structure to 3D MR images in a different investigation. Spasov et al. [5], using a collection of MR images, genetic values, and clinical data, presented the APOe4 model. This model improved model speed, lowered computational complexity, and overcame overfitting when compared to models like AlexNet VGGNet utilized in earlier research. The mathematical PFSCTL model, developed by Jain et al. in [21], which was pre-trained with CNN and VGG-16, served as a feature extract for classification and was supported by the idea of transfer learning. In [22], Ge et al. created a 3D multi-scale CNN model as a novel model for AD diagnosis. Multi scale model feature fusion has been proposed. Song et al. suggested a graph-theoretic tool-based GCNN classification model [23]. They trained and evaluated the network using structural connection graphs that constituted a multi-class model in order to categorize the AD spectrum into four groups. In [24], Liu et al. suggested a multimodel deep learning framework. Their algorithm, which is CNN-based, utilizes MRI to categorize AD patients and automatically segments the hippocampus. CNN Multitask with 3DDCCN Combining learned characteristics improves model categorization. In [25], Parmar et al. created a 3D CNN model to categorize AD into 4 groups using 4D FMRI data. CNN structure was applied again to 3D MRI scans to classify the stage of AD by Basaia et al. in [26].

There are obvious repetition in the wide range of research studies. Due to the absence of implementation details and the lack of publicly available frameworks, the majority of research are hardly replicable. Finally, several of these articles can indicate a biased performance as a result of poor or ambiguous validation or model selection techniques. In 2020, Wen et al. [27] provided overview and reproducible evaluation for convolutional neural networks for classification of Alzheimer's Disease. Their efforts for bridging the gaps in the literature is remarkable.

3

Feasibility Report

In this study, we refer deep learning pipelines which require high hardware and software resources to classify Alzheimer's disease stages.

3.1 Technical Feasibility

This section focuses on the technical resources available to the organization. Organizations can use feasibility report to assess whether their technical resources are adequate and whether their technical staff is capable of turning concepts into functional systems. Evaluation of the proposed system's hardware, software, and other technical requirements is often referred to as technical feasibility.

3.1.1 Software Feasibility

Python is one of the dominant language for artificial intelligence tasks. Due to the substantial open-source contributions made by the Python community, the project will therefore be built using Python and its libraries, which enable users to implement it more easily.

The fundamental libraries used in the project are as follows:

- **NumPy:** NumPy is the essential package for scientific computing with Python. It is possible to define and operate on N-dimensional arrays and matrices.
- **Pandas:** Pandas is a Python library that offers quick and adaptable data structures made to make working with "relational" or "labeled" data simple. Additionally, it is a library with strong data-cleaning filtering features.
- **sklearn:** It is a Python library that contains many algorithms for Machine Learning and allows us to easily manipulate the results of algorithms. In this study, train-test-split function is employed to prepare our dataset.

- **PyTorch:** PyTorch is an open source deep learning library.
- **Tensorflow:** Tensorflow is also an open source deep learning library that can be used where in some implementations are remarkably easier than PyTorch.
- **Matplotlib:** Python's Matplotlib toolkit provides a complete tool for building static, animated, and interactive visualizations. Matplotlib makes image read/write operations easy to use.
- **NiBabel:** This package provides read +/- write access to some common medical and neuroimaging file formats, including NIfTI and DICOM.

Linux is the first operating system to be taken into account because it is free and open-source. Some users do not like Linux because the installation procedure is more complicated and requires more user involvement. Windows, on the other hand, is not strategic due to project owner do not have and accustomed to using MacOS. MacOS operating system is preferred in the project due to its generalizability and easy to use.

Visual Studio Code, Jupyter Notebook, Google Colaboratory and Kaggle Notebooks are environments that available to implement project. These can be used as cross-platforms due to their own special features. IDE choice does not affect the results of the project. Besides these, .csv file format is used to handle with dataset and .txt is employed to create useful utilizations. ".nii" files are handled with python library and no external api is used to open.

3.1.2 Hardware Feasibility

To develop the optimum and sustainable system, one computer with Windows/Linux/MacOS operating system, minimum 16GB RAM memory, 200GB+ disk space, 2080Ti GPU, and Intel Core i7-10700K are required.

However, most parts of the project are developed by the system with Apple M1 chip CPU, Memory: 16 GB Hynix LPDDR4.

Cloud storage options are practical for information storage. However, research groups and organizations may not typically use cloud services for this reason if they have confidential material on their clouds. Thus, to prevent issues, primary disk with 200GB+ space is an ideal choice. System maintenance is possible in both situations.

The training step is implemented using Google Drive due to its seamless data connection with Google Colaboratory. It's easy to store datasets, brain slices, and outcomes on Google Drive. There is no cost until reaching to 15GB Cloud limit.

However, a 100GB package was bought for 5 TL per month in order to complete the project¹. Additionally, Google Colaboratory's GPU and TPU utilization time limits exist. According to Google, notebooks are powered by connections to virtual computers with maximum lives of up to 12 hours². When kept idle for too long, notebooks will also disconnect from virtual machines. The behavior of the maximum VM lifetime and the idle timeout may change over time or depending on your usage. In order for Colab to provide free computational resources, this is necessary.

3.2 Social Feasibility

Users of the system could include technical professionals, research teams, or medical staff. The system provides a user-friendly code interface. Researchers and those who are simply curious have the opportunity to use the system in another way by contributing their original ideas.

3.3 Management Feasibility

The sustainability and development of codes and models can be provided by the hospital's technical team or by independent medical research groups.

3.4 Legal Feasibility

Basic system requirements license payments, server rental/purchase costs, and the costs of the personnel who will use this system will all be paid through an official channel, and all necessary legal duties will be met.

The Alzheimer's Disease Neuroimaging Initiative (ADNI) database provided the information required to realize this project (adni.loni.usc.edu).

The system has been set up in a distinctive, goal-oriented structure where it is capable of carrying out its own fundamental tasks. If the system requires specific permits, licenses, or structural adjustments because it violates an ethical agreement they can be provided, modified and corrected.

¹<https://one.google.com/about/plans?hl=tr>

²<https://research.google.com/colaboratory/faq.html>

3.5 Economic Feasibility

Since the Python software and packages used in the project are free, the project does not represent a legal duty. At the same time, open-source data from Kaggle, which is once again free, and Python & libraries can offer licenses that allow visualizations without any financial obligation for everyone. The only paid requirement is google drive package: Basic 100GB, 57.99TL/year, billed annually.

3.6 Time Feasibility

An undergraduate student worked on the project. The project was implemented as following Gantt Chart/Diagram. Time was managed by initially planning the course of events, then literature review and finding the proper dataset were done prior to the interim report. Subsequently, data were preprocessed and trained on various deep learning models. Ultimately, the results were evaluated and final report was written.

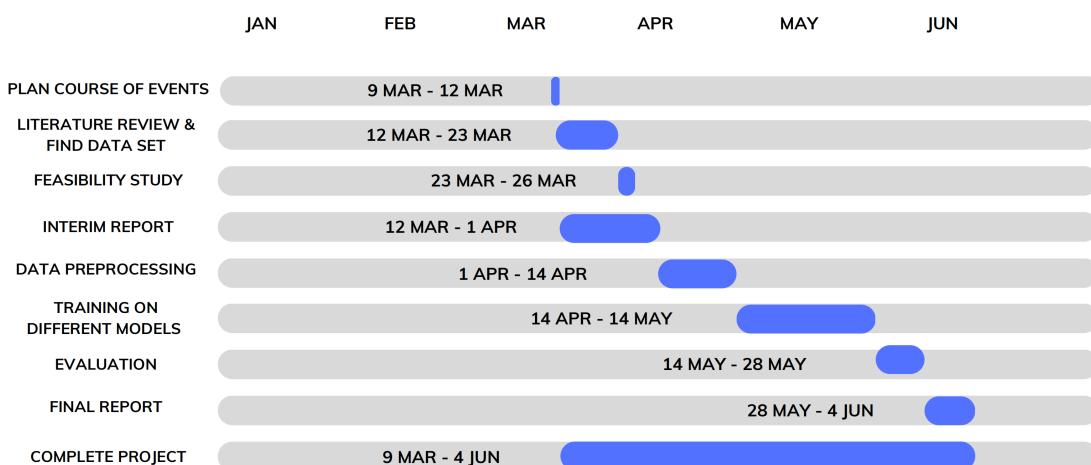


Figure 3.1 Gantt Diagram

4

System Analysis

Deep learning systems provide a tremendous amount of convenience in issues, where human cognitive and physical abilities are not sufficient. However, there is a major hurdle to medical deep learning. Even if it is used only to support the decisions of clinicians and physicians in diagnosis, prognosis and treatment planning, the system should be reliable and explainable.

To underline importance of reliability, explainability and creating testing standards, this project also aims to analyze current literature, explaining old/new technologies and apply some of these deep learning approaches.

Computer scientists, clinicians and patients are the main subjects of our system. The tools prepared by computer scientists must be convenient to domain requirements (i.e. breast cancer, brain tumors and Alzheimer's disease are quite different). Doctors/clinicians are also need to comprehend that why relevant tool is beneficial, since it will have an impact on their decisions. Moreover, around the world, patients almost always expect an explanation for what is going to happen to them. Therefore, the systems that clinicians utilized, must be reliable and explainable with minimal doubt.

4.1 Use Case Diagram

In common systems, first, computer scientist seeks for a path using given data. The data might include genetic information, radiomics or 3D MR images. Then, scientist will be needed to decide the approach (e.g. deep learning, machine learning). Subsequently, more detailed methodologies (e.g. ViT, slice-based CNN) are investigated, since the deep learning is chosen. In some cases, machine learning may provide more reliable and fast outcomes. Eventually, computer scientist finds an outcome to diagnosis, prognosis or/and treatment planning. Ultimately, using the outcomes, clinician can decide next treatment.

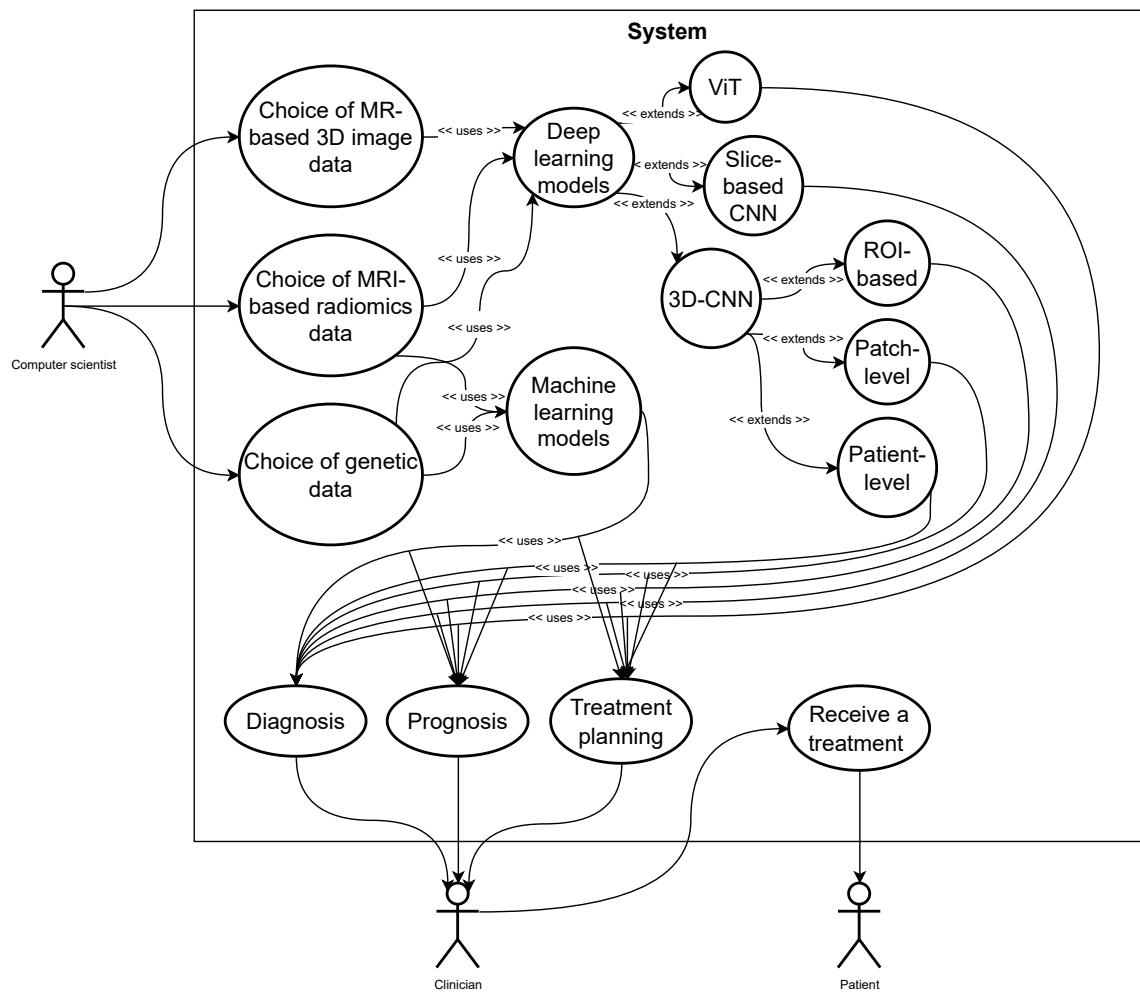


Figure 4.1 Use Case Diagram

5

System Design

The recent initiative efforts revealed many datasets for Alzheimer's disease. ADNI and OASIS (Open Access Series of Imaging Studies)¹ are the best examples of standardized open-source data to take a step forward both in research and practice. Roughly, but concurrently the developments in artificial intelligence offers many advantages using these datasets. On the other hand, information pollution is also a significant concern of modern studies. Many unknown sourced medical datasets are available in many platforms on the Internet. Therefore, this section will be starting with introducing ADNI dataset, which is standardized and well-known AD dataset, and proceeding with demonstrating an example of unemployable dataset can be found in Kaggle.

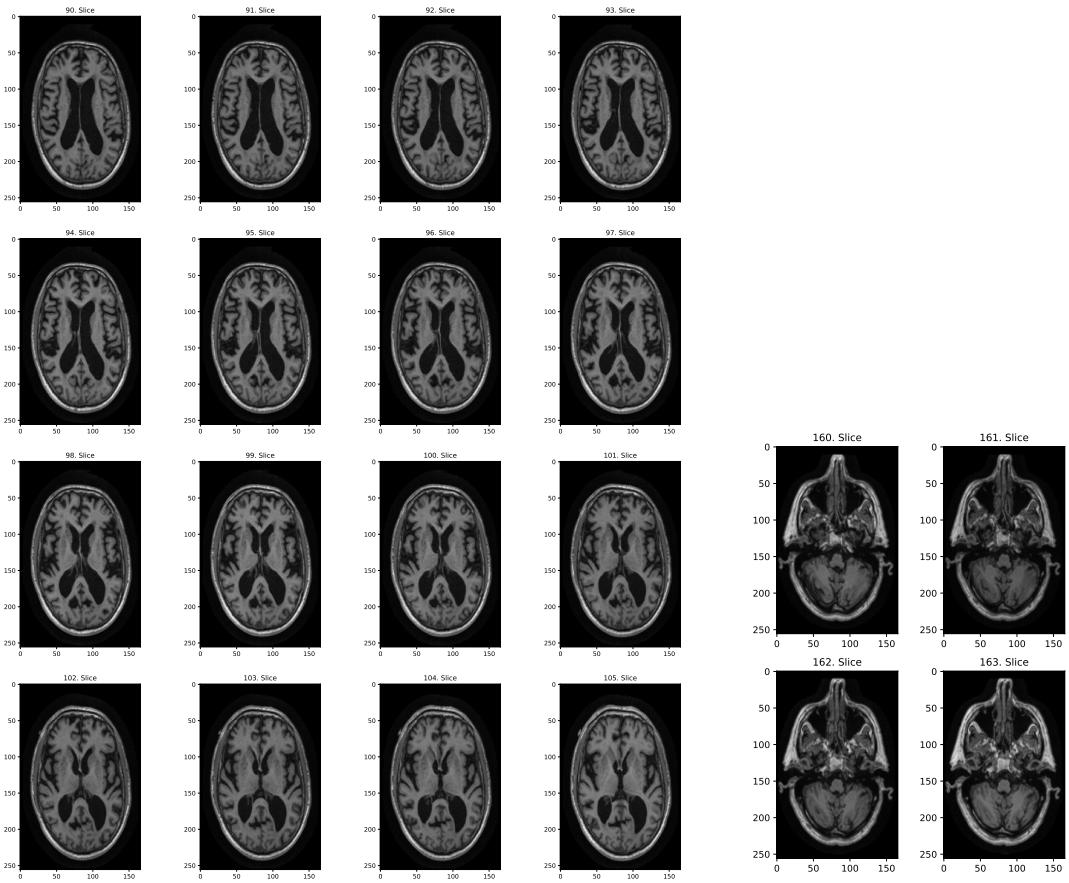
Afterwards, the most common pre-processing steps for brain imaging, and different deep learning approaches will be surveyed.

5.1 Overview of the Dataset: ADNI

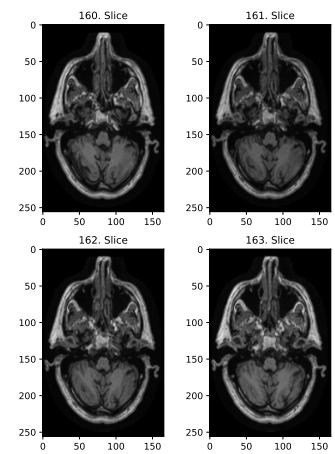
ADNI² platform contains many different default datasets. In this study, ADNI1:Complete 3Yr 1.5T (2182) dataset is utilized. In total 2182 3D NIfTI image files, with 981 MCI, 748 CN, and 453 AD are available. However, when we group several visits of subjects, we found that data contain 382 unique subjects with 148 MCI, 135 CN, and 99 AD. All subjects have raw 3D MRI data with 1.5 Tesla. In approximate average, each T1-weighted MRI datum consists of 240 axial, 240 coronal, and 176 sagittal slices. The shape of the image can be varying ((256, 256, 166), (192, 192, 160), and (256, 256, 180)) depending on the technicians and procedures of MR imaging.

¹<https://www.oasis-brains.org>

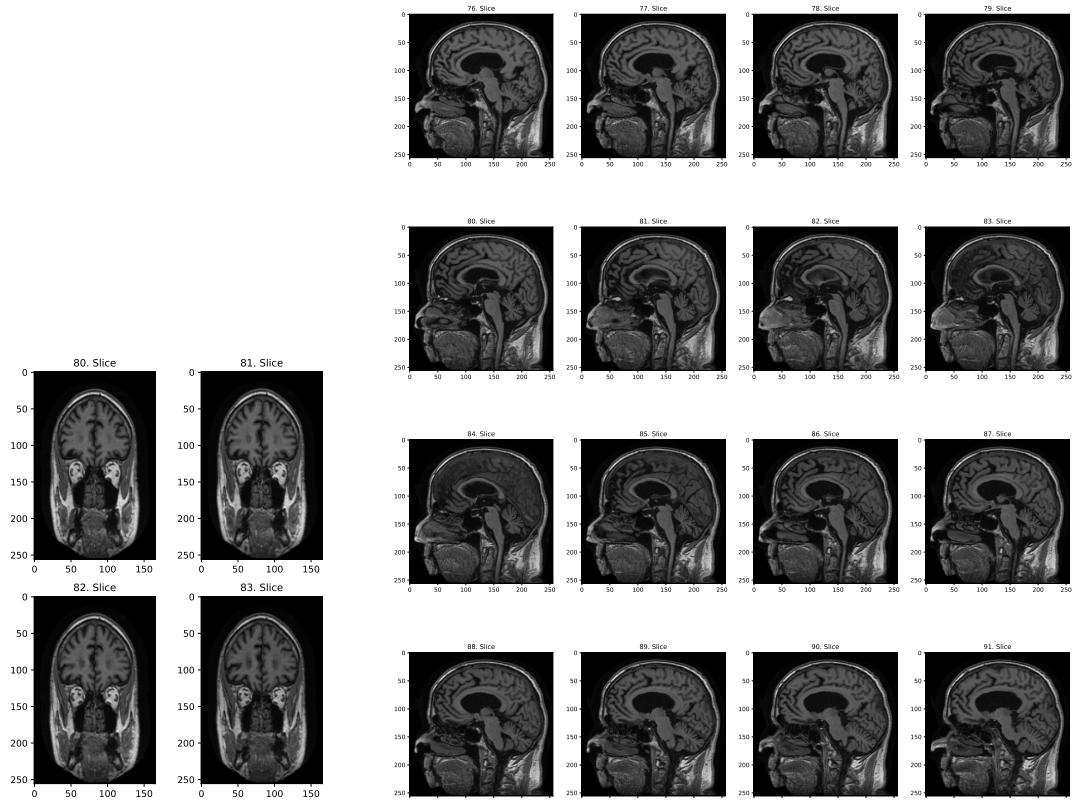
²<https://adni.loni.usc.edu>



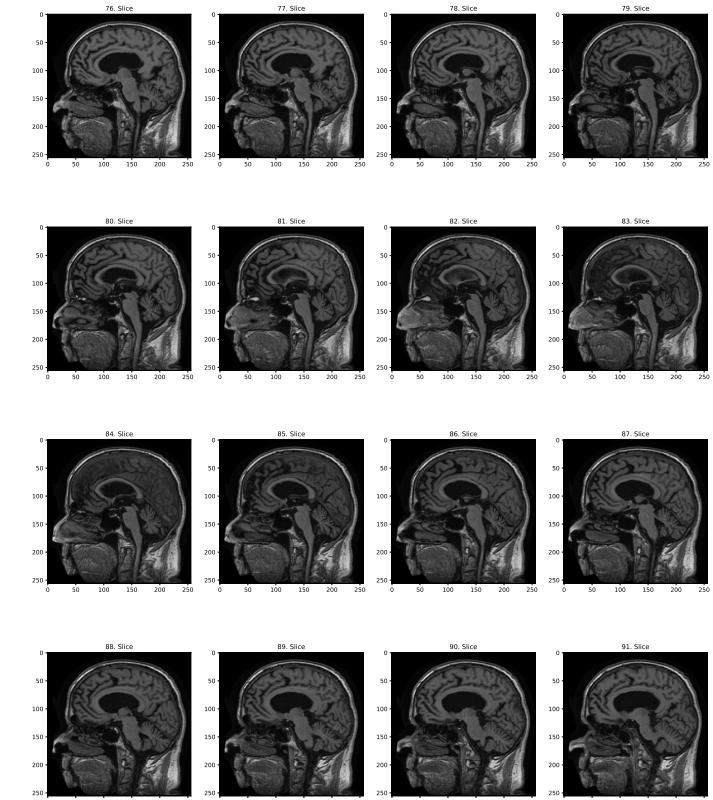
(a) Axial Slices (90 to 105)



(b) Axial Slices (160 to 163)



(c) Coronal Slices (80 to 83)



(d) Sagittal Slices from (76 to 91)

Figure 5.1 ADNI raw slices.

5.2 Overview of the Dataset: Kaggle

Two different versions of same dataset are investigated in Kaggle. 1) Open-source baseline dataset³ and 2) augmented version of baseline dataset⁴.

Baseline dataset contains total 6.400 2D slices with unknown location includes 896 Mild Demented, 64 Moderate Demented, 3.200 Non-Demented and 2.240 Very Mild Demented subjects, and augmented dataset contains total 33.984 samples with 8.960 Mild Demented, 6.464 Moderate Demented, 9.600 Non-Demented and 8.960 Very Mild Demented subjects. Augmented dataset employed as training set and baseline utilized as test set to simulate common wrong applications in the literature.

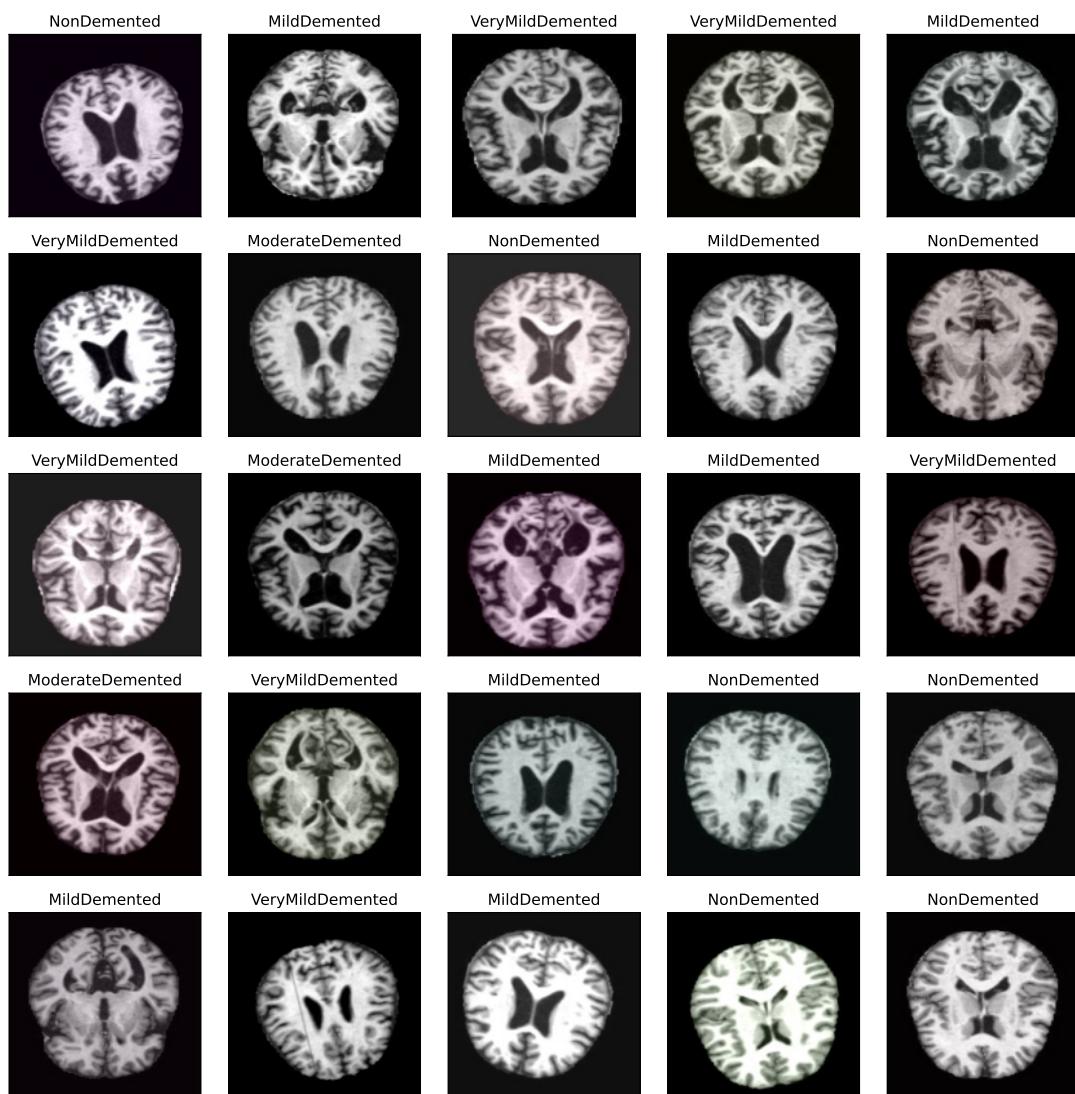


Figure 5.2 Random samples from Kaggle augmented dataset (training set in our wrong simulation study).

³<https://www.kaggle.com/datasets/tourist55/alzheimers-dataset-4-class-of-images>

⁴<https://www.kaggle.com/datasets/uraninjo/augmented-alzheimer-mri-dataset>

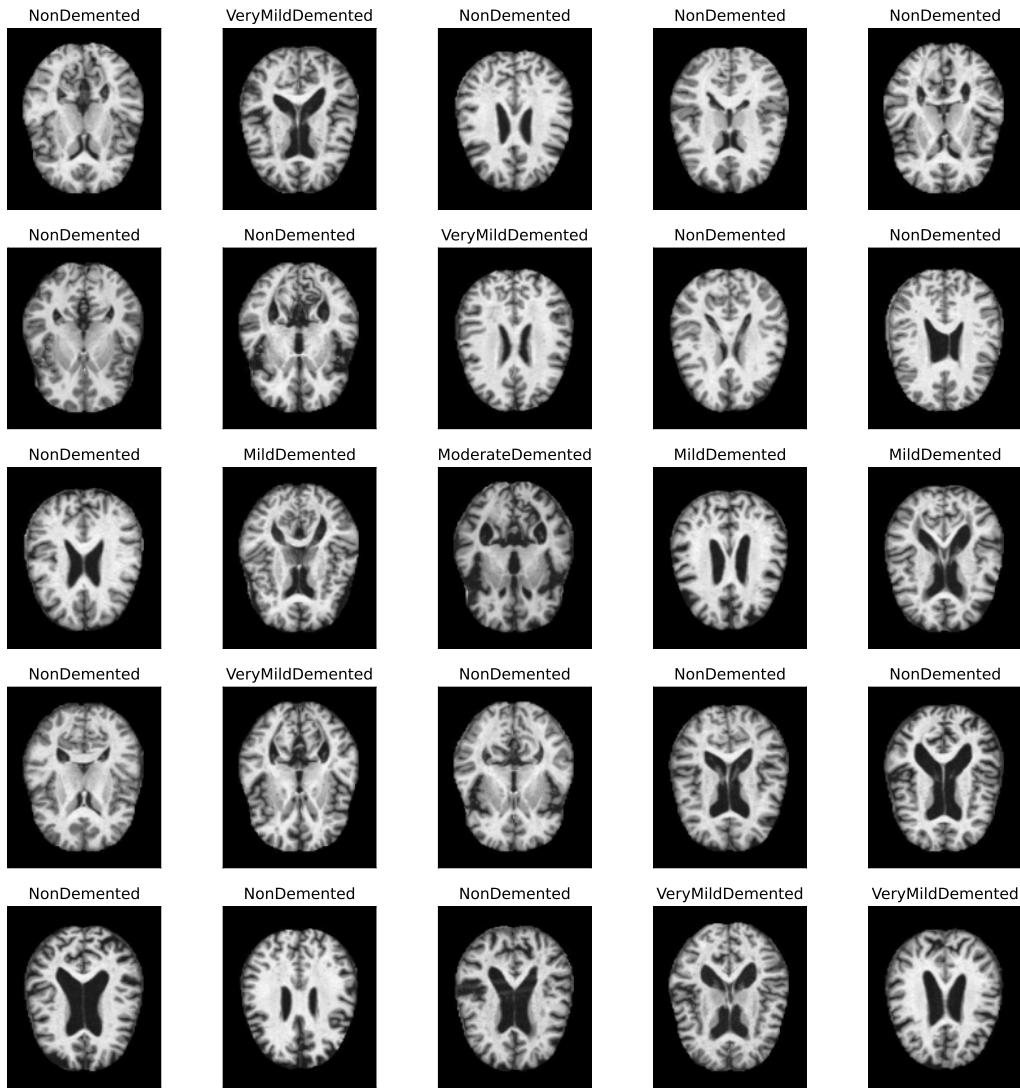


Figure 5.3 Random samples from Kaggle baseline dataset (test set in our wrong simulation study).

There are important issues that makes Kaggle dataset nonusable such as data source, patient id of images, slice number of images, and splitting method are not available. One of the important concepts to consider for deep learning in medical research is the appropriate train-test-val splitting of patient data due to leakage concerns.

5.3 Data Handling & Preprocessing

Magnetic resonance imaging (MRI) data must be preprocessed in order to eliminate undesirable artifacts and prepare the data in accordance with standards. A number of commonly used software for processing neuroimaging data include SPM, AFNI, FSL, FreeSurfer, and Workbench. The repeatability of neuroimaging investigations may be hampered by the inconsistent results produced by various data preparation

procedures. A common preprocessing steps for T1-weighted volume-based structural MRI include de-oblique, re-orientation, field inhomogeneity correction, non-brain tissue removal (i.e., skull stripping), registration and segmentation.

The standard dataset provided by ADNI has undergone various preprocessing steps, including Gradient Unwarping, B1 Correction, and scaling. However, these steps were not applied to all patients. Although the independent FreeSurfer software can serve as a tool for many preprocessing steps in MRI, it requires significant computational power and is costly to apply to each patient. Therefore, we completed this study using skull-stripping, which we deemed the most important preprocessing method.

From above tools, for skull-stripping, we utilized SynthStrip [28] (2022), which improved version of past stripping methods, and can be used through FreeSurfer [29].

We propose 2 ways of coding flow: intuitive and logic/math-grounded. In both implementation, the first step, creating command file, is exactly same, but differ at getting slices and cutting the empty pixels, which are the second and third steps.

5.3.1 Flow of Coding (Intuitive)

Three different utilization code pieces are written in order to handle raw data format and file structure. First is to prepare all necessary paths (e.g. input and output paths) for downloaded 80+ GB ADNI files, to be able to utilize FreeSurfer and SynthStrip in an automated way. Since the FreeSurfer requires manual input (e.g. *mri_synthstrip -i input -o output -no-csf*), we also created .txt files that allow us to send all inputs without any further involvement to the tool for all subjects (see in Fig. 5.4). *-no-csf* command is about cerebrospinal fluid (CSF). In the tissue that covers the brain and spinal cord of all vertebrates, a clear, colorless bodily fluid, is present. In the first preprocessing, we did not included the CSF (see in Fig. 5.7). Moreover, we made a patient-based train-test-val distribution in order to avoid data leakage, and we used the path information prepared with this methodological distribution while preparing the cmd commands.

```
mri_synthstrip -i /Users/toygar/Desktop/Bitirme/data/ADNI/027_S_0404/MPR_GradWarp_B1_Correction_N3_Scaled/2008-05-13_10_33_22.0/I106485/
ADNI_027_S_0404_MR_MPR_GradWarp_B1_Correction_N3_Scaled_Br_2008052131751599_550048_I106485.nii -o /Users/toygar/Desktop/Bitirme/data/train/
027_S_0404-2008-05-13.nii --no-csf
mri_synthstrip -i /Users/toygar/Desktop/Bitirme/data/ADNI/022_S_0130/MPR_GradWarp_B1_Correction_N3_Scaled/2009-03-03_13_44_17.0/I142344/
ADNI_022_S_0130_MR_MPR_GradWarp_B1_Correction_N3_Scaled_Br_20090422152149841_S63993_I142344.nii -o /Users/toygar/Desktop/Bitirme/data/train/
022_S_0130-2009-03-03.nii --no-csf
mri_synthstrip -i /Users/toygar/Desktop/Bitirme/data/ADNI/033_S_1098/MPR_R_GradWarp_B1_Correction_N3_Scaled/2006-11-21_13_38_28.0/I42832/
ADNI_033_S_1098_MR_MPR_R_GradWarp_B1_Correction_N3_Scaled_Br_20070306095352673_S22792_I42832.nii -o /Users/toygar/Desktop/Bitirme/data/train/
033_S_1098-2006-11-21.nii --no-csf
```

Figure 5.4 Sample cmd command file for FreeSurfer.

Then, other code piece helps to extract 2D slices from the 3D MR images, and move them into particular train-test-val files, where prepared with the prior code piece (see

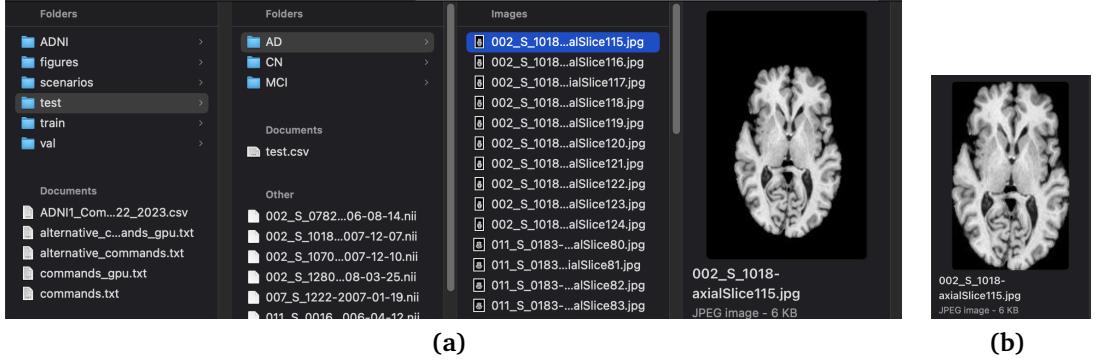


Figure 5.5 a) Automated slice extraction into particular pathway (code piece 2), and b) slices after removing empty pixels (code piece 3).

our automated path format in Fig. 5.5a). Ultimately, a custom code piece is written to simply cut the empty pixels around the brain tissue (see in Fig. 5.5b).

Only one MR image file is used for each subject, although entire visiting data (i.e., 1 subject may come 6 times for per 6 months) are available due to the fact that it requires enormous amount of file processing power for our circumstances. On the other hand, if we look at the issue from another angle, it might be that we already should not take images of other visits of the same patient, since we do not aim to find the development of the disease in same person, but only to distinguish the stages of the disease.

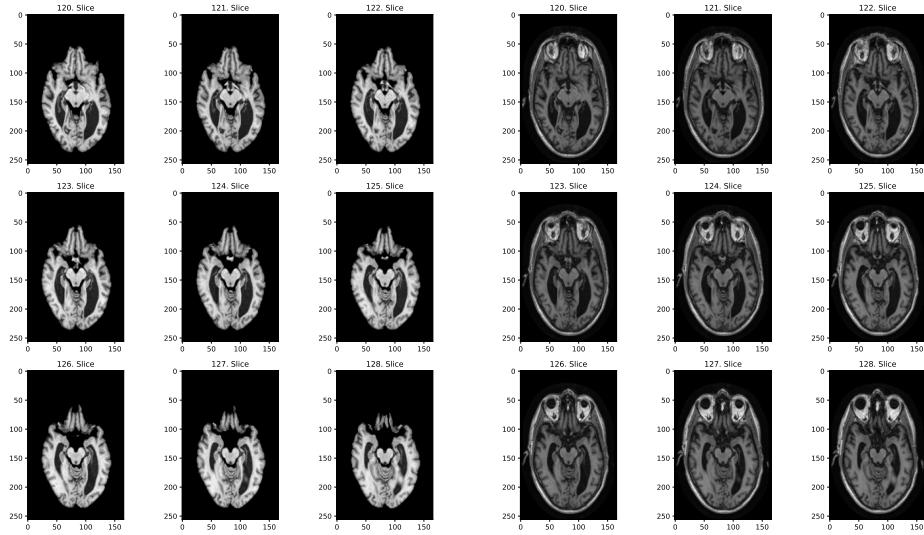
All slices are formatted to *subjectID_sliceNo*. The code also allows to create custom formats, since another information is needed in the filename. In first attempts, code in Fig. 5.6 is utilized for extracting slices.

```
def get_slices(self, subject_id, path, dataset, group, axis, region, width, iter, format):
    img_path = self.take_mri(subject_id, os.path.join(path, dataset))
    img = self.read_image(img_path)

    if region == 'mid': ## If iter = 1: slice count = width*2
        if (img.shape[0] > 200 or img.shape[1] > 200) and img.shape[2] >= 166:
            start = 120 - width; end = 120 + width; sagittal_start = 90 - width; sagittal_end = 90 + width
        else:
            start = 85 - width; end = 85 + width; sagittal_start = 85 - width; sagittal_end = 85 + width
    elif region == 'mid_to_back': ## If iter = 1: slice count = width
        if (img.shape[0] > 200 or img.shape[1] > 200) and img.shape[2] >= 166:
            start = 120 - width; end = 120; sagittal_start = 90 - width; sagittal_end = 90
        else:
            start = 80 - width; end = 80; sagittal_start = 80 - width; sagittal_end = 80
    else: #mid_to_forward ----> If iter = 1: slice count = width
        if (img.shape[0] > 200 or img.shape[1] > 200) and img.shape[2] >= 166:
            start = 120; end = 120 + width; sagittal_start = 90; sagittal_end = 90 + width
        else:
            start = 85; end = 85 + width; sagittal_start = 85; sagittal_end = 85 + width
```

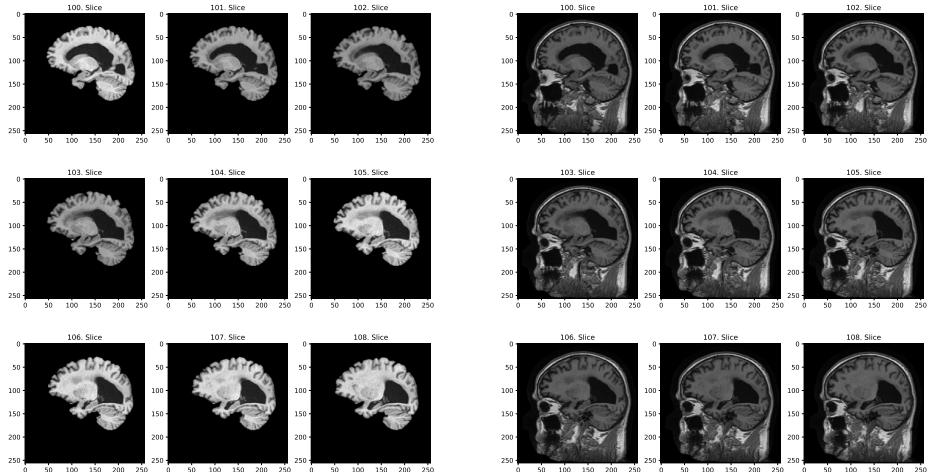
Figure 5.6 Intuitive function for extracting slices from 3D images.

Code simply aims to determine 3 different modes for the regions that user can extract slices. We restricted the access, since entire slices do not include brain tissue. The user can provide *axis*: (*axial*, *sagittal*, *coronal*), *width*: scope of slices, *iter*: jump amount (i.e., if "1" is chosen takes every slices in the scope, whereas "2" takes 1 in 2



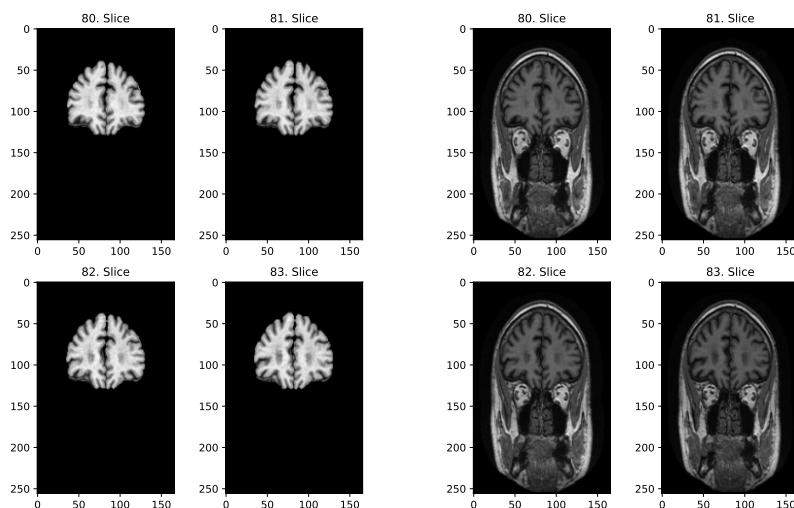
(a) Stripped - Axial

(b) Original - Axial



(c) Stripped - Sagittal

(d) Original - Sagittal



(e) Stripped - Coronal

(f) Original - Coronal

Figure 5.7 Skull-Stripping outcomes of the data.

slices), *format*: (.pdf, .png), *region*: ('mid', 'mid_to_back', 'mid_to_forward'). We also controlled, and changed the intervals through variation of different subjects' slices. As we mentioned above their output dimensions are varying ((256, 256, 166), (192, 192, 160)). Moreover, with the addition of many other artifacts (e.g. moving heads), extracting same slice for two different subjects, is rough. Therefore, we managed the difference as much as possible with several restriction and rules. To avoid this issue, in a more advanced way, we could use 3D CNNs or ROI-based slice extraction with deep learning.

SynthStrip expands on the strong groundwork established by earlier investigations of deep-learning algorithms for brain extraction, allowing us to select from a variety of network designs perfectly suited for this particular purpose. A crucial first step for many quantitative studies is the elimination of non-brain signal from neuroimaging data, and the precision of this process directly affects the outcomes.

5.3.2 Flow of Coding (Logic-based)

In this section, we changed the method of getting center of the 3D MR images, that has an direct impact on extraction step. We also changed and tried different cutting approach. In intuitive coding, we simply removed entire empty rows and columns from images, however, there is also possibility to lose important information while resizing these images. Therefore, in this section after exact same cutting with previous one, we refilled images with zeros to protect aspect-ratio for DL utilization.

```
def get_from_exact_center(self, subject_id, path, dataset, group, axis, region, width, iter, format):
    img_path = self.take_mri(subject_id, os.path.join(path, dataset))
    img = self.read_image(img_path)

    n_i, n_j, n_k = img.shape
    center_i, center_j, center_k = n_i // 2, n_j // 2, n_k // 2

    if region == 'mid': ## If iter = 1: slice count = width*2
        if axis == 'axial':
            start = center_i - width; end = center_i + width;
        elif axis == 'coronal':
            start = center_j - width; end = center_j + width;
        else: #Sagittal
            start = center_k - width; end = center_k + width;
    elif region == 'mid_to_back': ## If iter = 1: slice count = width
        if axis == 'axial':
            start = center_i - width; end = center_i;
        elif axis == 'coronal':
            start = center_j - width; end = center_j;
        else: #Sagittal
            start = center_k - width; end = center_k;
    else: #mid_to_forward ----> If iter = 1: slice count = width
        if axis == 'axial':
            start = center_i; end = center_i + width;
        elif axis == 'coronal':
            start = center_j; end = center_j + width;
        else: #Sagittal
            start = center_k; end = center_k + width;
```

Figure 5.8 Logic-based function for extracting slices from 3D images.

```

center_i: 128, center_j: 128, center_k: 83 n_i: 256 n_j: 256 n_k: 166
center_i: 96, center_j: 96, center_k: 80 n_i: 192 n_j: 192 n_k: 160
center_i: 128, center_j: 128, center_k: 83 n_i: 256 n_j: 256 n_k: 166
center_i: 128, center_j: 128, center_k: 83 n_i: 256 n_j: 256 n_k: 166
center_i: 128, center_j: 128, center_k: 83 n_i: 256 n_j: 256 n_k: 166
center_i: 96, center_j: 96, center_k: 80 n_i: 192 n_j: 192 n_k: 160
center_i: 128, center_j: 128, center_k: 83 n_i: 256 n_j: 256 n_k: 166
center_i: 96, center_j: 96, center_k: 80 n_i: 192 n_j: 192 n_k: 160
center_i: 96, center_j: 96, center_k: 80 n_i: 192 n_j: 192 n_k: 160
center_i: 128, center_j: 128, center_k: 83 n_i: 256 n_j: 256 n_k: 166
center_i: 83, center_j: 128, center_k: 128 n_i: 166 n_j: 256 n_k: 256

```

Figure 5.9 Variety of 3D images through different MRI protocols can be seen.

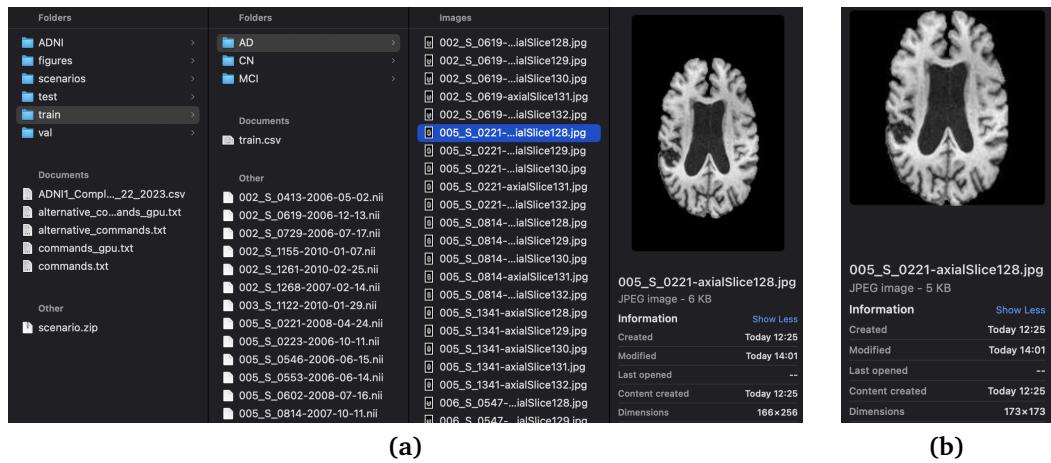


Figure 5.10 a) Automated slice extraction into particular pathway (code piece 2), and **b)** slices after removing empty pixels and resizing to square with padding (code piece 3).

5.4 Deep Learning Approaches

Traditional CNN-based models (e.g. VGG19, ResNet101, Xception, MobileNet, DenseNet169, and InceptionV3) can be used as a baseline in any kind of image classification task. Moreover, with the recent developments, CNN architectures have come a long way to employ 3D image processing, helping model to learn subject-based patterns for slices, rather learning only the patterns of random slices.

It is also necessary to mention the transformers, which have turned all the rankings upside down in current computer vision literature. Natural language processing (NLP) tasks were the first to use *transformers*, that success has been proven by language models such as BERT and GPT-3. A methodology for classifying images called the Vision Transformer, or ViT, applies a Transformer-like design to patches of the image.

5.4.1 Traditional Deep Learning: 2D Slice-based

5.4.1.1 VGG

The VGG [30] (Visual Geometry Group) model is a deep convolutional neural network architecture that was introduced in 2014 and has been widely used in computer vision tasks, including medical image analysis.

VGG is known for its simplicity and elegance, with a fixed structure consisting of a stack of convolutional layers followed by fully connected layers. Moreover, researchers have experimented with variants of VGG by changing the number of layers or adding regularization techniques to improve its performance. Additionally, transfer learning, which involves using pre-trained VGG models trained on large image datasets like ImageNet, has been applied to medical image analysis tasks with promising results.

The network structure. An image of size 224x224 is fed into VGGNet. By removing a 224x224 square from the center of each image submitted for the ImageNet competition, the developers of the models were able to maintain a constant image input size. The VGG convolutional filters employ a 3x3 receptive field, which is the smallest choice. The linear transformation of the input for VGG also uses an 1x1 convolution filter, and has a maximum of 16 convolutional layers. The fundamental VGG architecture is shown below:

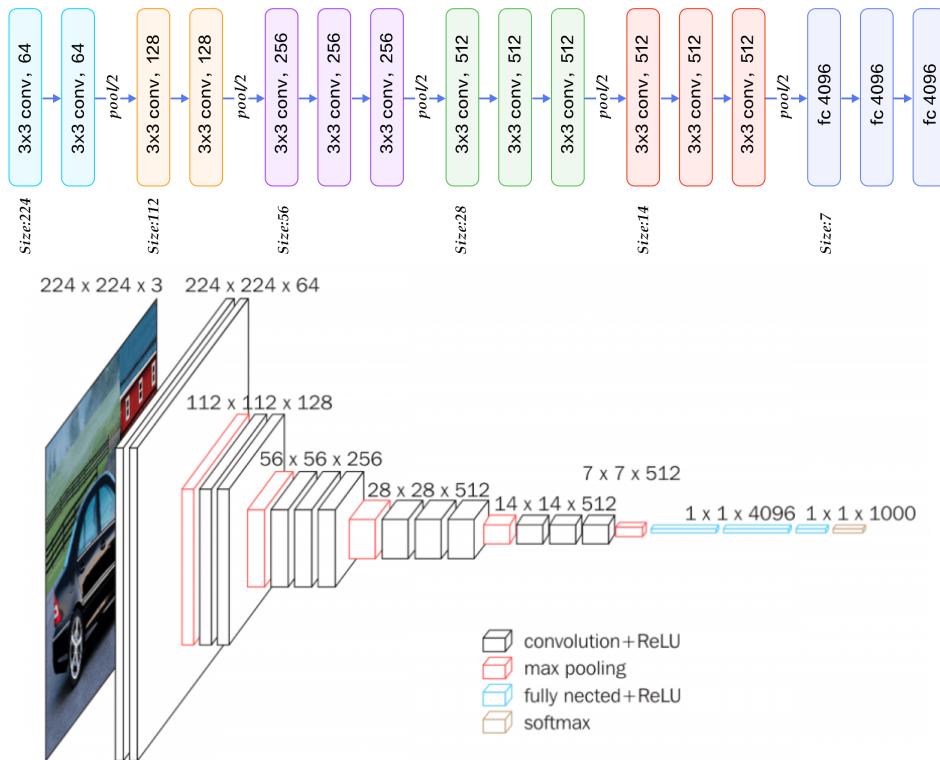


Figure 5.11 VGG Network.

5.4.1.2 ResNet

ResNet (Residual Neural Network) [31] is a deep learning architecture that was introduced in 2015 and has since been widely used in computer vision tasks, including medical image analysis. The key innovation of ResNet is the use of residual connections, which address the problem of vanishing gradients in very deep networks.

Why? In a typical deep neural network, each layer feeds into the next layer, with the output of each layer serving as the input to the next. As the network becomes deeper, however, the gradients that are backpropagated during training can become very small, which can lead to the problem of vanishing gradients. This can make it difficult to train very deep networks effectively.

ResNet addresses this problem by introducing skip connections that allow the output of a layer to be added directly to the output of an earlier layer. This creates a shortcut that bypasses the intervening layers and allows the gradients to flow more easily through the network. These skip connections create "residual blocks" that can learn the residual mapping between the input and the output of the block, rather than trying to learn the entire mapping directly. This makes it easier to train very deep networks and improves their performance.

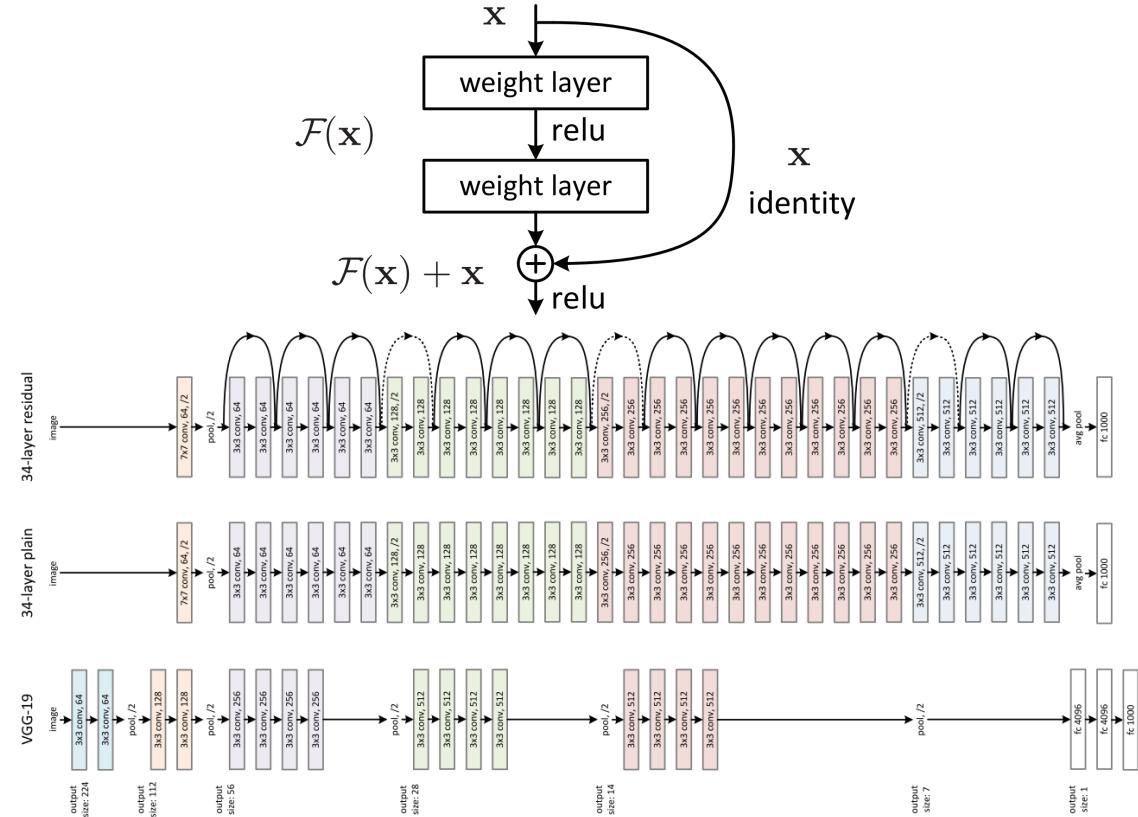


Figure 5.12 Fundamental approach of residual networks, and ResNet comparison with plain and VGG19 networks.

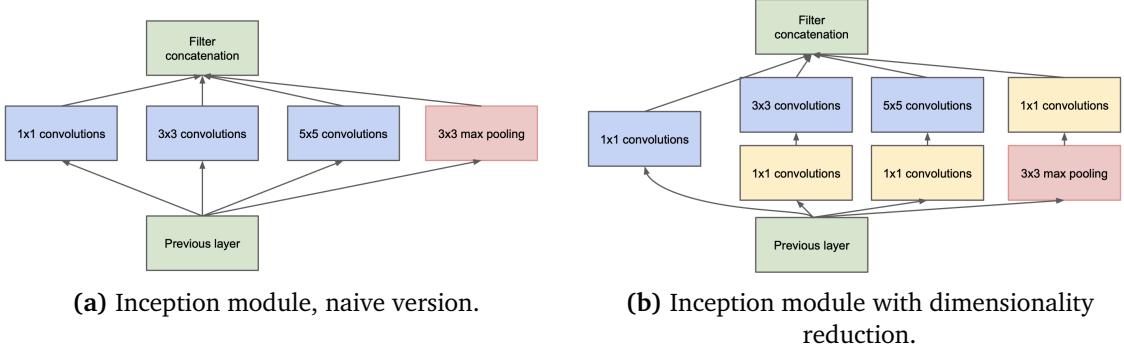


Figure 5.13 Inception module

5.4.1.3 Inception

The Inception model [32, 33] was introduced in 2014 and was designed to address the issue of efficient feature extraction in deep learning models. Inception models use multiple layers of convolutional neural networks with varying filter sizes to extract features from the input data. By using multiple filter sizes, the model is able to capture features of different scales, which can be important in medical imaging where the features of interest can be at varying sizes.

However, while Inception models have shown promise in AD classification, there are limitations that need to be addressed. One limitation is the interpretability of the model, as it can be difficult to understand how the model is making its predictions. Another limitation is the generalizability of the model to different populations, as the model may not perform as well on data from populations that are different from the training dataset. Overall, more research is needed to further evaluate the potential of Inception models in AD diagnosis and to address these limitations.

Concept. Deep neural networks that perform well must be vast. A neural network required numerous additional network layers and units inside those layers in order to be categorized as huge. Extraction of features at various scales is advantageous for convolutional neural networks, and also the biological human visual brain works by recognizing patterns at various scales that combine to create larger perceptions of objects. The famous psychologist Hebbian proposed a principle – *neurons that fire together, wire together* for learning theory, and the paper is essentially based on this idea.

5.4.1.4 Xception

Xception [34] is a type of deep learning model that has been used for AD classification using both structural and functional brain imaging data, and has shown promising

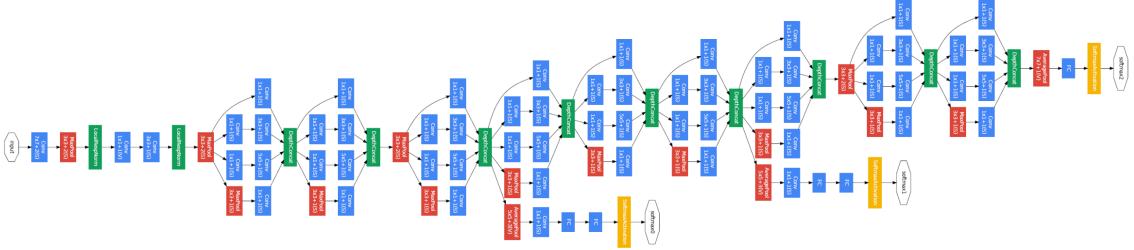


Figure 5.14 GoogLeNet architecture.

results with high accuracy and efficiency. It was proposed as a modification of the Inception architecture, in which depth-wise separable convolutions replace standard convolutions. This approach reduces the number of parameters in the network and improves its efficiency.

Difference from Inception. While Inception networks use a combination of different-sized convolutional filters to capture features at multiple scales, Xception networks use depthwise separable convolutions to achieve the same goal with fewer parameters and computations. Xception has been shown to outperform Inception in some image classification benchmarks, but the choice of architecture ultimately depends on the specific task and available resources.

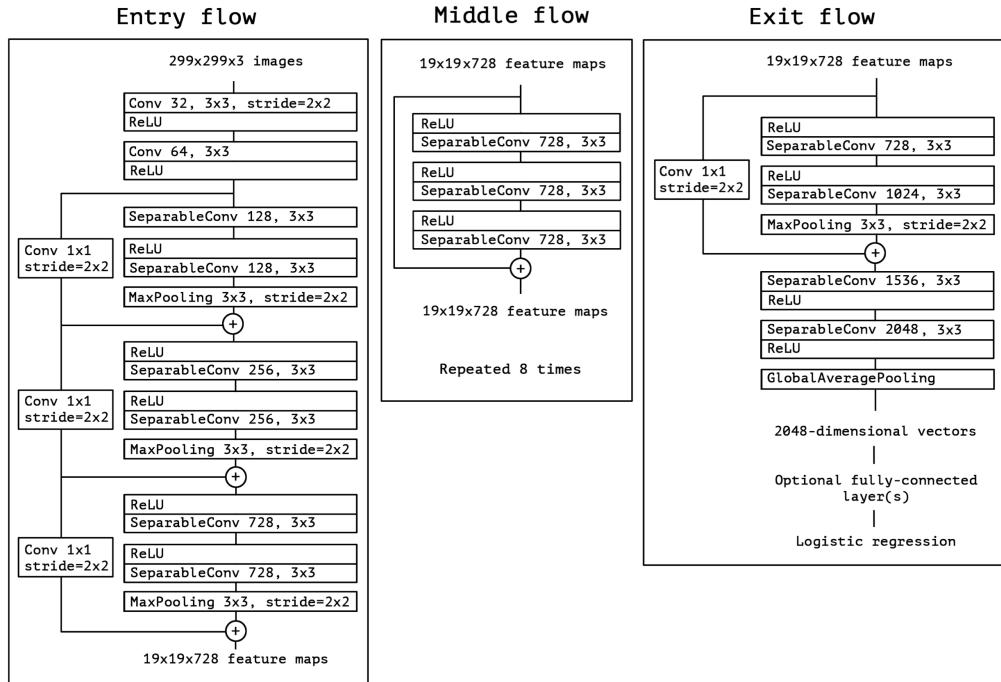


Figure 5.15 Xception architecture.

5.4.1.5 MobileNet

MobileNets, a type of lightweight deep learning model, have also been explored for the classification of AD using brain imaging data. These models are designed to be computationally efficient and are well-suited for deployment on mobile and edge devices. MobileNets have been used for AD classification using both structural and functional brain imaging data, and have shown promising results with high accuracy and efficiency.

Structure. Depth-wise separable convolutions, which Xception also demonstrated its potential efficiency, were used in MobileNet. When compared to a network with conventional convolutions of the same depth in the nets, it dramatically reduced the number of parameters. Lightweight deep neural networks are the outcome of this approach. From two operations, a depthwise separable convolution is created: depth-wise convolution, and point-wise convolution (Fig. 5.16⁵).

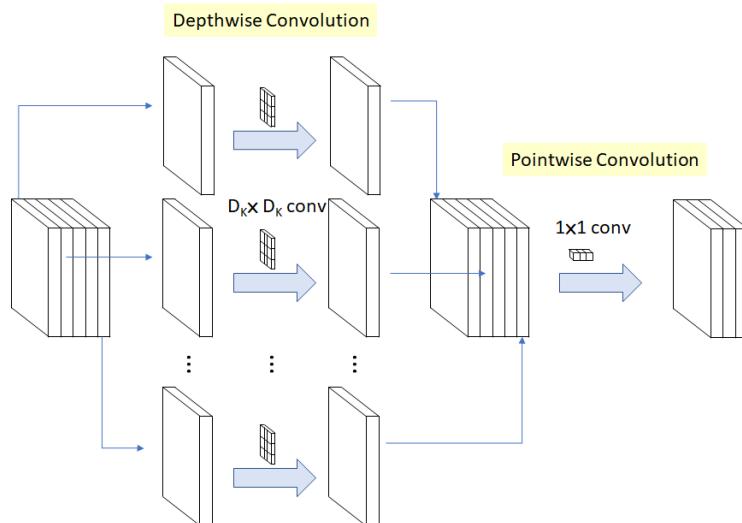


Figure 5.16 MobileNet depth-wise and point-wise convolution architecture.

5.4.1.6 DenseNet

DenseNets [35], a type of deep learning model, have shown promise in the classification of Alzheimer's disease (AD) using brain imaging data. These models utilize densely connected neural networks that facilitate the flow of information throughout the network, allowing for more efficient feature extraction and classification. DenseNets have been used for AD classification using both structural and functional brain imaging data, and have shown improved accuracy compared to traditional machine learning algorithms. However, more research is needed to further

⁵<https://medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fbb2cd470>

evaluate the potential of DenseNets in AD diagnosis and to address limitations such as model interpretability and generalizability to different populations.

What problem DenseNets solve? ResNets have a large number of parameters, since each layer has weights that must be learned, and several ResNets variations have shown that many layers can be removed, due to their negligible contributions. Contrary to common belief, by connecting in this manner (Fig. 5.17⁶) DenseNets need fewer parameters than a comparable standard CNN, since redundant feature maps are not necessary to learn. Due to the information flow and gradients, very deep networks also had difficulty being trained. Since each layer of DenseNets has direct access to the gradients from the loss function and the original input image, this problem is resolved.

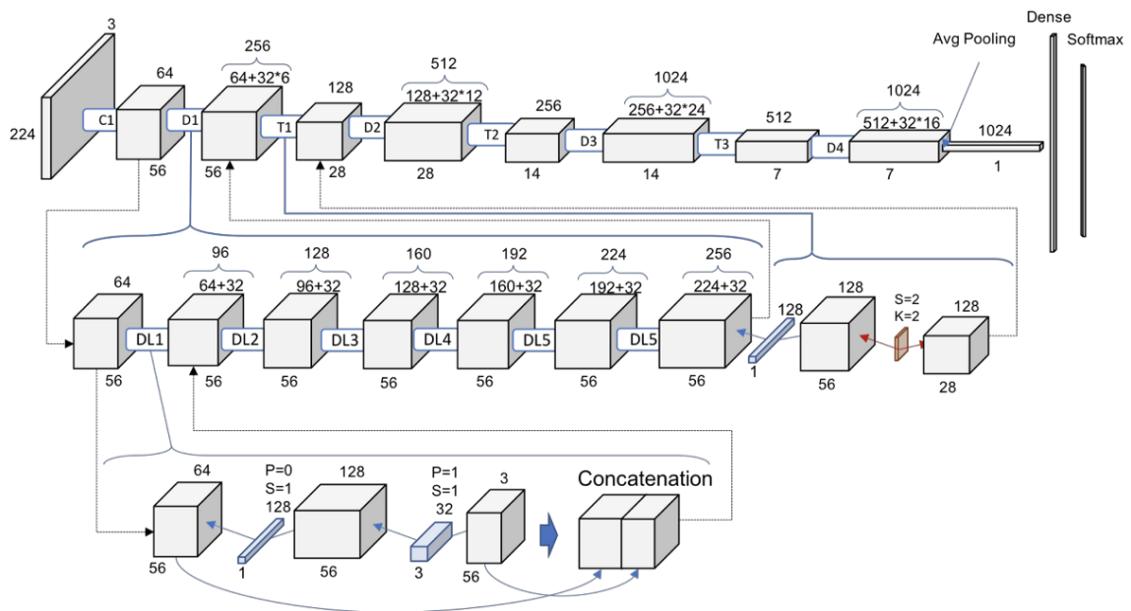


Figure 5.17 Visualization of DenseNet architecture.

5.4.2 3D patch-level CNN

Some studies concentrated on the 3D patch-level categorization to make up for the lack of 3D information in the 2D slice-level approach. These frameworks take as input a collection of 3D patches that have been taken out of an image. Since the number of samples would equal the number of patches, this might theoretically lead to a higher sample size, similar to the 2D slice-level technique (and not the number of subjects). Patches also have a lower memory usage, which might be advantageous if one only has a limited amount of RAM, and fewer parameters to learn. This last benefit, however, is only available if all patches are applied to the same network.

⁶<https://towardsdatascience.com/understanding-and-visualizing-densenets-7f688092391a>

The size and stride of patches must be chosen for 2D-slice level approaches. Based on the MRI preprocessing, these hyperparameters will be selected differently (e.g. a non-linear registration is likely needed for smaller patches).

5.4.3 ROI-based CNN

ROI-based CNNs are a type of deep learning approach that is commonly used for medical image analysis. These models aim to identify and classify specific regions of interest (ROIs) within a medical image, such as lesions, tumors, or other pathological features. By focusing on these specific ROIs, ROI-based CNNs can achieve high accuracy and efficiency in medical image analysis tasks.

The general structure of an ROI-based CNN involves training a CNN model on a large dataset of annotated medical images. During the training process, the model learns to extract relevant features from the images and use these features to classify different ROIs within the image. The model is typically trained using a supervised learning approach, where the ground-truth ROI labels are provided during training.

In current literature, choice of the ground-truth ROI is the Hippocampus in common when diagnosing Alzheimer's disease on 3D MR images. The reason for that it is widely known that the hippocampus is affected early in degeneration.

Once the model has been trained, it can be used to analyze new medical images and identify the specific ROIs within the image. This typically involves applying the trained model to each image patch within the image, and using the output of the model to generate a probability map indicating the likelihood that each patch contains an ROI. These probability maps can then be used to segment the image and identify the specific ROIs within it.

One advantage of ROI-based CNNs is their ability to focus on specific ROIs within an image, rather than analyzing the entire image as a whole. This can improve the accuracy and efficiency of the model, particularly when dealing with complex medical images where there may be a large amount of irrelevant information. Additionally, ROI-based CNNs can be used for a wide range of medical image analysis tasks, including tumor detection, lesion segmentation, and disease classification.

However, there are also some limitations to the ROI-based CNN approach. One limitation is the requirement for a large dataset of annotated images for training the model. This can be particularly challenging in medical imaging, where obtaining and annotating such a dataset can be time-consuming and expensive. Additionally, ROI-based CNNs may not generalize well to new imaging modalities or populations,

as they are often trained on specific datasets with limited variability. Finally, there are challenges related to interpretability and explainability of the model, as ROI-based CNNs can be difficult to interpret and understand. To address these limitations, ongoing research is focused on developing new methods for training and validating ROI-based CNNs, as well as methods for improving the interpretability of these models.

5.4.4 3D subject-level CNN

The 3D subject-level CNN is a deep learning approach that aims to classify subjects as either healthy controls or individuals with AD based on their 3D MRI scans. This approach involves training a CNN model on a large dataset of 3D MRI scans, which allows the model to learn to extract features from the scans and use these features to differentiate between healthy and AD subjects.

The 3D subject-level CNN model takes in a 3D MRI scan of a subject as input, which is typically preprocessed to remove any non-brain tissue and normalize the intensity values. The model then applies a series of 3D convolutional layers to the input image, which learn to extract relevant features from the image at different spatial scales. These convolutional layers are typically followed by pooling layers, which reduce the dimensionality of the features and help to prevent overfitting.

After the convolutional and pooling layers, the features are flattened and passed through a series of fully connected layers, which learn to combine the extracted features and generate a final classification output. The final output of the model is a binary classification label indicating whether the subject has AD or is a healthy control.

One advantage of the 3D subject-level CNN approach is that it can incorporate information from the entire 3D volume of the MRI scan, rather than just analyzing individual 2D slices as in some other approaches. This can improve the accuracy and robustness of the model, as it allows the model to capture information about the overall structure and shape of the brain.

However, there are some limitation such as limited interpretability, flexibility, and generalizability: One challenge with deep learning models like the 3D subject-level CNN is that they can be difficult to interpret, meaning it can be challenging to understand why the model made a particular prediction. This is particularly problematic in medical applications, where clinicians need to be able to understand and interpret the outputs of a model in order to make informed decisions. There is ongoing research into methods for improving the interpretability of deep learning

models, but this remains a limitation of the approach. Moreover, the approach is a relatively rigid framework that is designed to perform a specific task (i.e., classifying subjects as healthy controls or individuals with AD). While this can be useful for specific applications, it may not be as flexible or adaptable to other tasks or datasets. For example, if a researcher wanted to use a 3D CNN for a different disease or condition, they would need to retrain the entire model from scratch using a new dataset. Another limitation of the approach is that it may not generalize well to new populations or imaging modalities. This is because the model is trained on a specific dataset, and may not be able to generalize to new data that have different characteristics or features. To address this limitation, it is important to validate the model on independent datasets and to consider methods for transfer learning or domain adaptation.

5.4.5 Vision Transformers

Vision transformers [36–38], also known as self-attention networks, are a class of deep learning models that have shown promise in medical research for image classification and segmentation tasks. These models are based on the transformer architecture, which was originally developed for natural language processing but has since been adapted for image analysis. Vision transformers use self-attention mechanisms to extract features from images, allowing them to capture global relationships between image regions and to focus on relevant regions for a given task. In medical research, vision transformers have been used for tasks such as automated diagnosis and segmentation of medical images, including MRI and CT scans. These models have the potential to improve the accuracy and efficiency of medical image analysis and to enable the development of more personalized treatment plans for patients. However, further research is needed to fully understand the capabilities and limitations of vision transformers in medical applications.

Structure. The general structure of vision transformers for medical imaging applications is similar to that of vision transformers for image analysis in other domains. The model consists of a series of layers, each of which contains a self-attention mechanism and a feedforward neural network. The self-attention mechanism allows the model to attend to relevant image regions and capture global relationships between them, while the feedforward network processes the attended features and generates output predictions. In medical imaging applications, the input to the model is typically a 2D or 3D image, such as a magnetic MRI or CT scan. The image is first divided into patches, which are then fed into the model as a sequence of vectors. The model processes the sequence of vectors using the self-attention and

feedforward layers, and generates output predictions for the given task, such as image segmentation or diagnosis. The specific architecture and hyperparameters of the model may vary depending on the task and dataset being used.

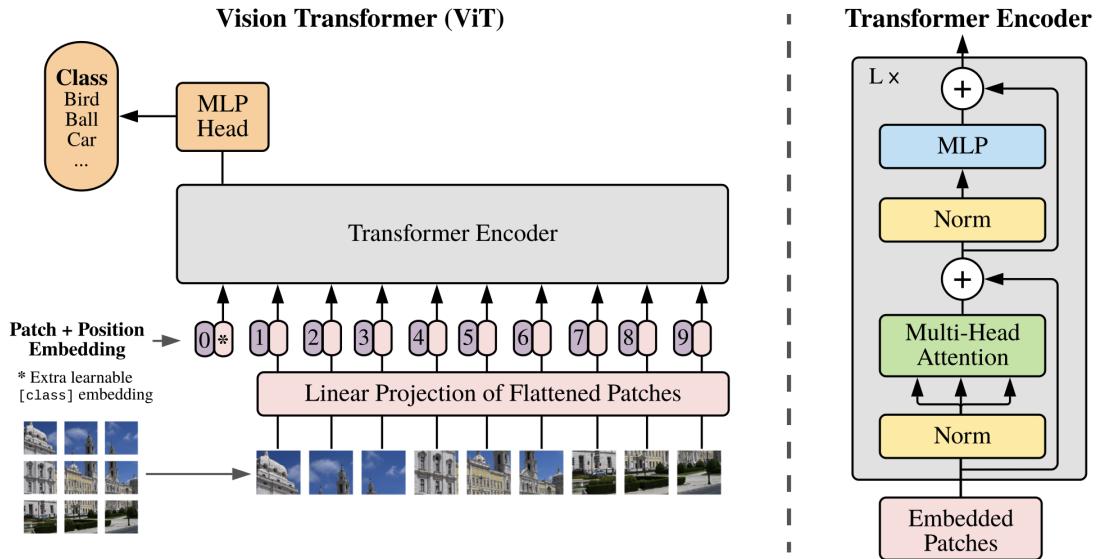


Figure 5.18 ViT, classification model overview [37].

6

Implementation

6.1 Standardized Implementation: 2D Slice-based

The most reliable datasets are ADNI and OASIS in Alzheimer’s disease research for many reasons. These initiatives provide entire procedure applied to the subjects, which is quite important while reasoning. Therefore, we also utilized ADNI dataset to realize the study.

The main implementation that we aimed to provide for this dataset is that the method of retrieving particular slices from the 3D MR images and separate slices into train and test files without data leakage. In our method, first we separated unique patients into train, test and val files, where all slices will be settled. Then, any slices can be retrieved by determining the axis, and relevant support parameters. Ultimately, we handled the empty pixels in slices by using Fig. 5.5b and Fig. 5.10b.

We utilized Kaggle GPU (P100) to run, Tensorflow library to implement, and trained ten different deep learning models for 50 epochs with early-stopping utility, and 100 batch size. Image size was chosen as (128, 128) for intuitive coding, and (160,160) for logic-based coding. We applied several experiments based on different data utilization procedure.

General description for following inputs:

axis. input: (“axial”, “coronal”, “sagittal”, “all”). The “all” parameter is to do all axis cases in one command.

region. input: (“mid”, “mid_to_back”, “mid_to_forward”). We offer 3 options here. Region parameter represents the regions of the 3D image that we have defined. *mid*: takes the slices above and below the value determined as mid (pre-defined, see in Fig. 5.6) with the width of “width” by “iter”. *mid_to_back*: takes the slices below the value determined as mid with the width of “width” by “iter”. *mid_to_forward*: takes the slices above the value determined as mid with the width of “width” by “iter”.

format. input: (“.pdf”, “.jpg”, “.png”). Provide available image format.

Procedure 1. Classification of AD, MCI and CN. The data were extracted from skull-stripped 3D MR images and using 5.3.1. The following inputs are employed:

```
$ u = utils()
$ u.run(train, path, 'train', axis = 'all', region='mid',
       width=12, iter=4, format = '.jpg')
$ u.run(val, path, 'val', axis = 'all', region='mid',
       width=12, iter=4, format = '.jpg')
$ u.run(test, path, 'test', axis = 'all', region='mid',
        width=12, iter=4, format = '.jpg')
```

Train:	Validation:	Test:
AD -> 1242	AD -> 270	AD -> 270
CN -> 1692	CN -> 378	CN -> 360
MCI -> 1872	MCI -> 396	MCI -> 396
+ _____	+ _____	+ _____
4806	1044	1026

Figure 6.1 Procedure 1. Total number of slices.

Procedure 2. Classification of AD and CN. The data were extracted from skull-stripped 3D MR images and using 5.3.1. The following inputs are employed:

```
$ u = utils()
$ u.run(train, path, 'train', axis = 'all', region='mid',
       width=16, iter=2, format = '.jpg')
$ u.run(val, path, 'val', axis = 'all', region='mid',
       width=16, iter=2, format = '.jpg')
$ u.run(test, path, 'test', axis = 'all', region='mid',
        width=16, iter=2, format = '.jpg')
```

Train:	Validation:	Test:
AD -> 3312	AD -> 720	AD -> 720
CN -> 4512	CN -> 1008	CN -> 960
+ _____	+ _____	+ _____
7824	1728	1680

Figure 6.2 Procedure 2. Total number of slices.

Procedure 3. Classification of AD and CN. The data were extracted from skull-stripped 3D MR images and using 5.3.1. The following inputs are employed:

```
$ u = utils()
$ u.run(train, path, 'train', axis = 'axial', region='mid',
       width=15, iter=1, format = '.jpg')
$ u.run(val, path, 'val', axis = 'axial', region='mid',
       width=15, iter=1, format = '.jpg')
$ u.run(test, path, 'test', axis = 'axial', region='mid',
       width=15, iter=1, format = '.jpg')
```

Train:	Validation:	Test:
AD -> 2070	AD -> 450	AD -> 450
CN -> 2820	CN -> 630	CN -> 600
+ _____	+ _____	+ _____
4890	1080	1050

Figure 6.3 Procedure 3. Total number of slices.

Procedure 4. Replication of Procedure 1 using 5.3.2.

Procedure 5. Replication of Procedure 2 using 5.3.2.

Procedure 6. Replication of Procedure 3 using 5.3.2.

Procedure 7. Classification of AD and CN. The data were extracted from skull-stripped 3D MR images same as with Procedure 6. However, data augmentation methods are utilized while training as following:

```
#Randomly rotate images in the range (degrees, 0 to 180)
$ rotation_range=20,
#Randomly zoom image
$ zoom_range = 0.15,
#Randomly shift images horizontally (fraction of total width)
$ width_shift_range=0.2,
#Randomly shift images vertically (fraction of total height)
$ height_shift_range=0.2,
```

Procedure 8. Replication of Procedure 7 data augmentation approach, with extracted data from Procedure 6.

Procedure 9. Classification of AD and CN. The data were extracted from skull-stripped 3D MR images same as with procedure 3. However, validation set is excluded. Custom CNN is created, trained on Alzheimer's data, then its trained layers are taken to retrieve features. Ultimately, features are used to feed random forest and test is predicted:

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 126, 126, 32)	896
max_pooling2d (MaxPooling2D)	(None, 63, 63, 32)	0
conv2d_1 (Conv2D)	(None, 61, 61, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 30, 30, 64)	0
conv2d_2 (Conv2D)	(None, 28, 28, 64)	36928
max_pooling2d_2 (MaxPooling2D)	(None, 14, 14, 64)	0
conv2d_3 (Conv2D)	(None, 12, 12, 128)	73856
flatten (Flatten)	(None, 18432)	0
dense (Dense)	(None, 64)	1179712
dense_1 (Dense)	(None, 1)	65

Total params: 1,309,953
Trainable params: 1,309,953
Non-trainable params: 0

(a) Custom network.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 126, 126, 32)	896
max_pooling2d (MaxPooling2D)	(None, 63, 63, 32)	0
conv2d_1 (Conv2D)	(None, 61, 61, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 30, 30, 64)	0
conv2d_2 (Conv2D)	(None, 28, 28, 64)	36928
max_pooling2d_2 (MaxPooling2D)	(None, 14, 14, 64)	0
conv2d_3 (Conv2D)	(None, 12, 12, 128)	73856
flatten (Flatten)	(None, 18432)	0

Total params: 130,176
Trainable params: 130,176
Non-trainable params: 0

(b) Feature extraction network.

Figure 6.4 Procedure 4: Creating custom network and using as feature extractor.

6.2 Simulating Wrong Implementations

In wrong approach, the Kaggle datasets are utilized to simulate wrong training procedure. This unknown sourced dataset is common among artificial intelligence beginners and Kagglers. More interestingly, published papers can also be found that using Kaggle dataset. The [39] includes many additional mistakes in the paper, and have over 40 cites.

We utilized Kaggle GPU (P100) to run, Tensorflow library to implement, and trained on ten different deep learning models for 50 epochs with early-stopping utility, and 400 batch size. Image size was chosen as (176, 176).

7

Performance Analysis

7.1 Standardized Results: 2D Slice-based

The 2D slice-based approach using deep learning models has been widely explored as a method for automated detection and classification of AD. This approach involves analyzing 2D slices of MRI images using convolutional neural networks (CNNs) to extract features and classify images as either AD or CN (cognitively normal). In this analysis, we will show the results of 2D slice-based approach using deep learning models for AD diagnosis and prediction, the limitations of these models, and the potential future directions for improving their performance. In Figs. 7.1-7.17, the classification reports and training details are provided for different models and datasets.

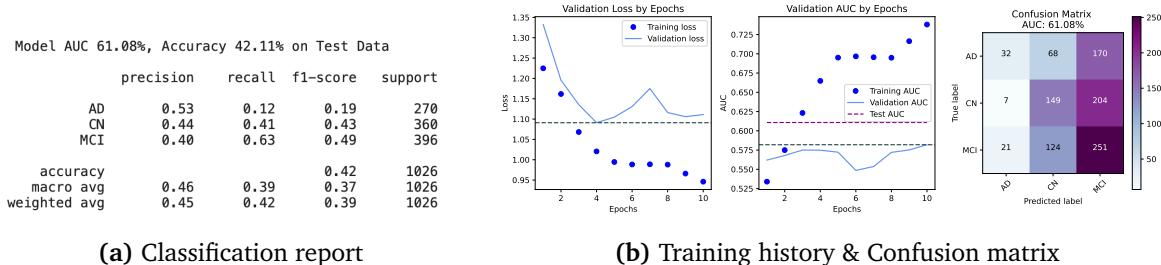


Figure 7.1 Procedure 1: Classification of AD, MCI and CN using VGG19.

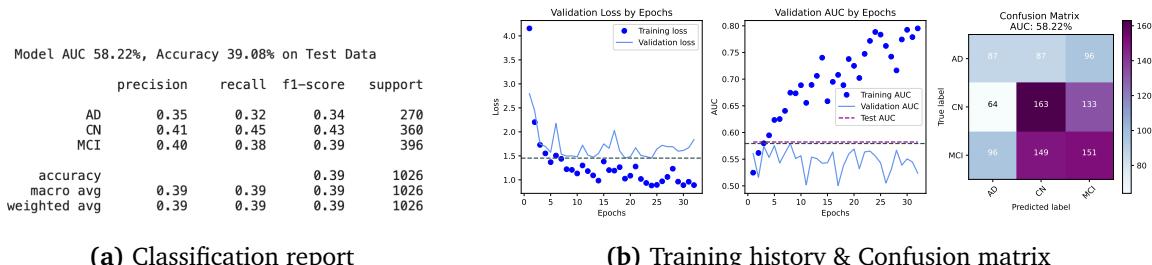


Figure 7.2 Procedure 1: Classification of AD, MCI and CN using InceptionV3.

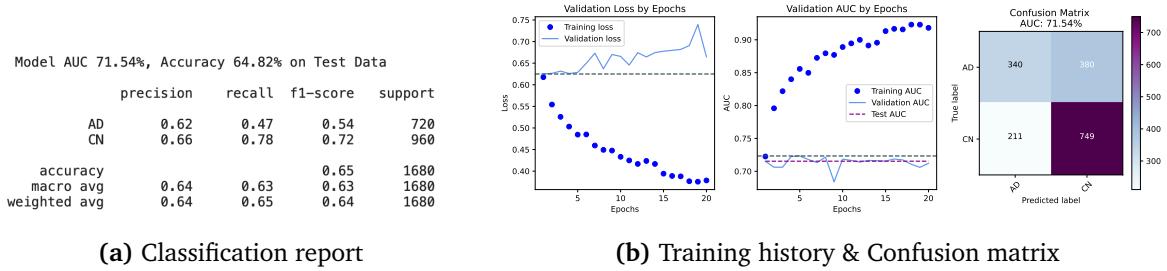


Figure 7.3 Procedure 2: Classification of AD and CN using VGG19.

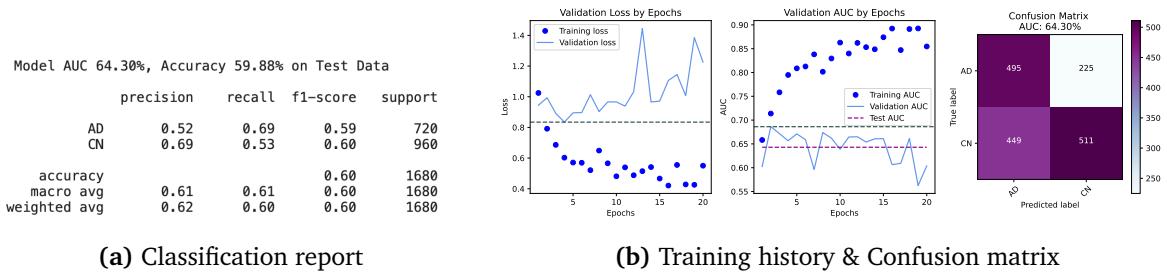


Figure 7.4 Procedure 2: Classification of AD and CN using InceptionV3.

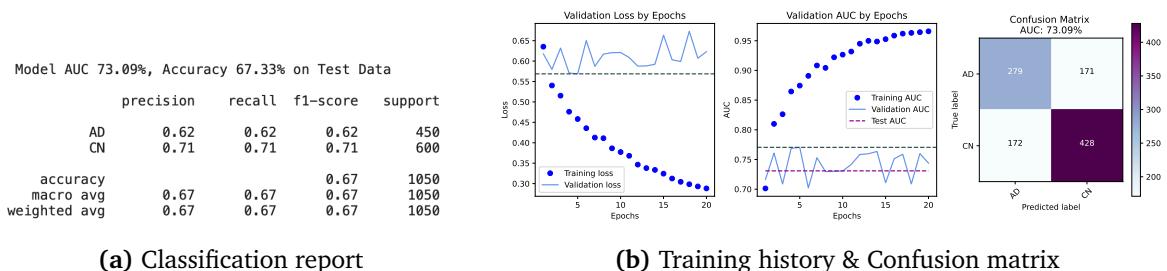


Figure 7.5 Procedure 3: Classification of AD and CN using VGG19.

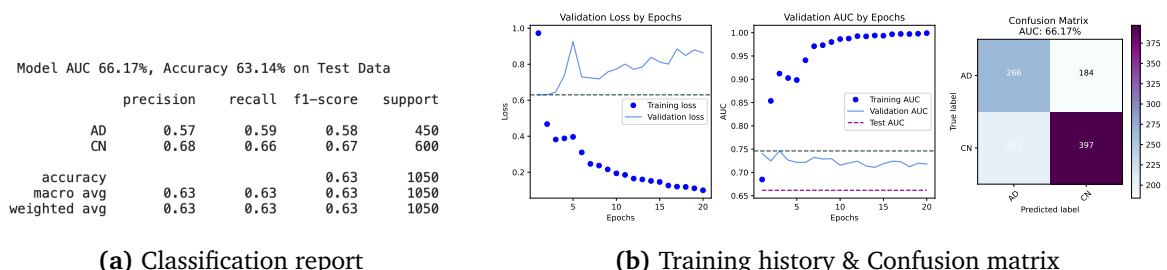


Figure 7.6 Procedure 3: Classification of AD and CN using InceptionV3.

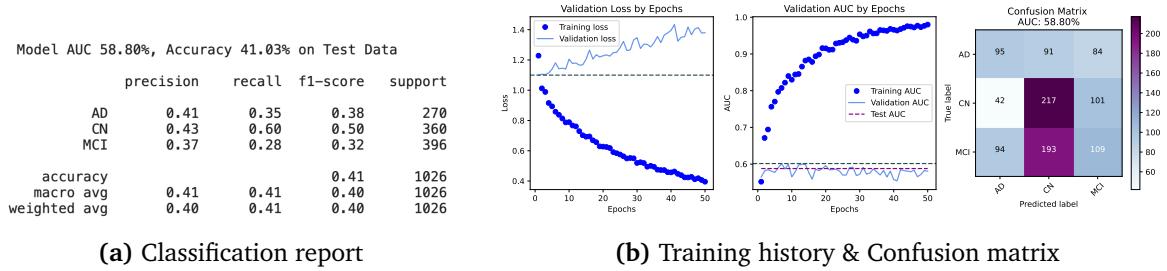


Figure 7.7 Procedure 4: Classification of AD, MCI and CN using VGG19.

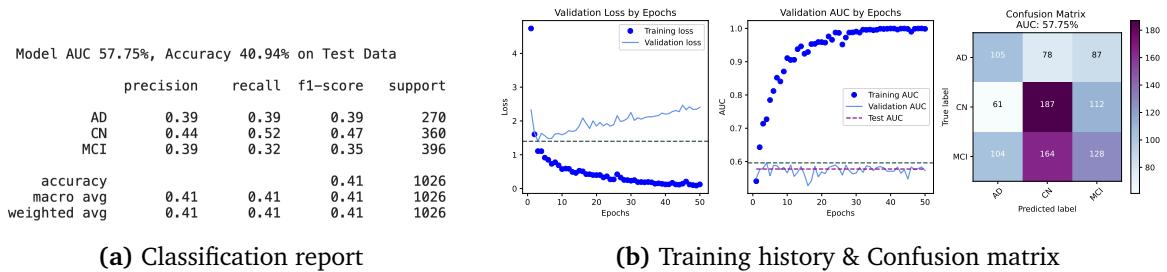


Figure 7.8 Procedure 4: Classification of AD, MCI and CN using InceptionV3.

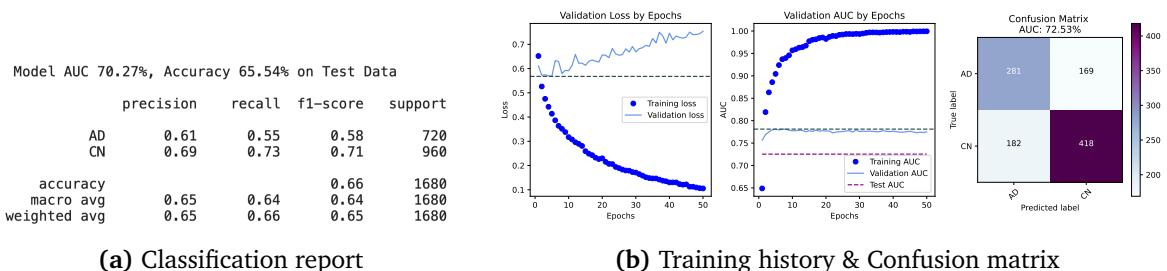


Figure 7.9 Procedure 5: Classification of AD and CN using VGG19.

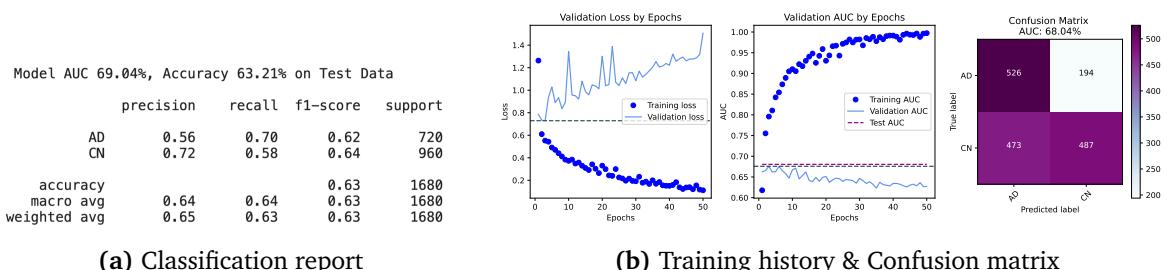


Figure 7.10 Procedure 5: Classification of AD and CN using InceptionV3.

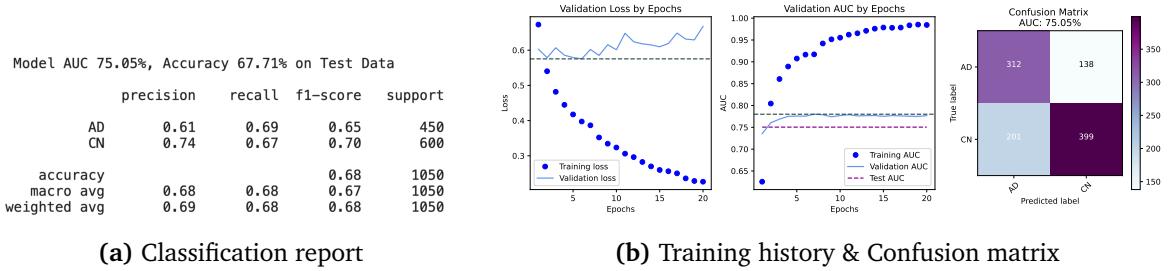


Figure 7.11 Procedure 6: Classification of AD and CN using VGG19.

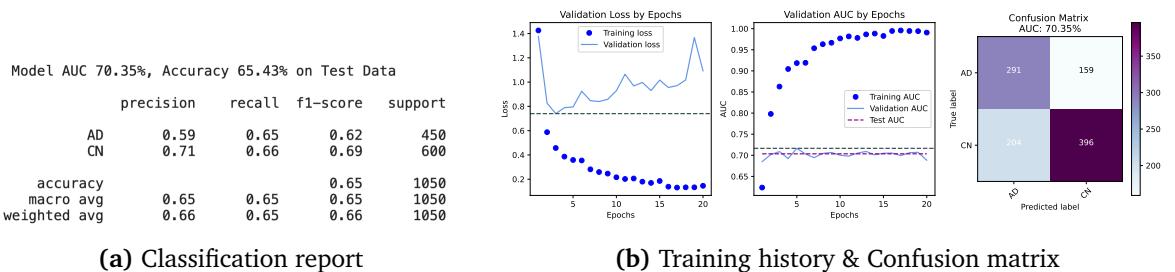


Figure 7.12 Procedure 6: Classification of AD and CN using InceptionV3.

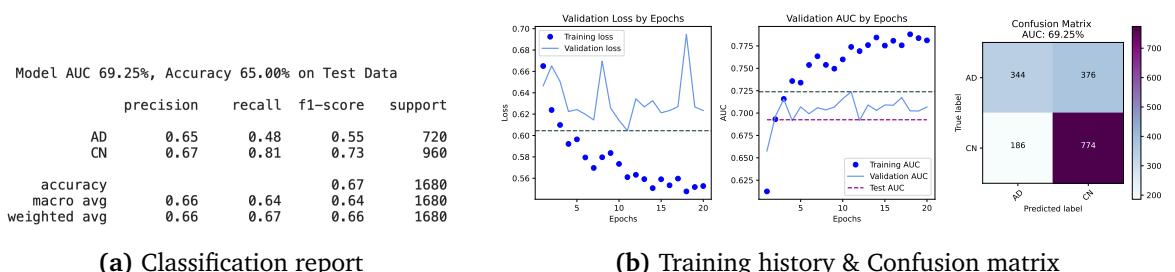


Figure 7.13 Procedure 7: Classification of AD and CN using VGG19.

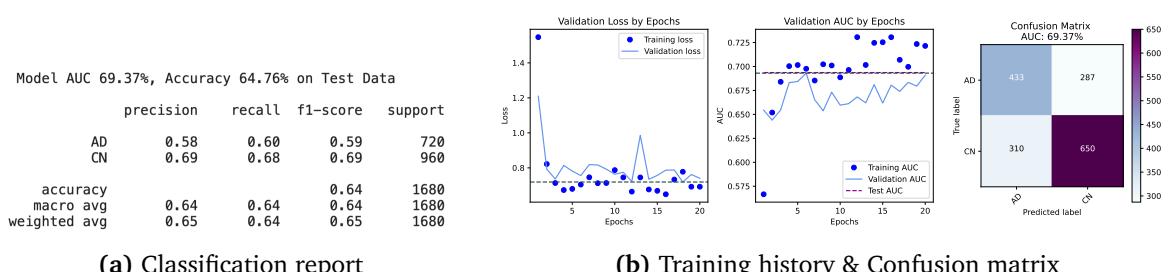


Figure 7.14 Procedure 7: Classification of AD and CN using InceptionV3.

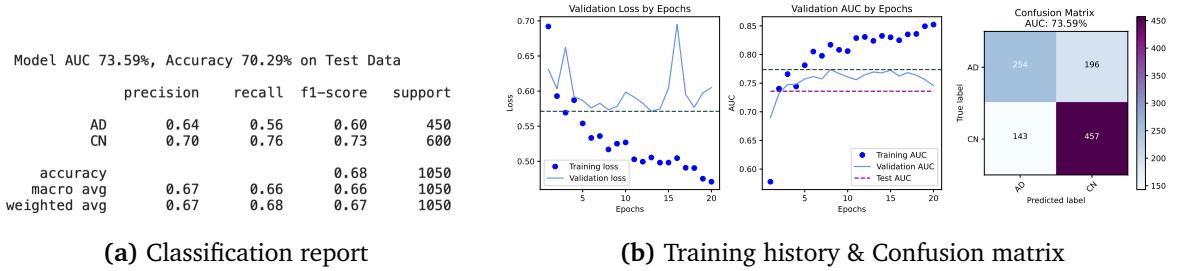


Figure 7.15 Procedure 8: Classification of AD and CN using VGG19.

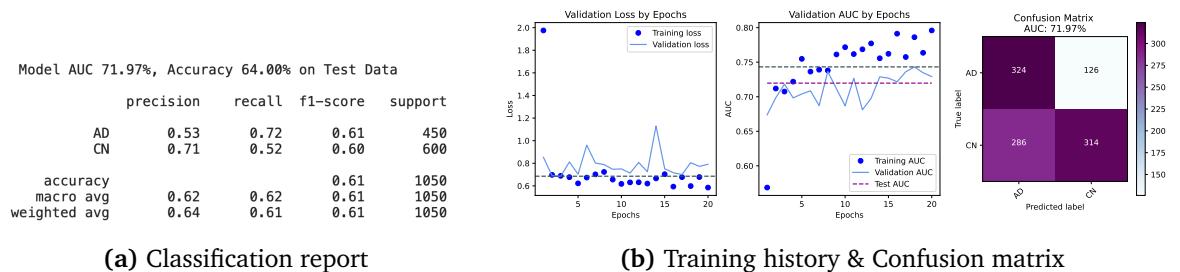


Figure 7.16 Procedure 8: Classification of AD and CN using InceptionV3.

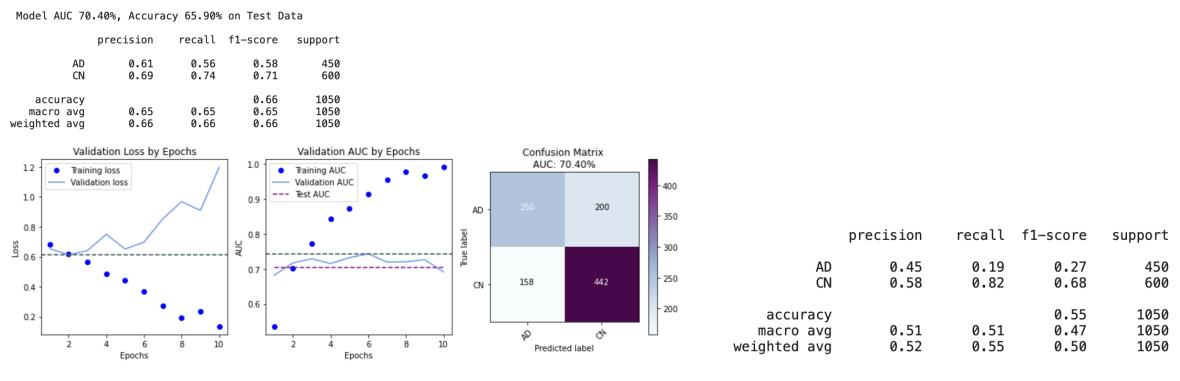


Figure 7.17 Procedure 9: Classification of AD and CN using custom network and random forest.

7.2 Simulating Wrong Analysis

To simulate wrong training task, we utilized the following deep learning models: VGG16, VGG19, ResNet50, ResNet 101, Xception, MobileNet, DenseNet121, DenseNet169, MobileNetV2 and InceptionV3. We trained the models with augmented samples without providing any original images. Entire original samples are utilized to test our trained models. Although the results are meaningless in terms of medical sciences, some interpretations might be extracted from the differences between response of the deep learning models to have deeper understanding about which model is more sensitive to data leakage. In Figs. 7.18-7.27, the classification reports and training details are provided for different models and datasets.

Table 7.1 Total traning time spend for different deep learning models.

Models	Total Time (min)
VGG16	78.84
VGG19	77.55
ResNet50	81.22
ResNet101	66.45
Xception	76.90
MobileNet	37.30
MobileNetV2	40.40
DenseNet121	101.72
DenseNet169	67.68
InceptionV3	48.15

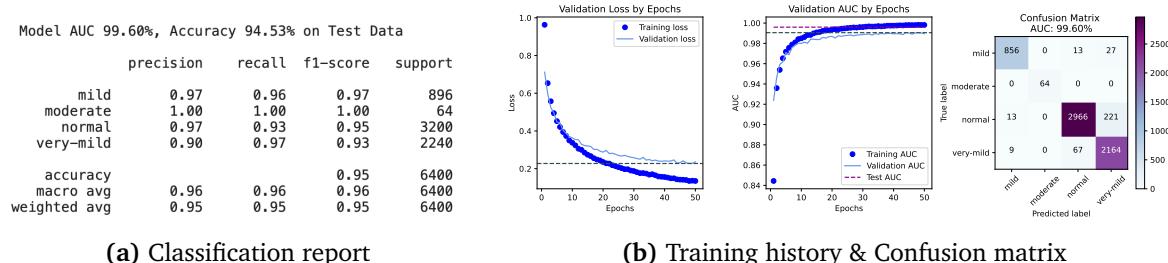


Figure 7.18 VGG16

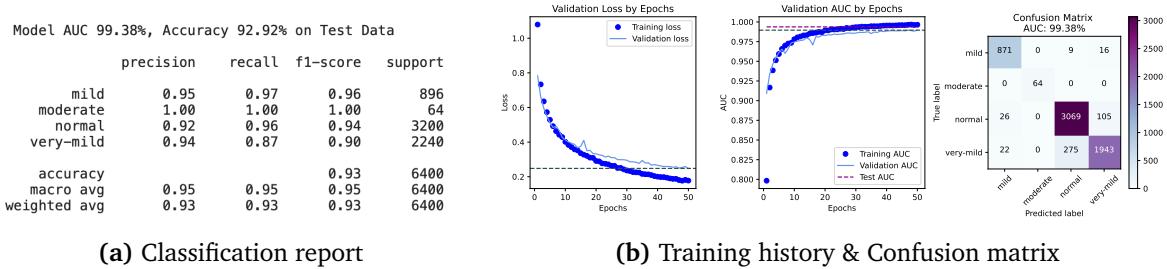


Figure 7.19 VGG19

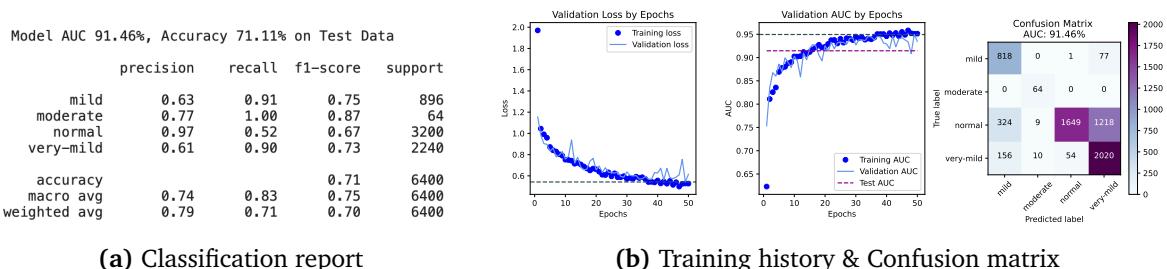


Figure 7.20 ResNet50

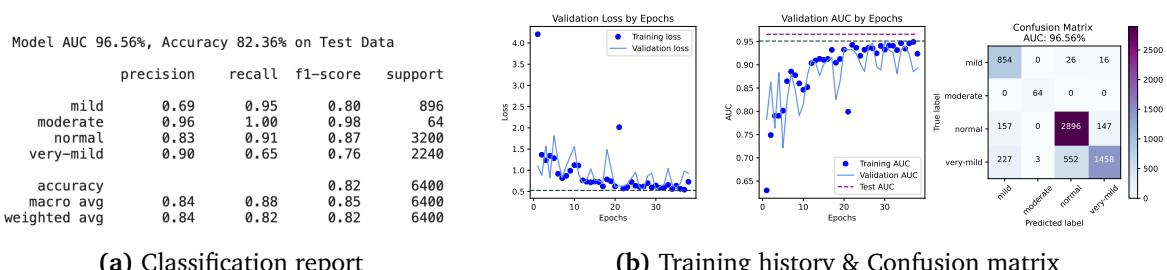


Figure 7.21 ResNet101

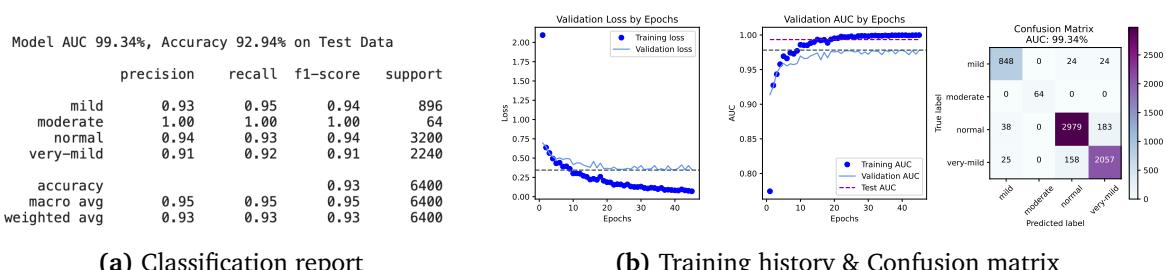


Figure 7.22 Xception

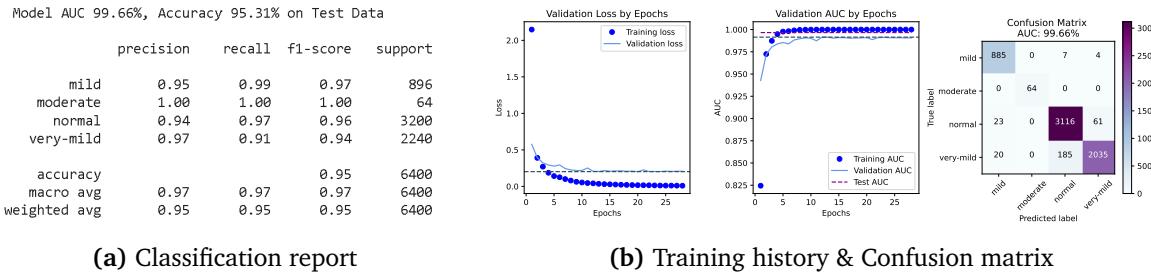


Figure 7.23 MobileNet

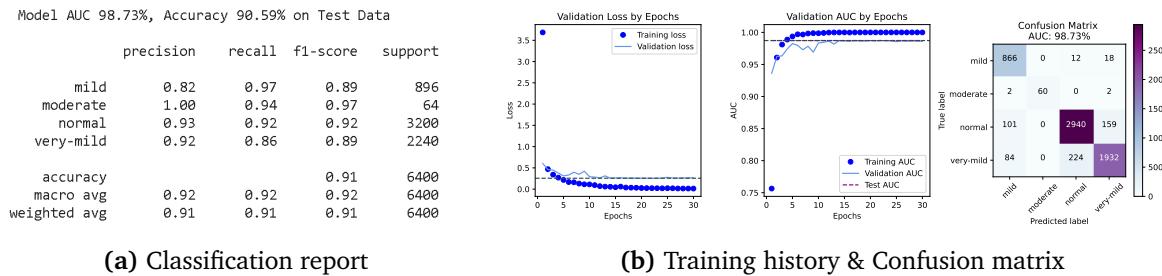


Figure 7.24 MobileNetV2

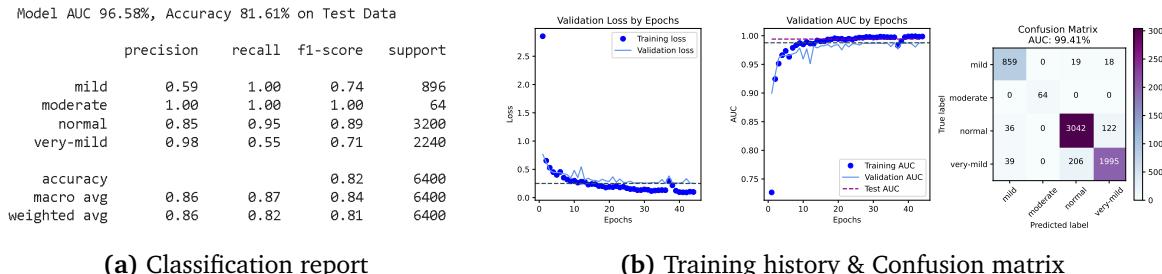


Figure 7.25 DenseNet121

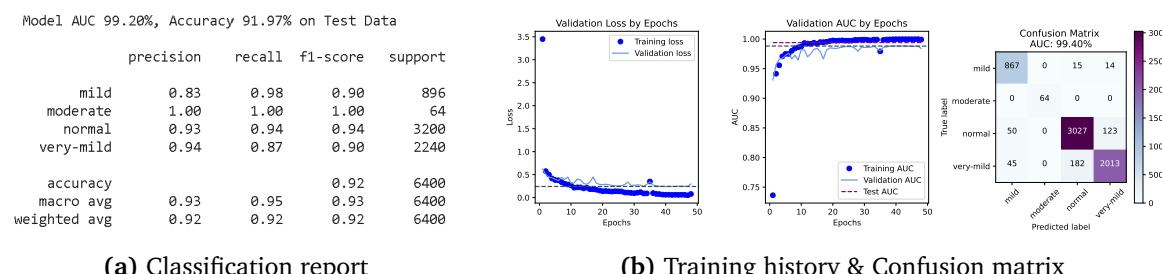
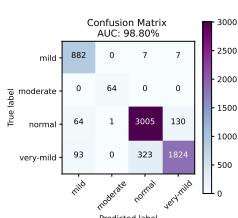
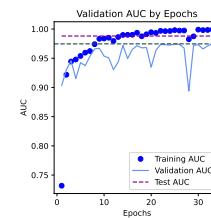
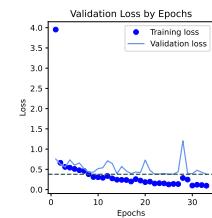


Figure 7.26 DenseNet169

Model AUC 98.16%, Accuracy 87.67% on Test Data

	precision	recall	f1-score	support
mild	0.73	0.98	0.84	896
moderate	0.98	1.00	0.99	64
normal	0.95	0.85	0.90	3200
very-mild	0.86	0.86	0.86	2240
accuracy			0.88	6400
macro avg	0.88	0.93	0.90	6400
weighted avg	0.89	0.88	0.88	6400



(a) Classification report

(b) Training history & Confusion matrix

Figure 7.27 InceptionV3

8

Discussion

8.1 Why is there a need to set particular standards?

Deep learning techniques have shown promise in improving our understanding of Alzheimer's disease (AD) and its diagnosis. However, the quality of data used to train deep learning models is critical in determining the accuracy and reliability of the model's predictions. Therefore, preparing the data with certain standards is essential to ensure that the results obtained from the model are meaningful and applicable in the context of AD.

The standards for data preparation in AD deep learning research are based on several factors, including the quality and quantity of the data, the preprocessing techniques used to clean and normalize the data, and the selection of appropriate features that are relevant to AD pathology. These standards help to ensure that the data is consistent, unbiased, and representative of the AD population being studied. Furthermore, they help to minimize the risk of overfitting and improve the generalizability of the model's predictions.

Adhering to these standards can facilitate the clinical translation of deep learning models in AD research. For instance, deep learning models have been developed to predict the progression of AD or to distinguish AD patients from healthy individuals using various neuroimaging modalities. Models developed using high-quality data that meet these standards can be integrated into clinical workflows to support decision-making by healthcare providers.

Another benefit of adhering to these standards is the potential for reproducibility. Deep learning models developed using standardized data preparation techniques can be easily reproduced by other researchers, allowing for the validation and refinement of the model's predictions. This can help build confidence in the accuracy and reliability of the model's predictions and accelerate the translation of the model into clinical practice.

However, obtaining high-quality data in AD research can be challenging due to the variability of AD pathology and disease progression. Additionally, preprocessing techniques can introduce bias into the data, which can heavily impact the accuracy and reliability of the model's predictions. Therefore, it is essential to carefully evaluate and validate the data preparation techniques used in AD deep learning research to ensure that they are appropriate and effective.

8.2 Can we extract useful information from wrong approach?

The results in the Section 7.2 provided no information in clinical terms. Moreover, almost all models showed overfitting, and over %96 AUC and %90 accuracy. However, scores might be used to differentiate practical responses of 10 different deep learning models to the dataset that contains data leakage. In training set, no original image had shown to the models. Nevertheless, deep pre-trained models, although to varying degrees, could figure the origins of the training samples out during testing. For instance in ResNet50 (Fig. 7.20), AUC score was 91.46% and accuracy score was 71.11%, whereas MobileNet (Fig. 7.23) had AUC 99.66% and accuracy 95.31%. The difference between ResNet50 and MobileNet may be providing that MobileNet have deeper understanding that the test samples were used as training in another formation, while ResNet50 had little clue. Xception, VGG16, VGG19, DenseNet169, MobileNetv2 and InceptionV3 models might be learning the patterns of the origins of augmented data, whereas DenseNet121, ResNet50 and ResNet101 have lesser clue about the origins of the original MRI slices. The network size also affected the learning patterns that can be interpreted by evaluating the results of ResNet50 and ResNet101 and DenseNet121 and DenseNet169.

8.3 Study limitations

The Alzheimer's Disease Neuroimaging Initiative (ADNI) is a large multi-center study that has collected extensive neuroimaging data, including 3D MRI, from individuals with Alzheimer's disease (AD) and healthy controls. While the ADNI data has been a valuable resource for academic research in AD, there are certain limitations that should be considered when using this data.

One limitation of using ADNI 3D MRI data is the potential for selection bias. The ADNI cohort includes individuals with mild cognitive impairment, AD, and healthy controls who were selected based on specific inclusion and exclusion criteria. This can limit the generalizability of findings to the broader AD population, as the ADNI cohort may not be representative of the wider AD population.

Another limitation of using ADNI 3D MRI data is the potential for variation over time in image acquisition protocols and processing techniques. Although efforts have been made to standardize the imaging protocols and processing pipelines, variations in scanner hardware, software, and imaging parameters can lead to differences in image quality, dimension distribution and resolution. This can impact the accuracy and reliability of the results obtained from the data.

A further limitation of using ADNI 3D MRI data is the potential for missing data or incomplete follow-up (not a case for our study). The ADNI study has a longitudinal design, with participants undergoing repeated imaging and clinical assessments over time. However, participants may drop out of the study or miss certain timepoints, leading to missing data or incomplete follow-up. This can limit the ability to investigate disease progression or treatment response in the ADNI cohort.

One of the major limitations of our study is that we have opted for a 2D slice-based approach to implement the task. This decision is associated with a critical issue in that, despite the 2D approach allowing us to extract sections from the 3D volume, the manifestation of the disease may not be apparent in the extracted sections, even if the individual is afflicted with the ailment. This inherent problem is of paramount significance, as it may introduce considerable bias and hinder the accuracy of our findings. Conversely, a 3D subject-based convolutional neural network (CNN) obviates this issue, as the entire volume is evaluated holistically, and the literature has recently emphasized the superiority of such approaches. However, replicating and validating the majority of the 3D studies is a daunting task, which exceeds the technical and human resources at our disposal in this project.

Finally, another limitation of using ADNI 3D MRI data is the potential for ethical concerns related to data sharing and participant privacy. The ADNI study is a publicly available dataset, and researchers must adhere to strict data sharing and usage agreements. Additionally, participants may have concerns about their privacy and confidentiality, which can impact the ability to obtain informed consent or to share data outside of the ADNI consortium.

9

Conclusion

In conclusion, adhering to certain standards for data preparation in AD deep learning research is crucial for obtaining meaningful and clinically relevant results. These standards help to ensure the quality, consistency, and representativeness of the data used to train deep learning models, which can improve the accuracy and reliability of the model's predictions. While there are challenges associated with adhering to these standards, they are essential for building confidence in the model's predictions and accelerating their translation into clinical practice for AD diagnosis and management.

References

- [1] H. A. Helaly, M. Badawy, and A. Y. Haikal, “Deep learning approach for early detection of alzheimer’s disease,” *Cognitive Computation*, vol. 14, no. 5, pp. 1711–1727, Nov. 2021. doi: 10.1007/s12559-021-09946-2. [Online]. Available: <https://doi.org/10.1007/s12559-021-09946-2>.
- [2] J. Wen *et al.*, “Convolutional neural networks for classification of alzheimer’s disease: Overview and reproducible evaluation,” *Medical Image Analysis*, vol. 63, p. 101694, May 2020. doi: 10.1016/j.media.2020.101694.
- [3] E. Altinkaya, K. Polat, and B. Barakli, “Detection of alzheimer’s disease and dementia states based on deep learning from mri images: A comprehensive review,” 2020.
- [4] T. Lancet, “The three stages of alzheimer’s disease,” *The Lancet*, vol. 377, no. 9776, p. 1465, Apr. 2011. doi: 10.1016/s0140-6736(11)60582-5. [Online]. Available: [https://doi.org/10.1016/s0140-6736\(11\)60582-5](https://doi.org/10.1016/s0140-6736(11)60582-5).
- [5] S. Spasov, L. Passamonti, A. Duggento, P. Lio, and N. Toschi, “A multi-modal convolutional neural network framework for the prediction of alzheimer’s disease,” vol. 2018, Jul. 2018, pp. 1271–1274. doi: 10.1109/EMBC.2018.8512468.
- [6] “2018 alzheimer’s disease facts and figures,” *Alzheimer’s & Dementia*, vol. 14, no. 3, pp. 367–429, Mar. 2018. doi: 10.1016/j.jalz.2018.02.001. [Online]. Available: <https://doi.org/10.1016/j.jalz.2018.02.001>.
- [7] Wurina, Y.-F Zang, and S.-G. Zhao, “Resting-state fmri studies in epilepsy,” *Neuroscience bulletin*, vol. 28, pp. 449–55, Aug. 2012. doi: 10.1007/s12264-012-1255-1.
- [8] A. Fagan and D. Holtzman, “Perrin rj, fagan am, holtzman dm. multimodal techniques for diagnosis and prognosis of alzheimer’s disease. nature 461: 916-922,” *Nature*, vol. 461, pp. 916–22, Oct. 2009. doi: 10.1038/nature08538.
- [9] J. Barnes *et al.*, “Vascular and alzheimer’s disease markers independently predict brain atrophy rate in alzheimer’s disease neuroimaging initiative controls,” *Neurobiology of aging*, vol. 34, Mar. 2013. doi: 10.1016/j.neurobiolaging.2013.02.003.
- [10] G. McKhann, D. Knopman, and H. Chertkow, “The diagnosis of dementia due to alzheimer’s disease: Recommendations from the national institute on aging and the alzheimer’s association workgroup,” *Alzheimer’s & Dementia: Journal of the Alzheimer’s Association*, vol. 7, Jan. 2011.

- [11] B. Dubois and M. L. Albert, "Amnestic MCI or prodromal alzheimer's disease?" *The Lancet Neurology*, vol. 3, no. 4, pp. 246–248, Apr. 2004. DOI: 10.1016/s1474-4422(04)00710-0. [Online]. Available: [https://doi.org/10.1016/s1474-4422\(04\)00710-0](https://doi.org/10.1016/s1474-4422(04)00710-0).
- [12] A. Segato, A. Marzullo, F. Calimeri, and E. D. Momi, "Artificial intelligence for brain diseases: A systematic review," *APL Bioengineering*, vol. 4, no. 4, p. 041503, Dec. 2020. DOI: 10.1063/5.0011697. [Online]. Available: <https://doi.org/10.1063/5.0011697>.
- [13] A. Payan and G. Montana, "Predicting alzheimer's disease: A neuroimaging study with 3d convolutional neural networks," *ICPRAM 2015 - 4th International Conference on Pattern Recognition Applications and Methods, Proceedings*, vol. 2, Feb. 2015.
- [14] S. Sarraf and G. Tofighi, "Classification of alzheimer's disease structural MRI data by deep learning convolutional neural networks," *CoRR*, vol. abs/1607.06583, 2016. arXiv: 1607.06583. [Online]. Available: <http://arxiv.org/abs/1607.06583>.
- [15] E. Hosseini-Asl, R. Keynton, and A. El-Baz, "Alzheimer's disease diagnostics by adaptation of 3d convolutional network," *CoRR*, vol. abs/1607.00455, 2016. arXiv: 1607.00455. [Online]. Available: <http://arxiv.org/abs/1607.00455>.
- [16] S. Korolev, A. Safiullin, M. Belyaev, and Y. Dodonova, "Residual and plain convolutional neural networks for 3d brain mri classification," Jan. 2017. DOI: 10.1109/ISBI.2017.7950647.
- [17] S. Wang, P. Phillips, Y. Sui, B. Liu, M. Yang, and H. Cheng, "Classification of alzheimer's disease based on eight-layer convolutional neural network with leaky rectified linear unit and max pooling," *Journal of Medical Systems*, vol. 42, Mar. 2018. DOI: 10.1007/s10916-018-0932-7.
- [18] A. Khvostikov, K. Aderghal, A. Krylov, G. Catheline, and J. Benois-Pineau, "3d inception-based cnn with smri and md-dti data fusion for alzheimer's disease diagnostics," Jul. 2018. DOI: 10.13140/RG.2.2.30737.28006.
- [19] I. Sahumbaiev, A. Popov, J. Ramírez, J. Gorriz, and A. Ortiz, "3d-cnn hadnet classification of mri for alzheimer's disease diagnosis," Nov. 2018, pp. 1–4. DOI: 10.1109/NSSMIC.2018.8824317.
- [20] J. Ruiz, M. Mahmud, M. Modasshir, M. S. Kaiser, and f. Initiative, "3d densenet ensemble in 4-way classification of alzheimer's disease," in Sep. 2020, pp. 85–96, ISBN: 978-3-030-59276-9. DOI: 10.1007/978-3-030-59277-6_8.
- [21] N. Jain, A. Aggarwal, N. Jain, and M. Noordian, "Convolutional neural network based alzheimer's disease classification from magnetic resonance brain images," Jan. 2019.
- [22] C. Ge, Q. Qu, I. Y.-H. Gu, and A. Store Jakola, "Multiscale deep convolutional networks for characterization and detection of alzheimer's disease using mr images," in *2019 IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 789–793. DOI: 10.1109/ICIP.2019.8803731.

- [23] T.-A. Song *et al.*, “Graph convolutional neural networks for alzheimer’s disease classification,” vol. 2019, Apr. 2019, pp. 414–417. DOI: 10.1109/ISBI.2019.8759531.
- [24] M. Liu *et al.*, “A multi-model deep convolutional neural network for automatic hippocampus segmentation and classification in alzheimer’s disease,” *NeuroImage*, vol. 208, p. 116459, Dec. 2019. DOI: 10.1016/j.neuroimage.2019.116459.
- [25] H. Parmar, B. Nutter, L. Long, S. Antani, and S. Mitra, “Spatiotemporal feature extraction and classification of alzheimer’s disease using deep learning 3d-cnn for fmri data,” *Journal of Medical Imaging*, vol. 7, Oct. 2020. DOI: 10.1117/1.JMI.7.5.056001.
- [26] S. Basaia *et al.*, “Automated classification of alzheimer’s disease and mild cognitive impairment using a single mri and deep neural networks,” *NeuroImage: Clinical*, vol. 21, p. 101645, Dec. 2018. DOI: 10.1016/j.nicl.2018.101645.
- [27] J. Wen *et al.*, “Convolutional neural networks for classification of alzheimer’s disease: Overview and reproducible evaluation,” *Medical image analysis*, vol. 63, p. 101694, 2020.
- [28] A. Hoopes, J. S. Mora, A. V. Dalca, B. Fischl, and M. Hoffmann, “SynthStrip: Skull-stripping for any brain image,” *NeuroImage*, vol. 260, p. 119474, Oct. 2022. DOI: 10.1016/j.neuroimage.2022.119474. [Online]. Available: <https://doi.org/10.1016/j.neuroimage.2022.119474>.
- [29] B. Fischl, “FreeSurfer,” *NeuroImage*, vol. 62, no. 2, pp. 774–781, Aug. 2012. DOI: 10.1016/j.neuroimage.2012.01.021. [Online]. Available: <https://doi.org/10.1016/j.neuroimage.2012.01.021>.
- [30] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [31] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [32] C. Szegedy *et al.*, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.
- [33] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.
- [34] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.
- [35] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.
- [36] J.-B. Cordonnier, A. Loukas, and M. Jaggi, “On the relationship between self-attention and convolutional layers,” *arXiv preprint arXiv:1911.03584*, 2019.

- [37] A. Dosovitskiy *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.
- [38] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, “Transformers in vision: A survey,” *ACM computing surveys (CSUR)*, vol. 54, no. 10s, pp. 1–41, 2022.
- [39] S. Murugan *et al.*, “Demnet: A deep learning model for early diagnosis of alzheimer diseases and dementia from mr images,” *IEEE Access*, vol. 9, pp. 90 319–90 329, 2021. DOI: 10.1109/ACCESS.2021.3090474.

Curriculum Vitae

FIRST MEMBER

Name-Surname: Toygar Tanyel

Birthdate and Place of Birth: 14.04.2000, İzmir

E-mail: toygar.tanyel@std.yildiz.edu.tr

Phone: 546 416 2000

Practical Training:

Title: Research Assistant (ITU)

Project: Tubitak, PIMI Lab.

Advisor: Assoc. Prof. İlkay Öksüz

Mar 2023 to Present

Title: Research Assistant (YTU)

Project: Tubitak 2232, IHIRC Lab.

Advisor: Dr. Bilgin Keserci

Sep 2022 to Present

Title: Research Intern (Boğaziçi Univ.)

Project: Tubitak

Advisor: Prof. Dr. Burak Güçlü

Aug 2022 to Nov 2022

Title: Research Engineer

Project: Tubitak TEYDEB, Tapin2Consumer

Company: Optdcom

Feb 2022 to Aug 2022

Title: Research Intern (YTU)

Project: Tubitak 1001, Sort Optimization of Training Samples in Machine Learning

Advisor: Prof. Dr. M. Fatih Amasyali

Nov 2021 to Jun 2022

Title: Undergraduate Researcher (YTU)

Project: BAP Research Project, Autonomous rover design for extraterrestrial life trace research

Nov 2020 to Oct 2021

Project System Informations

System and Software: Windows/MacOS/Linux Operating System, Computer Vision, Machine Learning & Deep Learning, Python, Medical Knowledge

Required RAM: 16GB

Required Disk: 200GB+