Alma Mater Studiorum - Università di Bologna

DOTTORATO DI RICERCA IN

INGEGNERIA BIOMEDICA, ELETTRICA E DEI SISTEMI

Ciclo 34

Settore Concorsuale: 09/G2 - BIOINGEGNERIA

Settore Scientifico Disciplinare: ING-INF/06 - BIOINGEGNERIA ELETTRONICA E INFORMATICA

DEVELOPMENT AND CHARACTERIZATION OF DEEP LEARNING TECHNIQUES FOR NEUROIMAGING DATA

Presentata da: Selamawet Workalemahu Atnafu

Coordinatore Dottorato

Michele Monaci

Supervisore

Stefano Diciotti

Co-supervisore

Mauro Ursino

Esame finale anno 2022

Abstract

Deep learning methods are extremely promising machine learning tools to analyze neuroimaging data. However, their potential use in clinical settings is limited because of the existing challenges of applying these methods to complex, high-dimensional neuroimaging data. In fact, by surveying the literature, data scarsity, data leakage, interpretability and reproducibility are the pitfalls of the existing deep learning systems that are designed to analyze neuroimaging data. The survey also discovered a widely spreaded type of data leakage caused by splicing volumetric magnetic resonance image data based on individual 2D slices, for the purpose of training and validating a 2D CNN, leading to erroneous overestimated model performances. However, this type of data leakage has given less attention by the neuroimaging research community. Although there are a few deep learning tools that perform leakage-free pre-processing and provide an effective model training platform, due to the abscence of explainability feature, they are barely trusted by the clinicials.

Hence, the goal of this PHD is first, to quantitatively assess the extent to which a biased model outputs an overestimated performance due to the presence of data leakage.

Second, an openly available, interpretable, and leakage-free deep learning software that is versatile enough to be used by many researchers is developed. The software is written in a python language and is developed using Keras framework, with Tensorflow backend, and other python packages to conduct both classification and regression analysis. In addition, it has a wide range of options in terms of model architectures, model training, and validation schemes including nested cross validation that allows model selection, hyperparametric optimization and unbiased model evaluation.

The software was applied to the study of mild cognitive impairment (MCI) in patients with small vessel disease (SVD) using multi-parametric MRI data, including T1-weighted, T2-weighted FLAIR and feature maps [fractional anisotropy (FA) and mean diffusivity (MD)] extracted from diffusion tensor imaging (DTI). The cognitive performance of 58 patients with MCI and SVD measured by five neuropsychological tests (MoCA, SDMA, TMT-A, Stroop, ROC-F and Visual

search) is predicted using a multi-input CNN model taking brain image and demographic data. Each of the cognitive test scores was predicted using different MRI-derived features. As MCI due to SVD has been hypothesized to be the effect of white matter damage, DTI-derived features MD and FA produced the best prediction outcome of the TMT-A score which is inline with the hypothesis in the literature.

In a second study, an interpretable deep learning system aimed at 1) classifying Alzheimer disease and healthy subjects 2) examining the neural correlates of the disease that causes a cognitive decline in AD patients using CNN visualization tools and 3) highlighting the potential of interpretability techniques to capture a biased deep learning model is developed. Structural magnetic resonance imaging (MRI) data of 200 subjects (100 AD and 100 HC) obtained from OASIS dataset was used by the proposed CNN model. The model was trained using a transfer learning-based approach in a 5-fold cross-validation loop producing a balanced accuracy of 71.6%. Brain regions in the frontal and parietal lobe showing the cerebral cortex atrophy were highlighted by the visualization tools.

Acknowledgements

I would first of all like to thank my supervisor Prof. Stefano Diciotti, who has been an ideal teacher. mentor and thesis supervisor, offering precious support and encouragement with a perfect blend of insight and humor. I am proud of, and greatful for, my time with Prof. Diciotti, without whom competent guidance this work would not have been possible.

I would also like to thank my co-supervisor Prof. Mauro Ursino for his valuable advice and support in completing my work.

A delightful thanks also goes to Prof. Alba García Seco de Herrera and Prof. Luca Citi and all their collaborators, for the opportunity granted me to work in collaboration with them as part of my Ph.D. internship.

A heartful thanks also go to Dr. Marco Giannelli, Dr. Carlo Tessa, Dr. Emilia Salvadori, Prof. Anna Poggesi, Prof. Antonio Giorgio, Prof. Nicola De Stefano, Prof. Leonardo Pantoni, and Prof. Mario Mascalchi for giving me a chance to work and collaborate with them.

Finally, I would like to acknowledge all my colleagues in Bio-engineering for their hospitality and for the great time we had together.

Contents

List of figures

List of tables

List of acronyms

1D, one dimensional

2D, two dimensional

3D, three dimensional

4D, four dimensional

AD, Alzheimer's disease

ADNI, Alzheimer’s Disease Neuroimaging Initiative

AE, auto encoder

AI, artificial intelligence

ANN, artificial neural networks

ASD, autism spectrum disorder

AUC, area under ROC curve

CDR, Clinical Dementia Rating

CN, cognitively normal

CNN, convolutional neural network

CT, computed tomography

CV, cross validation

DAT, dopamine transporter

DBN, deep belief network

DCGAN, deep convolutional generative adversarial networks

DFWG, data format working group

DLTK, deep learning toolkit for medical imaging

DTI, diffusion tensor imaging

DWI, diffusion weighted image

EMD, earth Mover's Distance

FA, fractional anisotropy

FC, fully connected

FLAIR, fluid-attenuated inversion recovery

fMRI, functional magnetic resonance imaging

FOV, field of view

FPR, false positive rate

GAN, generative adversarial network

GAP, global average pooling

HC, healthy controls

ID, identification number

ILSVRC, Large Scale Visual Recognition Challenge

LM, logical memory

MCI, mild cognitive impairment

MD, mean diffusivity

MLP, multilayer perceptron

MMSE, Mini-Mental State Examination

MoCA, Montreal Cognitive Assessment

MPRAGE, Magnetization Prepared Rapid Gradient Echo

MR, magnetic resonance

MRI, magnetic resonance imaging

MSE, mean squared error

NEX, number of excitations

Nifti, neuroimaging informatics technology initiative

NIH, national institute of health

OASIS, Open Access Series of Imaging Studies

PD, Parkinson's disease

PET, positron emission tomography

PPMI, Parkinson's Progression Markers Initiative

ReLU, rectified linear unit

ResNet, residual neural network

ROC, receiver operating characteristics

ROCF, Rey-Osterrieth complex figure

ROI, region of interest

Rs-fMRI, resting state functional magnetic resonance imaging

SAE, stacked auto encoder

SD, standard deviation

SDMT, symbol digit modalities test

SGD, stochastic gradient descent

SHAP, shapley additive explanations

sMRI, structural magnetic resonance imaging

SPECT, single positron emission tomography

SVD, small vessel disease

SVM, support vector machine

SWEDD, scans without evidence of dopaminergic deficit

TBI, traumatic brain injury

TD, delay time

TD, typically developing

TE, echo time

TI, inversion time

TMT-A, trial making test part A

TPR, true positive rate

TR, repetition time

VGG, visual geometry group

VMAT-2, vesicular monoamine transporter type 2

VMCI, vascular mild cognitive impairment

WGAN, Wasserstein generative adversarial networks

WM, white matter

Chapter 1

1. Introduction

1.1 Imaging the structure and function of the brain

Medical imaging has become a standard tool for examining the structure, function and pathology of the human brain. It allows increasing our knowledge of how the brain and the other parts of the nervous system work and what structural or functional changes may be associated with a given clinical presentation of a disease or medical condition(Kassubek, 2017). The use of various neuroimaging techniques, such as magnetic resonance imaging (MRI), positron emission tomography (PET), and single positron emission tomography (SPECT) among the popular ones, for the *in vivo* investigation of neurological disorders, have increased substantially(Young et al., 2020). These tools have been used in the prediction, diagnosis, and monitoring of disease progression. It has also been helping to define imaging bio-markers that can be employed to infer structural and functional brain alterations associated with various neurological disorders. These imaging markers may be used primarily for early diagnosis, planning treatment strategies, assessing its effects, and tracking disease progression. Various studies have employed neuroimaging techniques combined with a variety of analysis methods to illustrate the association between the changes in clinical measures and structural and functional alterations of the brain caused by different neurological diseases, such as mild cognitive impairment (MCI), Alzheimer's disease (AD) and Parkinson's disease (PD)(Yin et al., 2013, Ibarretxe-Bilbao et al., 2009).

The standard machine learning techniques used to analyze neuroimaging data, such as sparse learning, support vector machine (SVM), Gaussian networks, random forest, decision tree, hidden Markov model, etc., generally require four steps: feature extraction, feature selection, dimensionality reduction, and feature-based classification or regression algorithm selection(Jo et al., 2019). One of the limitations of these approaches is the need for defining and manually crafting features based on a domain-specific knowledge that could represent the disease pathology. However, since medical images are very complex, the manual feature selection step is not convenient for non-expert users. Another drawback is that, due to their shallow architectures, conventional machine learning methods have less representational power to analyze high-

dimensional medical images, especially brain images(Pandya et al., 2019). Hence the effectiveness of neuroimaging to aid in clinical setup greatly depends on the use of more efficient data analysis methods.

1.2 Neuroimaging and deep learning

In recent years, the use of artificial intelligence (AI) methods with great representational power capable of analyzing large scale, high dimensional, complex raw neuroimaging data to generate features automatically has been attracting considerable attention of the neuroimaging research community(Plis et al., 2014). Deep learning, a machine learning approach that allows a model to automatically learn patterns from the raw input data, is a family of representation learning methods modeled by a combination of non-linear but simple functions or modules, hence can model very complex functions. The first simple module produces a representation of the raw input data, and each consecutive module hierarchically transforms the representation coming from the lower level into a slightly more abstract level(LeCun et al., 2015). Deep learning methods are known by their specific architecture, namely some form of neural networks, which are, to some extent, inspired by the structure of the human brain(Zaharchuk et al., 2018).

There are several advantages of using deep learning techniques for neuroimaging data analysis:

1. While standard machine learning methods extract features based on some *a priori* knowledge, which can only extract some features associated with a specific application, deep learning can find new features that are suitable to specific applications but have never been previously discovered by researchers(Liu et al., 2018b);
2. Deep learning methods can support better data interpretation and supervision, which can assist the physicians efficiently(Karthik et al., 2020). This is because they have great potential in capturing hidden representations and automatically extracting features, especially from the complex neuroimaging data(Karthik et al., 2020);
3. Their great representational power makes deep learning approaches convenient for analyzing the complex neuroimaging data, as the pathology-specific associations are embedded at intricate abstract levels. Hence, these methods can outperform traditional machine learning methods substantially and particularly well, presenting a lower

asymptotic complexity in relative computational time, despite being more complex in their architecture and parameterization(Abrol et al., 2021); and

4. The training phase of deep learning approaches often involves the automatic and adaptive discovery of discriminative data representations at multiple levels of hierarchy in an end-to-end (input to output) learning procedure. Application of this radically different approach in an end-to-end manner can also have a provision backward mapping to the input image space through methodical interpretations, thus possibly allowing us to make inferences about brain mechanisms, for example, delineating the features in the input space that are most influential in predicting an attempted task. On the contrary, relevant spatial relationships may be lost at the dimensionality reduction stage, arguably, required for standard machine learning methods to work(Abrol et al., 2021).

1.3 Workflow of deep learning for neuroimaging data

1.3.1 Data acquisition

Data acquisition is the first step in any statistical image processing procedure. In neuroimaging, the data acquired includes clinically measured test scores, such as the cognitive status as measured by the Montreal cognitive assessment (MoCA) score, demographic variables of each participant, including age, sex, weight, education, race, etc., and brain images collected through a variety of imaging modalities. To make the AI systems developed for analyzing neuroimaging data useful for any researcher and to any user, the acquisition procedure should follow common standardization rules. In order to facilitate the subsequent processing phases, this process of collecting data must be carried out following appropriate precautions, including:

1. Use a unique and well-defined protocol so that the transversal uniformity of the data within the case history is guaranteed;
2. Conduct the acquisitions with the foresight to verify that there are no artifacts or parts of the volume of interest excluded, which even if not considered non-critical for the reporting phase by the clinician, can also constitute a vital limit during the subsequent computerized processing; and

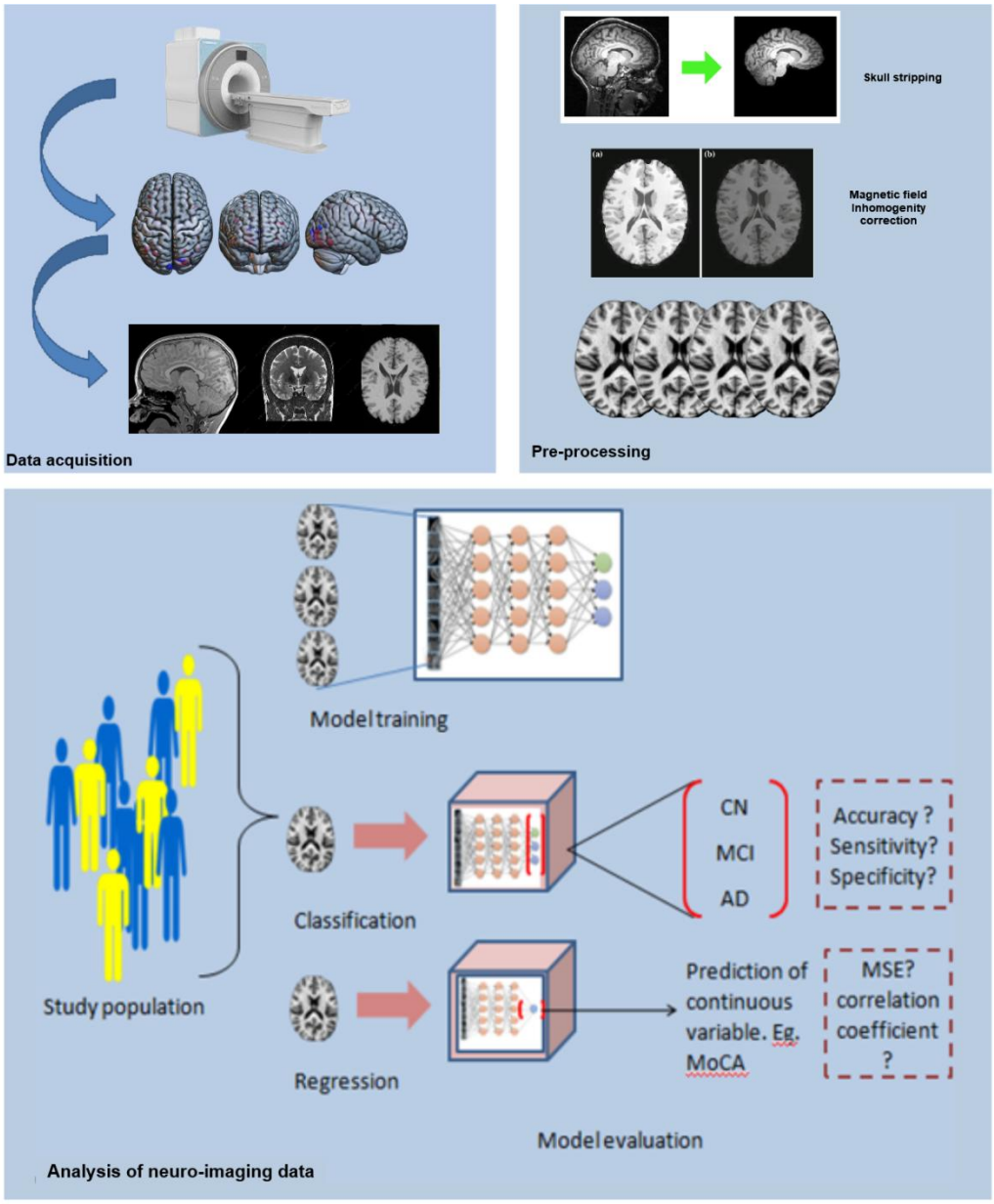


Figure 1.1: A workflow showing the use of deep learning for analyzing neuroimaging data. A) shows the data acquisition task using different modalities such as MRI. B) a pre-processing step used to enhance the brain images and improve their quality. C) illustrates the model training and evaluation phase for classifying a sample brain MRI in to different cognitive groups: CN, cognitively normal; MCI, mild cognitive impairment and AD, Alzheimer's disease.

3. In cases where it is desirable and as permitted by the specific technique used, try to obtain isotropic images, in which the volumetric element of the image, voxel, has the same size in the three orthogonal directions of space.

1.3.2 Data pre-processing

Pre-processing refers to the procedure of applying multiple operations, such as magnetic field inhomogeneity correction, on-brain tissue removal, and registration onto a standard space, that help to remove unwanted artifacts and transform the data into a standard format. This step is aimed at:

1. increasing the usability of the data by maintaining its original organization, for example, by emphasizing the contrast or filtering the noise;
2. extracting information that is not directly available in the starting image, such as images of the diffusion tensor model in the case of diffusion magnetic resonance (MR) imaging; and
3. Segmenting and/or mapping the volume based on atlases and standardized subdivision methods to perform specific measurements and/or obtain a particular topological organization of the data. Thus obtaining the descriptive attributes intended for the subsequent phases.

1.3.3 Analysis of neuroimaging data

Medical image analysis is the process of solving medical problems by extracting information from medical images collected based on different imaging modalities and applying digital image analysis techniques. Predicting disease onset, diagnosing and categorizing disease progression (stage), and following up treatment response are among medical problems that need a solution. These problems can be modeled by medical image analysis tasks such as classification, detection/localization, registration, segmentation, and prediction (regression). Classification and regression are the most frequently applied tasks on neuroimaging data using advanced analysis methods.

Classification

Classification refers to categorizing each subject of the study population into one of the classes or subgroups to which they belong. For example, an MRI scan of subjects in a study cohort can be classified as a cognitively normal individual, an AD brain or a brain with mild symptoms of cognitive decline, MCI, groups, or classes.

The classification algorithm is developed through a preliminary phase called training and a verification phase of model evaluation or validation. For this purpose, the whole dataset is split into a training set and a testing/validation set. During the training step, the model is fed with the training dataset along with the predefined parameters of the model. Hence, it learns how to map the input samples (x) to the output labels (y) in the training set. In the testing phase, test samples will be presented to the model to verify or validate that the model is performing well in classifying unseen test samples. It is important to note that, during the splitting of the dataset into training and test sets, the two sub-sampled datasets should be independent of each other. The performance of the model is quantified by the classification correctness and is measured in terms of different evaluation metrics. For example, accuracy, sensitivity, and specificity are the most common statistical metrics used for a binary classification task.

Regression

Regression analysis is an operation that aims to estimate the relationship between one or more dependent variables and a dependent variable, which represents a situation or characteristics of the subjects under study. To build a regressive system, a model is provided with training data, including the values of the input variables and a continuous output variable, which is the independent variable, along with the learning algorithm and the parameters of the model. During the validation of the model, input variables of test samples will be presented to the predictive system, and the model will produce the predicted value of the continuous variable. Here the performance of the model is measured in terms of the closeness of the predicted value of the variable to its actual value. Mean squared error (MSE) and Pearson's correlation coefficient (r) are the commonly used performance
metrics for regression analysis.

1.3.4 Interpretation of results

The results obtained by analyzing neuroimaging data using any machine learning method should be discussed well and be given a clinical interpretation(Stevens et al., 2020). First, the model's performance should be evaluated using statistical measures with respect to the pre-defined evaluation metrics and be clearly presented. A further assesment that involves identifying features that are given a higher importance by the model for the prediction analysis is also important to infer the association between the features and the variable predicted by the model

1.4 Application of deep learning techniques in neuroimaging

1.4.1 Overview of convolutional neural networks

Convolutional neural networks (CNNs) are particular types of deep learning models suited to image processing computations and could be applied for both classification and regression tasks. They have a layered architecture, where each layer consists of feature extracting elements called filters arranged in small 2D arrays. During the model training procedure, these filters are applied to the image and convolved with the pixels of the image, producing feature maps of that layer. The convolution operation between two 2D arrays is represented mathematically by:

$$G(m,n) = (f * I)[m,n] = \sum_j \sum_k h[j,k]\, I[m-j, n-k]$$

(1.1)

Where, f represents a 2D filter array and I an image.

An image dataset is a collection of either gray scale or color images. A gray scale image is represented as a 2D array of numbers arranged in a dimension of *height x width* (*h x w*). Each pixel is represented by a number between (0, 255), where the value represents the gray scale intensity of that pixel (Figure 1.2). While, for a color RGB image, there are three different channels: R-red, G-green, and B-blue channels. In each channel, the pixel values represent the pixel's intensity for that specific color.

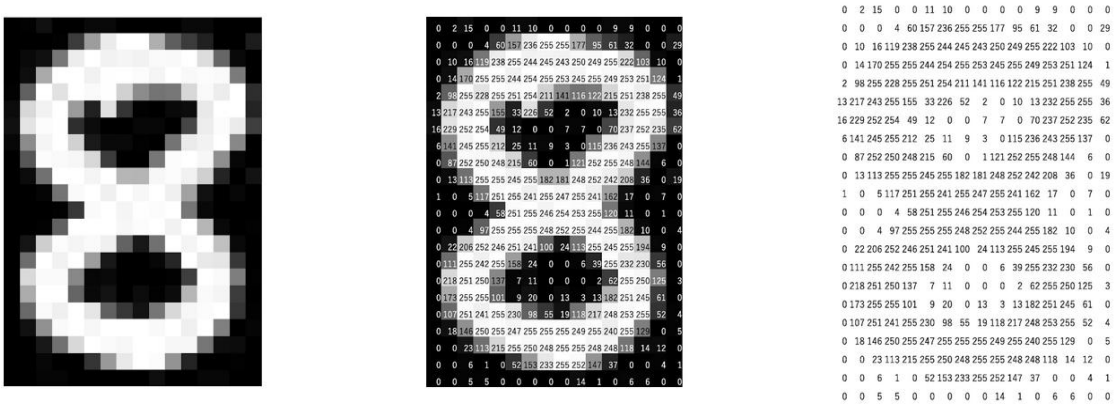


Figure 1.2: An example of a gray scale image representing the number 8. It is composed of pixels arranged as a 2D array of dimension height x weight. Each pixel's value represents the gray scale intensity value in the range of 0 to 255, 0-representing black pixels(Ünal, 2019).

In each layer of a CNN, more than one filter is included to extract a variety of features. Each filter tries to capture different features, shapes, and edges. Considering one layer (e.g., the first layer) of a CNN, if an input image "I" of dimensions *n x n* is presented to a convolutional layer consisting of *d* filters of size *f x f*, the computations performed to get an output of this layer are:

1. First, the image *I* will be convolved with each of *d* filters, where the convolution involves an element-wise dot product between the pixels of the image and the elements of the filter, which are considered the "weights" of that layer;
2. After the convolution, a "bias" term is added to each element; and
3. Finally, the summed output passes through a non-linear activation function. These operations are illustrated in Figure 1.3.

According to these computations, a CNN model learns how to relate the input to the output by minimizing a pre-defined error function through an iterative optimization procedure. In this way, the trained CNN acquires knowledge about the specified classification or regression task and will be used to give predictions on unseen input data.

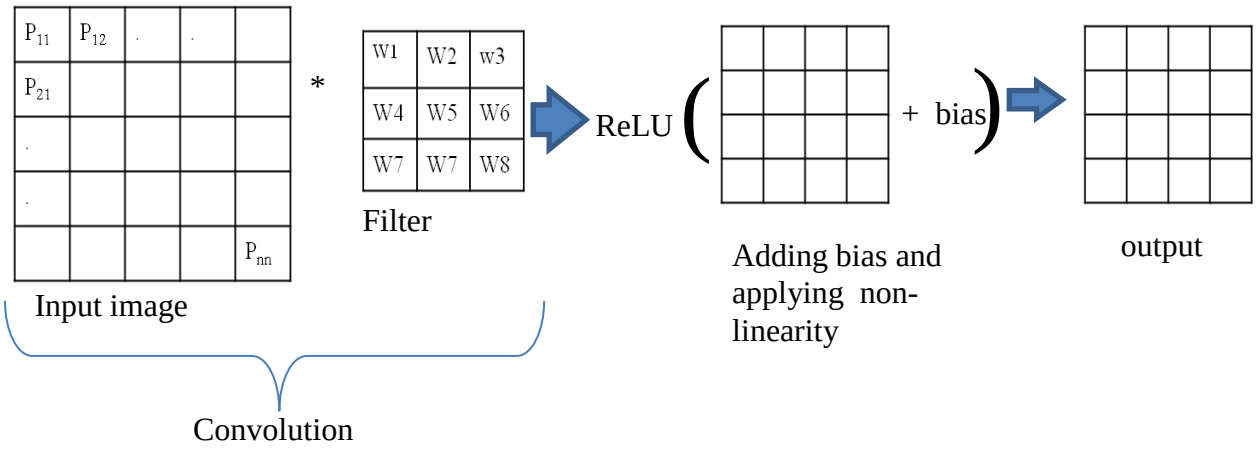


Figure 1.3: Computations applied to an input image. The image is first convolved with the filters, then a bias term is added to the result, and then it passes through a non-linear function.

1.4.2 Application examples of CNNs in neuroimaging

In recent years, deep learning has become a popular class of machine learning algorithms in computer vision and has been successfully employed in various tasks, including multimedia analysis (image, video, and audio analysis), natural language processing, and robotics(Hatcher and Yu, 2018). In particular, deep convolutional neural networks (CNNs) hierarchically learn high-level and complex features from input data, hence eliminating the need for handcrafting features, as in the case of conventional machine learning schemes(Goodfellow et al., 2016a). It has also become a natural trend to apply these methods for different neuroimaging data analysis tasks in order to better understand brain alterations caused by normal aging and to solve the clinical questions related to different neurological disorders (see Greenspan et al. and Zaharchuk et al. for reviews). Several studies employed deep learning methods for image improvement and transformation(Bahrami et al., 2016, Han, 2017, Li et al., 2014, Liu et al., 2018a, Vemulapalli et al., 2017, Zhu et al., 2018). Other studies performed lesion detection and segmentation (Chang, 2016, Dou et al., 2016, Maier et al., 2015) and image-based diagnosis using different CNNs architectures(Liu et al., 2015, Plis et al., 2014). Deep learning has also been applied to more complex tasks, including identifying patterns of disease subtypes, determining risk factors, and predicting disease progression (see, e.g., Zaharchuk et al. and Davatzikos for reviews).

Considering disease diagnosis, several studies employed convolutional neural network models to diagnose different neurological and psychiatric diseases such as Alzheimer's disease (AD), mild cognitive impairment (MCI), Parkinson's disease, autism spectrum disorder, and schizophrenia.

Most of the diagnostic investigations of neurological diseases have focused on AD and MCI. While the earliest studies have used stacked auto encoder (SAE)(Suk et al., 2015, Liu et al., 2015), auto encoder (AE) and deep belief network (DBN) (Suk et al., 2014)models to analyze AD from neuroimaging data, the latest studies applied deeper CNNs to perform AD diagnosis. Moreover, studies focusing on both 2D-CNN (Gupta et al., 2013, Liu and Shen, 2014, Sarraf et al., 2016, Billones et al., 2016, Liu et al., 2018c) and 3D-CNN (Payan and Montana, 2015, Hosseini-Asl et al., 2016b, Karasawa et al., 2018, Liu et al., 2018c) model types have achieved great results. Also, considering the different types of MCI, including the prodromal stage of AD and vascular MCI, interesting results have been achieved.

According to our literature survey, since 2015, there has been a blow in the number of neuroimaging publications using deep learning approaches. We also observed that surprisingly very good results had been achieved in most of the studies.

As we attempted to reproduce the results reported in some of the papers, we noticed that most studies neither shared their source code nor included enough information about the model architecture, hyperparameters used, and validation and evaluation methods followed to achieve such very good results. This motivates us to raise questions if those exciting results were associated with some methodological biases. The fact that studies(Saravanan et al., 2018, Hutson, 2018, Ching et al., 2018, Zhu et al., 2019), highlighted the different challenges and obstacles related to deep learning methods as employed in health care applications strengthen our claim of the need for carefully designing deep learning systems in healthcare and especially in neuroimaging to avoid the possible biases that overestimate the resulting model's performance.

Overfitting due to small dataset size(Zhu et al., 2019, Ching et al., 2018), data leakage(Thibeau-Sutre et al., 2021, Bussola et al., 2021, Saravanan et al., 2018, Wen et al., 2020), reproducibility problems(Hutson, 2018, Ching et al., 2018, Thomas et al., 2021) and lack of interpretability (Zhu et al., 2019, Thomas et al., 2021, O'Sullivan et al., 2020, Ching et al., 2018) are the major

challenges and pitfalls seen in the neuroimaging literature applying deep learning analysis methods.

However, there are studies that tried to tackle such pitfalls and challenges by following appropriate data pre-processing procedures and incorporating the latest AI techniques. An example classification study by Valliani and Soni (Valliani and Soni, 2017) applied a CNN model on structural MRI data to categorize subjects as healthy controls and AD or MCI. The authors performed both binary (AD vs. HC) and 3-way (AD vs. MCI vs. HC) classifications achieving accuracies 81.3% and 56.8%, respectively. Like most medical image datasets, their dataset size was relatively small to train a deep CNN from scratch. To reduce the problem of overfitting due to small training samples, they employed an AI technique called transfer learning which allows transferring knowledge of the CNN model to identify image features from a source task into a target task. Hence, rather than starting from randomly initialized model weights, the training procedure fine-tunes the MRI dataset. In addition, they also used a method called data augmentation, which involves the generation of more samples by applying simple affine transformations such as rotations, flips, and rotations to increase the size of the dataset. Another important point is that Valliani and Soni included only one slice from each MRI volume to avoid the introduction of a data leakage caused by including different slices of a single MRI volume to be included in both the training and test sets. Another paper by Qiu (Qiu et al., 2018) demonstrated a classification procedure of subjects as normal cognition (NC) and MCI from multimodal data that consists of MRI images and cognitive scores: Mini-Mental State Examination (MMSE) and logical memory (LM) test scores. They developed three models, two multilayer perceptron (MLP) taking clinical scores and a CNN model adapted from a pre-trained VGG-11 model; to reuse the pre-trained weights of VGG-11 since the dataset size was small. They trained three different VGG-11 models to prevent data leakage to accept three individual slices selected from the MRI volume, the output being the class probability obtained using a majority rule.

In both studies, the authors prevented data leakage and tried to avoid the risk of model overfitting due to small size of the dataset by using transfer learning (Qiu, et al., 2018; Valliani & Soni, 2017) and augmentation techniques. This surely prevents a bias in the model development procedure that could lead to overestimated results. However, both studies did not

openly share their studies hence, other researchers barly get a chance to reproduce the results reported by the authors. In addition, explainability tools that allow clinicians or health care experts to get a better interpretation of the results are missing. The absence of these two features, reproducibility and interpretability, reduce the trustworthiness of the findings obtained by many studies:

- For the neuroimaging community, to be benefited from the artificial intellegience (AI) research findings at a clinical level, the proposed AI systems are expected to be developed in a procedure which is free from any methodological bias;
- incorporate interpretability tools that will increase their trustworthiness;
- be openly available to others to allow knowledge sharing and to reproduce the results.

To bring the trend of developing reliable AI systems for neuroimaging applications, one solution could be to design open source frameworks that incorporate these basic features throughout the pipeline (starting from data pre-processing to results interpretation). Apart from providing a bias free data pre-processing, model training and evaluation environment, such open source software can also be used as a benchmark to compare the performances of different AI systems. Considering AI tools that are designed specifically for analyzing brain MRI data, there are a few open source deep learning tools that are intended to perform different analysis tasks(Gibson et al., 2018, Pawlowski et al., 2017, Kaczmarzyk et al.). However, these python based deep learning tools do not incorporate AI explainability feature that reduces their reliability. Consequently, to address these shortcomings, this study proposes an open source python software (https://github.com/Imaging-AI-for-Health-virtual-lab/Slice-Level-Data-Leakage) that incorporates versatile features for analyzing volumetric brain image data, specifically, MRI data. The software has features of versatility in terms of model architecture and choice of validation schemes, reliability achieved by providing a lekage-free data pre-processing and interpretability by integrating a number of model visualization techniques (refer Chapter 4 for a detail explanation).

1.5 Motivations and objectives of the study

Based on the considerations set out so far, having recognized the potential of applying deep learning tools to neuroimaging data, considering the methodological biases for the study of brain

structure and function and the alterations caused by neurological diseases, the objectives of the studies conducted in this work were:

1. The assessment of methodological pitfalls of the literature on deep learning applied to neuroimaging;
2. The design and development of interpretable, reproducible, and leakage-free deep learning software for classification/prediction analysis; which is characterized by:
    1. High versatility in terms of the choice of the CNN model architecture, which differ in terms of the number of inputs used, also on the number of MRI modalities employed;
    2. Inclusion of different validation techniques;
    3. Integration of techniques that help overcome overfitting when the available dataset is of small size; and
    4. Multi-tasking: the software can be used for either regression or classification tasks.
3. The study of neurological diseases not yet been investigated through deep learning approaches. From an applicative point of view, the focus was particularly on the study of vascular mild cognitive impairment (VMCI) in patients with small vessel disease (SVD), which is recognized as one of the main causes of cognitive impairment(Zhou and Jia, 2009). However, to date, it has only been studied using conventional machine learning approaches, based on manually extracting features associated with the disease pathology, usually drawing or delineating region of interest areas, that need expert knowledge, and not through advanced AI methodologies based on automatic feature extraction and analysis mechanisms.

In a broader sense, the motivation for this approach is the transfer of knowledge relating to the disease to the application level through tools potentially capable of:

1. Providing researchers with an environment that can preserve the data pre-processing and model training procedures from being contaminated by methodological flaws and hence producing reliable results supported by explainability tools; and
2. Delivering knowledge about individual patient's cognitive status, from raw neuroimaging data, without expert knowledge on the pathology of neurological disease.

Chapter 2

2. Overview of deep learning methods

2.1 Artificial neural networks and deep learning

2.1.1 Artificial neural networks

Artificial neural networks (ANN) are a subset of machine learning models concerned with algorithms inspired by how a human brain works, mimicking how biological neurons communicate. The basic unit of such networks is an artificial neuron, also known as a node, which has an analogy with the fundamental processing element of the human brain called the biological neuron. By simulating the basic functions of natural neurons, an artificial neuron performs four essential functions based on its simple structure shown in Figure 2.1.

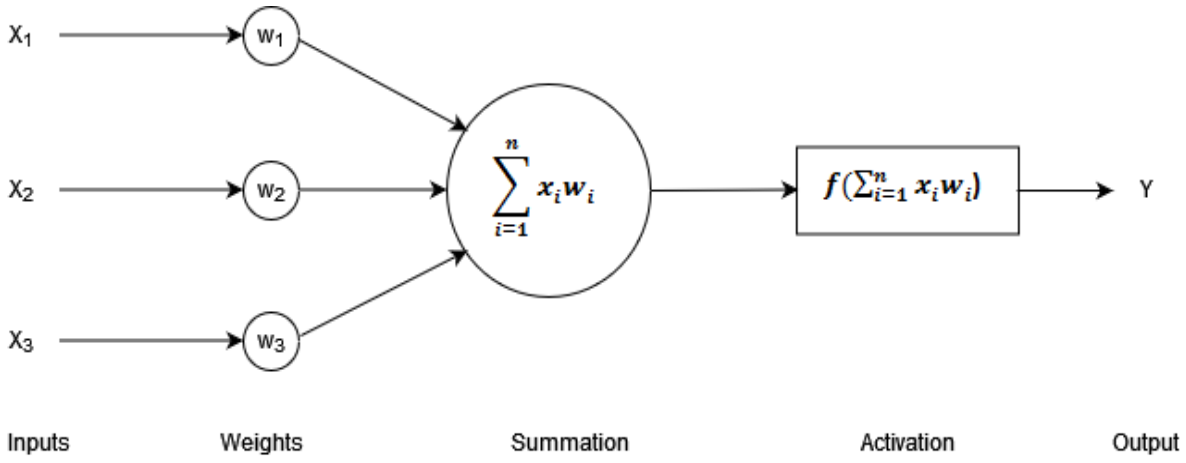


Figure 2.1: Basic structure of the artificial neuron. Each input X is associated with a weight W. The sum of all weighted inputs is passed onto a nonlinear activation function f that leads to an output Y.

First, the node receives raw inputs or weighted signals from other nodes through its incoming connections. Then it applies a summation operation, and later it passes the weighted sum of the

inputs through a non-linear activation function, the outcome being the activation of the node. Finally, for each outgoing connection, this activation value is multiplied by the specific weights and transferred to the next neuron(Dongare et al., 2012, Kalogirou, 2000).

Several such computational units or nodes are interconnected to create computational models called artificial neural networks. ANNs are structured in layers, where each layer consists of many nodes. The number of layers and nodes in a given ANN and how they are connected, representing the network topology, determine its architecture. Different architectures allow for the generation of functions of different complexity and power. Feed forward neural networks are the simplest and most commonly used class of ANNs(Rosenblatt, 1958, Ripley, 1993). Here, the signal flow is unidirectional, and each node sends information to the node in the next layer from which it does not receive any information. The connection is always in a forward direction, and there are no feedback loops(Micheli-Tzanakou, 2011). A simple feed-forward ANN consists of an input layer, where data is fed to the network, and one or more hidden layers transform the data as it flows through(Lundervold and Lundervold, 2019). Figure 2.2 shows an example of a feed-forward neural network made up of three layers.

An input layer has a number of nodes equal to the number of input variables. It is not an active layer, as it simply passes the raw inputs without applying any modifications to the next layer through its outgoing connections. Each node of this input layer is interconnected to all nodes of the next layer, creating a densely connected structure. When a node in the first hidden layer receives the inputs, it first assigns weights, and then it sums up the resulting products together, yielding a single number. If this number exceeds a pre-defined threshold value specified by the activation function (e.g., Sigmoid, ReLU), the node is activated and passes the sum to the nodes of the next layer. Instead, the node passes no data to the next layer if the number is lower. This way, all nodes in the first layer pass the weighted sum of the inputs to the next hidden layer or the output layer. The final layer, which is the output layer, consists of one or more data points based on the function of the network. For instance, an ANN model that classifies subjects between healthy controls and Alzheimer's disease patients will have a single output unit. While a network that categorizes subjects based on the different stages of AD will have five nodes, each node representing the cognitive states of cognitively normal, early mild cognitive impairment, mild cognitive impairment, late mild cognitive impairment, and Alzheimer’s disease. This

interconnection of many simple non-linear units, structured in layers, allows modeling a very complex function that represents the mapping from the input to the output, given enough training data.

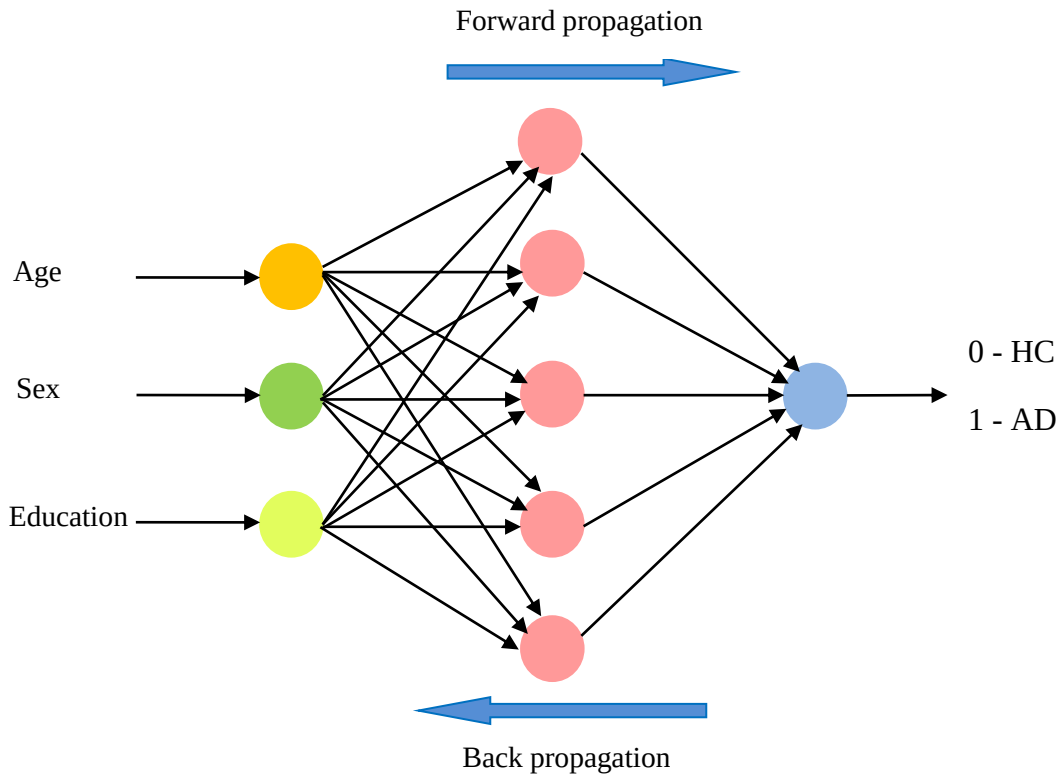


Figure 2.2: A simple feed-forward neural network architecture consisting of an input layer with three nodes to accept three input variables, one hidden layer having five nodes and an output layer with one neuron, is used for classifying the demographic input data representing a subject sample to be classified as a healthy control an AD patient(2019).

Training an ANN can be considered the process of updating network architecture and connection weights to effectively perform a specific task(Jain et al., 1996). Since, in the beginning, the weights and thresholds of ANNs are randomly assigned or initialized, the training procedure aims at iteratively adjusting the weights and thresholds of the network by feeding the model a training data, given as X and Y, until training data with the same labels consistently yield similar outputs. Usually, adjusting network parameters follows a learning rule that governs the updating

process, referred to as a learning algorithm. Back-propagation is a learning algorithm suitable for feed-forward ANNs(Krogh, 2008, Rumelhart et al., 1986).

2.1.2 Back-propagation

To train a feed-forward ANN employing a back-propagation learning algorithm, labeled training data, given as X and Y is required. The training starts by setting all the weights in the network to small random values. By passing a sample input x_i through the network, an output y_i is produced. A pre-defined cost function measures the difference between the desired output and the output by the network, which is the prediction error of a sample training data. Summing up the errors over all training samples yields the total error of the network. In the beginning, the error is expected to be large. By repeating this procedure over, usually hundreds of times, the error gets smaller and smaller and reaches a point where the error no longer changes(Krogh, 2008, Rumelhart et al., 1986).

2.1.3 Deep Learning

By increasing the depth of artificial neural networks(Yegnanarayana, 2009), the concept of deep learning has been proposed(Bengio et al., 2007, Bengio, 2009). A neural network that consists of more than three layers—inclusive of the inputs and the output—can be considered a deep learning algorithm. Figure 2.3 illustrates the general representation of deep learning models. Deep learning models, a family of representation learning methods, have a layered architecture, where each layer is composed of several non-linear neurons that can transform the representation of the input data at one level into a representation at a higher, slightly more abstract level. Hence, they can automatically discover the representation of the input data by capturing intricate structures in a high-dimensional data, which could be used for classification or prediction analysis(LeCun et al., 2015). Due to the increased depth (number of hidden layers) in deep learning models, compared to artificial neural networks, a more abstract high-level feature representation for the input data is formed by the multiple hidden layers to combine low-level features(Liu et al., 2018b).

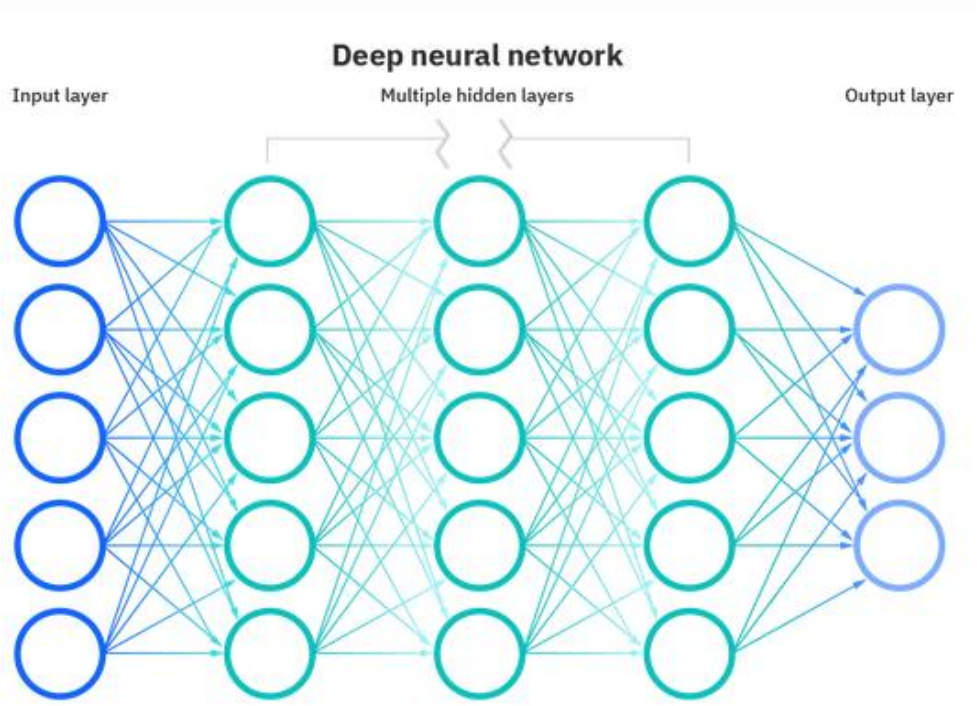


Figure 2.3:A general representation of deep learning models(2019).

2.2 Convolutional neural networks

CNN is the most established among the various deep learning algorithms especially for image processing tasks(Yamashita et al., 2018). It is a type of deep learning model which is suitable for processing data that come in the form of multiple arrays(LeCun et al., 2015), such as images, which is inspired by the organization of the animal visual cortex (Hubel and Wiesel, 1968). Due to their architecture, which is a structure of multiple consecutive stages, they can automatically and adaptively learn spatial hierarchies of features, starting from low-level features extracted by the initial layers to high-level patterns, captured by the last layers(Yamashita et al., 2018).

A typical CNN architecture consists of convolution layers and pooling layers placed alternatively at the beginning of the network and the final fully connected (FC) layers. Feature extraction is performed by convolution and pooling layers. Instead, the fully connected layer maps the extracted features to a final output, such as classification or prediction(Yamashita et al., 2018).

2.2.1 Convolution layers

The convolution layer is the main building block of CNNs. It consists of a linear convolution operation followed by a non-linear activation function. For a given image input, these layers perform a linear mathematical operation called convolution between the input image pixels and a small array of parameters, called the kernel. A kernel is an optimizable filter structured as a matrix of numbers, specified as width x height (for two-dimensional (2D) CNN), and can extract features from the input data. Hence, a kernel is moved over the image to overlap with each pixel, and convolution, the dot product between the kernel and the overlapped area of the image, is computed to get an output, called a feature map. Usually, multiple kernels, expressed as a hyperparameter *Number of filters*, are applied to get different feature maps (e.g., Horizontal edges, vertical edges) that represent different image patterns(Yamashita et al., 2018). This parameter provides depth to the convolution layers. Hence, the output of a convolution layer is represented as *Wo x Ho x Do.*

By stacking a number of these convolution layers, only the filters in the first layer are convolved with the input image outputting many feature maps equal to the number of filters in the first layer. Moreover, the next convolutions will be between the feature maps of the previous layer and the kernels. These consecutive operations allow extracting hierarchically and progressively more and more complex features(Yamashita et al., 2018).

The other important parameters when considering a convolution layer are padding and stride.

Padding: The convolution operation described above does not allow the center of each kernel to overlap the outermost element of the input tensor and reduces the height and width of the output feature map compared to the input tensor. Padding, typically zero padding, is a technique to address this issue, where rows and columns of zeros are added on each side of the input tensor to fit the center of a kernel on the outermost element and keep the same in-plane dimension through the convolution operation . Modern CNN architectures usually employ zero padding to retain in-plane dimensions to apply more layers. Each successive feature map would get smaller after the convolution operation without zero padding.

Stride: refers to the distance between successive kernel positions during the convolution operation. A stride value larger than 1, results in a down sampling of the output feature maps.

Usually, instead of using this parameter to reduce the dimension of feature maps an alternative technique called pooling, which is described below, is used after convolution operations.

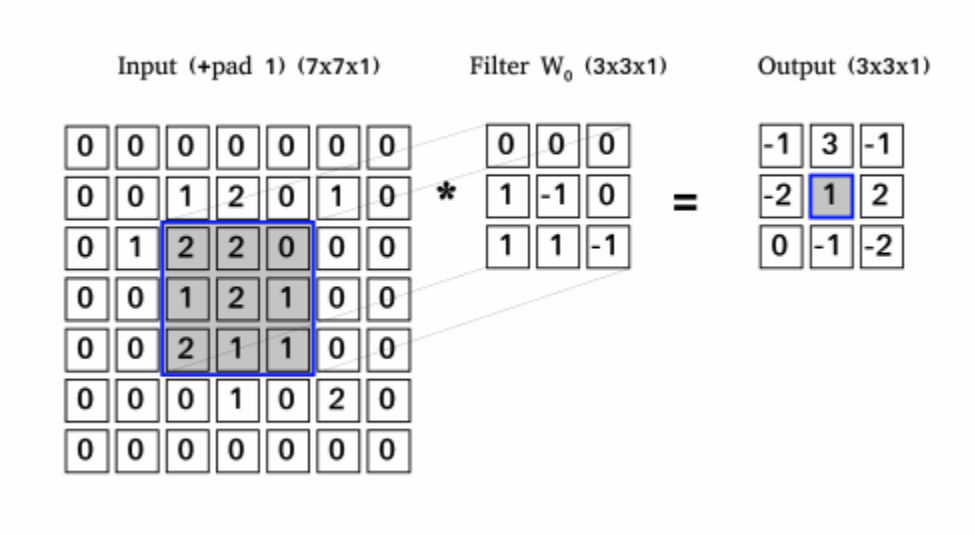


Figure 2.4: Example showing the convolution between an input image and a 3 x 3 kernel. An element-wise product between the filter and the overlapped image pixel values is computed and summed up to get an output(2022).

Activation function

After applying a linear operation, convolution on to the input image, the feature map is passed through a non-linear function, called activation function. By comparing the output with a threshold value, this unit's role is to fire or block the elements of the feature map matrix to pass to the next layer. Figure 2.5 illustrates the plots of activation functions that are commonly used in CNN models

2.2.2 Pooling layer

In most CNN architectures, it is common to insert a pooling layer between consecutive convolution layers. It is one of the layers without learnable parameters. It performs down sampling that reduces the spatial size (width and height) of the input volume, keeping the depth the same. This procedure helps to decrease the number of learnable parameters. Moreover, the pooling operation introduces an important feature of CNNs, which is invariance to small shifts and distortions(Yamashita et al., 2018). Max pooling and Global average pooling (GAP) are the two available functions used in CNN models.

Max Pooling: it receives feature maps from the previous convolution layer, extracts patches of the same size as the filter size of the pooling layer, and outputs the maximum value in each patch, discarding all the other values.

Global average pooling: rather than taking a patch of the feature map, it computes the average of the whole feature map of size *height x width*, down sampling into a 1x1 array. Here, also, the depth of the feature maps is retained. Unlike Max pooling, it is possible to apply this type of pooling only once just before the fully connected layers. The advantages of applying this type of pooling are twofold. First, it substantially reduces the number of learnable parameters. In addition, it enables the CNN to accept inputs of variable size(Lin et al., 2014).

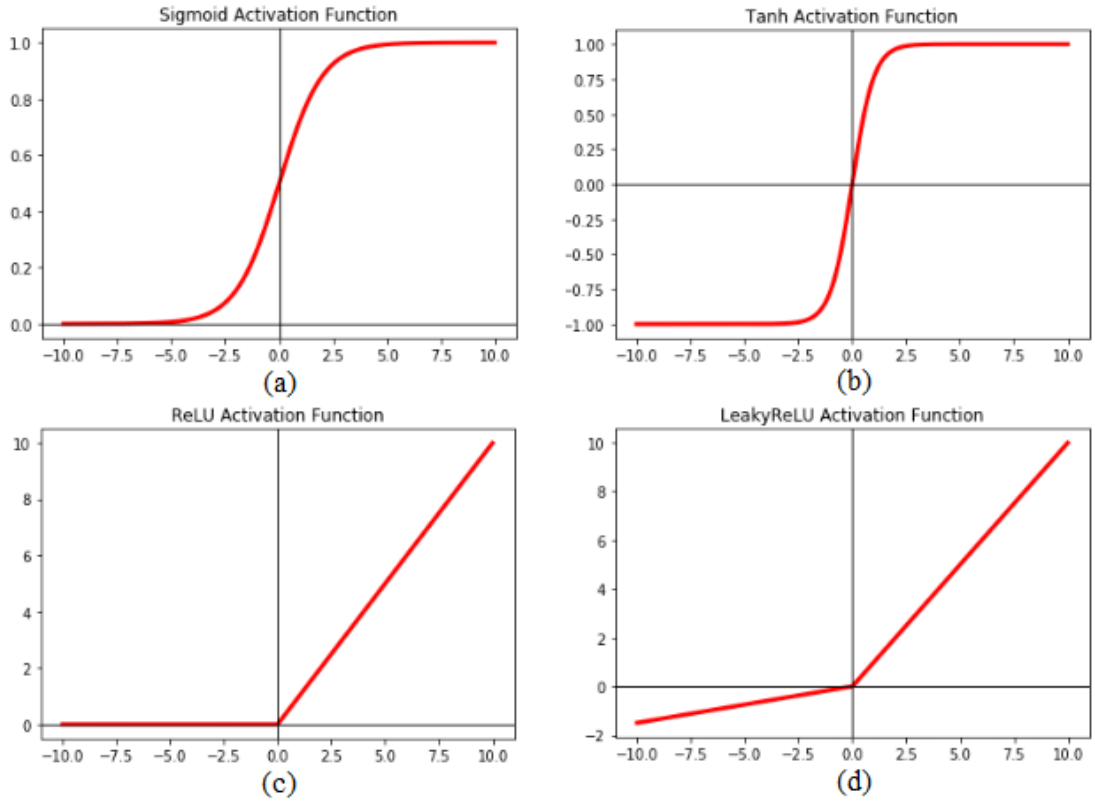


Figure 2.5: Plots of different activation functions. a) sigmoid activation, b) hyperbolic tangent (tanh) activation, c) Rectified linear unit (ReLu) activation and LeakyRelu activation(Yang, 2018).

2.2.3 Fully connected layer

To make the dimension of the feature maps coming from convolution or pooling layers fit with the input size of the fully connected layer, either a Global average pooling is applied just before adding a fully connected layer or a flatten layer is added, to transform the three dimensional (3D) array (height x width x depth) into a one-dimensional (1D) array of numbers. In fully connected layers, since every input is connected to every output by a learnable weight parameter, they have also known by the name dense layers. Similar to convolution layers, FC layers are followed by a non-linear activation function, usually ReLU activation. The features extracted by convolution layers are mapped by the FC layers, and the final FC layer performs classification or regression based on the feature values. Unlike the previous layers, the choice of the activation function for the final FC layer depends on the type of the task(Yamashita et al., 2018). For a prediction (regression) task, a "*linear"* or identity function is used. For classification of binary classes, "*sigmoid"*, and for multi-class classification, "*softmax"* activation is preferred.

2.3 Model training

Training a CNN involves finding the optimal parameters of the network, kernels in convolution layers, and weights in fully connected layers, which minimize the function that computes the differences between output predictions and given ground-truth labels on a training dataset, namely the loss function. Back-propagation algorithm is commonly used for training neural networks where loss function and gradient descent optimization are the key elements. During the forward pass of the input data, the model's performance, which is measured in terms of the loss function, is computed, and the error is propagated during backpropagation to update the values of kernels and weights by a small amount using a gradient descent algorithm(Yamashita et al., 2018). This combination of feeding the training data in the forward pass and backpropagation is iteratively applied a number of times, defined by the hyperparameter *epoch* number, to update the learnable parameters until the trained model provides a good performance as measured by the pre-defined metrics. Two key parameters are used during model training: loss function and an optimization algorithm.

Loss function: is a function that computes the error between the true labels/values of training samples and the prediction outputs. Cross entropy is the most commonly used loss function for

classification problems, where binary cross-entropy is employed for binary classification and categorical cross-entropy for multiple categories. Instead, the mean square error is typically applied to regression problems.

Optimizer: gradient descent is one of the most popular algorithms to perform optimization and is the most common way to optimize neural networks. It is an optimization algorithm that iteratively updates the learnable parameters, kernels, and network weights to minimize the loss. The optimization process computes the gradient of the loss function concerning each of the parameters (kernels and weights), where its value gives information about the direction in which the function has the steepest rate of increase. Hence, each parameter is updated in the negative direction of the gradient with an arbitrary step size determined by the value of a hyperparameter called the *learning rate*.

For a learnable parameter w, and loss function represented as L, the gradient is computed as:

$$gradient = \frac{\partial L}{\partial w} \tag{2.1}$$

After the gradient is calculated, each learnable parameter is updated as follows:

$$w := w - \alpha * \frac{\partial L}{\partial w} \tag{2.2}$$

Where α is the learning rate, one of the most critical hyperparameters that need to be initially set before the training starts. Several optimization algorithms, which are derivatives of the gradient descent algorithm, are provided to researchers by AI experts. The different varieties differ in the frequency of parameter updates and the technical improvements that speed up the training process. Stochastic gradient descent (SGD), SGD with momentum(Qian, 1999), adaptive gradient descent (Adagrad)(Duchi et al., 2011), adaptive moment estimation (Adam)(Kingma and Ba, 2014), and root mean squared propagation (RMSProp) are among the different families of gradient descent algorithm. Refer to Ruder (Ruder, 2016) for detailed reading.

2.4 Model validation and evaluation techniques

Model validation is the process of verifying that trained models are providing satisfactory outcomes to their input data, both quantitatively and qualitatively. The model is evaluated with a

testing dataset, a separate portion of the same dataset from which the training dataset is derived(Wang and Zheng, 2013). In addition, there is also an assumption that all samples in the dataset are independent and identically distributed (IID) which ensures that all samples have been drawn from the same probability distribution and are statistically independent of each other. Keeping this assumption, the whole dataset is split into training and testing sets. After the model is trained on the training set, it is then validated on the remaining test set. Evaluating the model on a separate test set is important to estimate how well a model performs on unseen test data, which is the generalization ability of the trained model. There are three popular model validation strategies. The reasons for evaluating the predictive performance of a model include:

- To estimate the generalization ability, which is the predictive performance of our model on future (unseen) data;
- To increase the predictive performance by tweaking the learning algorithm and selecting the best performing model from a given hypothesis space; and
- To identify a machine learning algorithm that is best suited for the problem at hand: thus, we want to compare different algorithms, selecting the best performing one and the best performing model from the algorithm's hypothesis space.

In all these tasks, even if the objective is to evaluate the model's performance, they require different approaches or different validation strategies.

Other essential points to be considered during model validation are the concepts of bias and variance.

Bias (statistical bias): is the difference between the expected prediction accuracy of our model and the true prediction accuracy. Mathematically, the bias of an estimator $\tilde{\beta}$ is the difference between its expected or mean value $E[\tilde{\beta}]$ and the true value of a parameter β being estimated.

$$\text{Bias}=E[\tilde{\beta}]-\beta \qquad (2.3)$$

So, if $E[\tilde{\beta}]-\beta=0$, then $\tilde{\beta}$ is an unbiased estimator of β. For example, if we compute the prediction accuracy on the training set, this would be an optimistically biased estimate of the absolute accuracy of our model since it would overestimate the true accuracy.

Variance: is a measure of the variability of our model's predictions if we repeat the learning process multiple times with small fluctuations in the training set. The more sensitive the model-building process is towards these fluctuations, the higher the variance.

In the formula, it is simply the statistical variance of the estimator $\tilde{\beta}$ and its expected value $E[\tilde{\beta}]$

$$\text{Variance}=E[(\tilde{\beta}-E[\tilde{\beta}])^2] \quad (2.4)$$

2.4.1 Holdout validation

It is the simplest model validation technique. Assuming that all data has been drawn from the same probability, first, the labeled dataset is split into training and test sets by performing a simple process of random subsampling. Then the model is trained on the training samples and validated on the test set. It is important that the test set is touched once to make sure that there is no bias introduced when the generalization accuracy is estimated. The fraction of correct predictions constitutes our estimate of the prediction accuracy. The reason for keeping aside a separate test set is that training and testing the model on the same dataset introduces a very optimistic bias due to overfitting. This is a situation where we cannot tell if the model is memorizing the training data or whether it generalizes well to new, unseen data.

This approach of dividing the data into training and test set, fitting the model on the training set, and testing on the remaining test samples has the following limitations:

1. Since hyperparameters are not learned during model fitting, this approach cannot perform hyperparameter tuning. Hence fixed hyperparameter values are used for training and validating the model;
2. The total dataset represents a random sample drawn from a probability distribution; and we typically assume that this sample is representative of the true population – more or less. When this representative dataset is divided in to training and test set, due to the kept aside sample as a test set , which is a process of sub-sampling without replacement, the statistic (mean, proportion, and variance) of the sample will be altered, causing a violation of IID and a change of class proportion. The degree to which sub-sampling

without replacement affects the statistic of a sample is inversely proportional to the size of the sample; and

3. If the model has not reached its capacity, the performance estimate would be pessimistically biased. Assuming that the algorithm could learn a better model from more data, we withheld valuable data set aside for estimating the generalization performance (i.e., the test dataset). Hence, our estimate of the generalization performance may be pessimistically biased. Although the pessimistic bias could be reduced by decreasing the proportion of the test set, it causes an increase in the variance of the model's performance and thus widens the confidence interval.

From points 2 and 3, we can note that holdout validation is a good choice and is fine for model evaluation when working with relatively large datasets (Raschka, 2018).

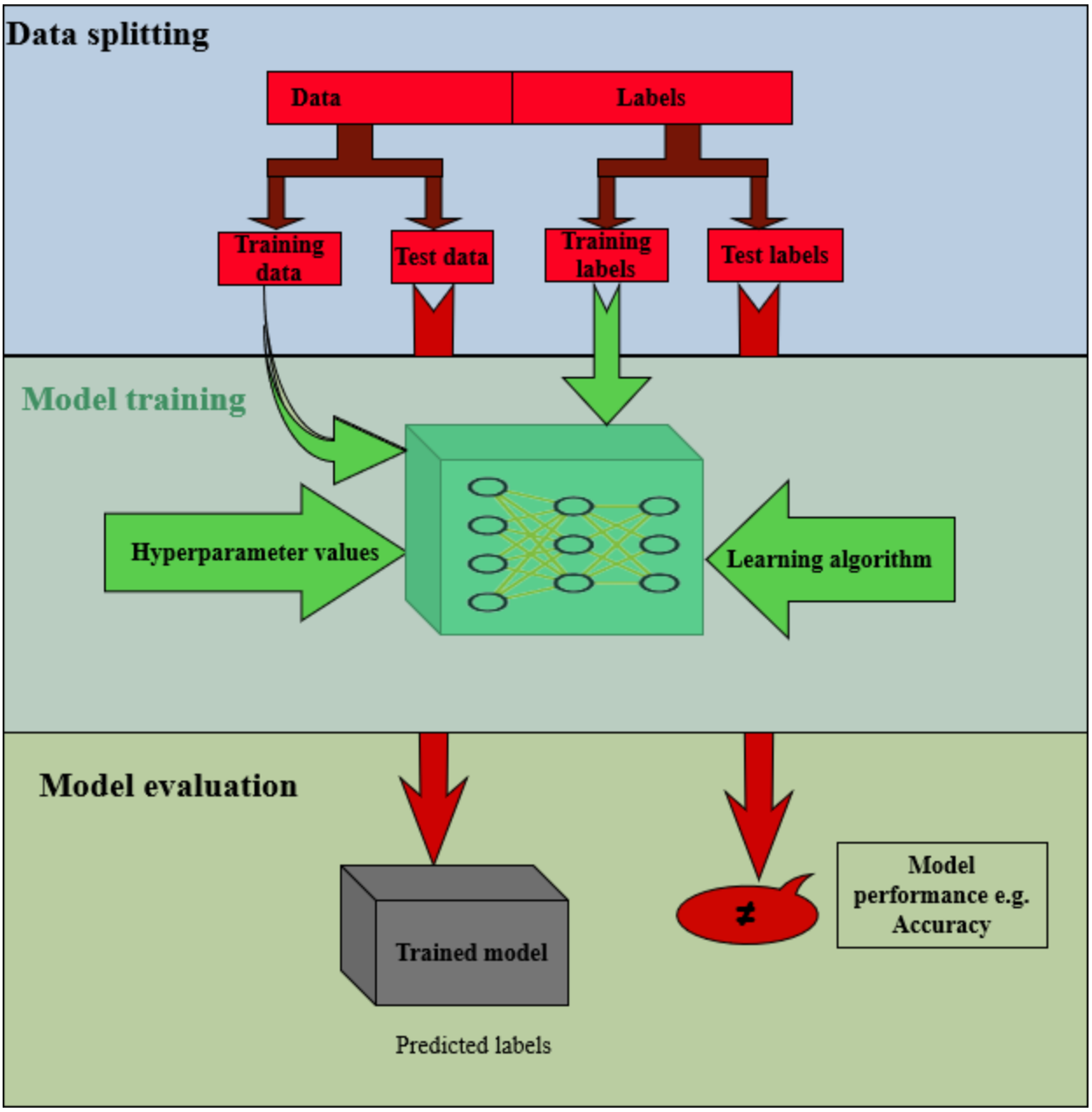


Figure 2.6: A holdout model validation. The dataset is split into training and testing sets, then the model is trained on the training sub-sampled set, ; Lastly, the trained model is evaluated on the test set.

2.4.2 Cross-validation (CV)

The process of finding the best-performing model from a set of models that were produced by different hyperparameter settings is called model selection. K-fold cross-validation is the most

common technique for model validation and model selection(Rodriguez et al., 2009), based on the idea that each sample in the dataset has the opportunity of being a test sample. The process involves splitting the dataset into k parts and, the model is trained k times, each time one part is used as a test set, and the other k-1 parts will be merged and are used to train the model. By doing this, each sample in the dataset will get a chance to be a test sample.

The main advantage of this approach is that it can reduce the pessimistic bias by using more training data in contrast to setting aside a large portion of the data as a test set. If the k-fold CV is used for model evaluation, the model will be trained with fixed hyperparameters. To decide the number of folds k, we need to consider the bias-variance trade with respect to k. The general trend when increasing the number of folds or *k is*:

- the bias of the performance estimator decreases (more accurate);
- the variance of the performance estimators increases (more variability);
- computational cost increases (more iterations, larger training sets during fitting); and
- Exception: decreasing the value of *k* in k-fold cross-validation to small values (e.g., 2 or 3) also increases the variance on small datasets due to random sampling effects.

For model selection also, first, the dataset is divided into k parts. Then, for each hyperparameter value, a model is trained to apply a KFold CV where the performance of each model trained on a specific hyperparameter value becomes the average performance computed over the K folds(Raschka, 2018). The procedure is illustrated in Figure 2.7. The main drawback of this approach is that, since the model selection is performed on the whole dataset, split into k folds, there is no separate test set to estimate the chosen best model's generalization ability. Consequently, another more robust method of model validation, namely nested cross-validation, is recommended in most deep learning applications for small to moderate-sized datasets.

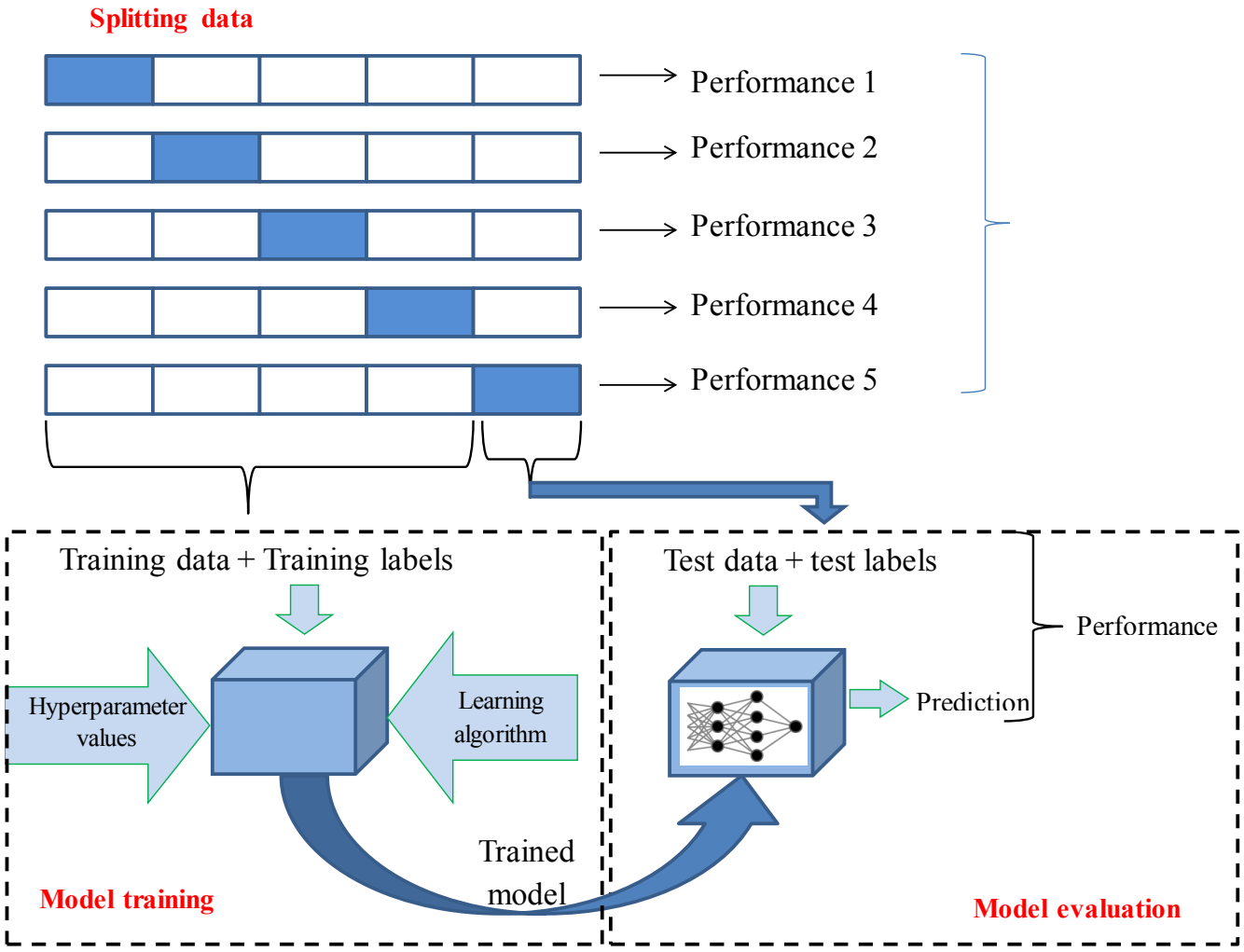


Figure 2.7: An example of a 5-fold cross-validation. The dataset is divided into 5 equal parts called folds. Then, the model is trained 5 times, each time, one fold is kept as a test set and the remaining 4 parts are merged together and used to train the model. The performance is computed as the average of the performance of the 5 folds.

2.4.3 Nested cross-validation

In practical applications, especially in medical imaging, there is a problem of finding a large dataset that is sufficient enough to keep aside a test set that can provide an unbiased estimate of the true generalization error of a model.

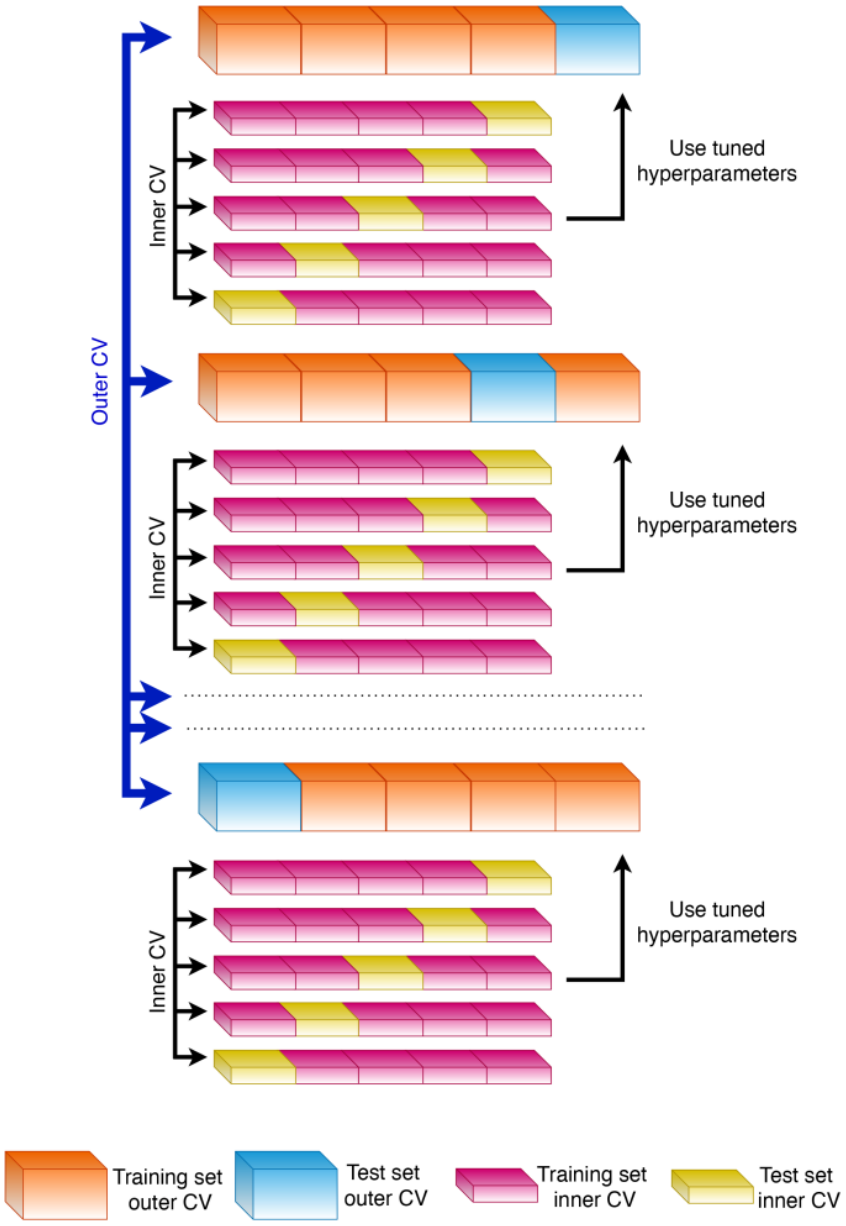


Figure 2.8: A procedure of nested CV. There are two nested folds where the inner fold is used to tune the hyperparameters of the model, and the inner one is used to evaluate the performance of the chosen model.

Reserving too much data for training results in unreliable estimates of the generalization performance, and set aside too much data for testing results in too little data for training, which hurts model performance(Raschka, 2018). Tuning hyperparameters and performing model

selection based on average k-fold performance or the *same* test set, introduces a bias into the procedure, and the trained model's performance estimates will not be unbiased anymore. Mainly, we can think of model selection as another *training* procedure, and hence, we would need a decently-sized, independent test set that we have not seen before to get an unbiased estimate of the models' performance. Often, this is not affordable.

Nested CV, which was first described by Iizuka (Iizuka et al., 2003) and Varma and Simon (Varma and Simon, 2006) when working with small datasets, is a procedure that offers a workaround for small-dataset situations that shows a low bias in practice where reserving data for independent test sets is not feasible. Nested CV reduces the bias, compared to regular k-fold cross-validation when used for both hyperparameter tuning and evaluation. Hence, it provides an almost unbiased estimate of the true error(Varma and Simon, 2006). The method of nested cross-validation is relatively straightforward as it merely is a nesting of two k-fold cross-validation loops: the inner loop is responsible for the model selection, and the outer loop is responsible for estimating the generalization accuracy, as shown in Figure 2.8.

2.5 Performance measurement

The quantififcation of the performance of predictive systems can be carried out through multiple statistical descriptors, each of which is designed to highlight the salient information related to each specific problem or to adapt to specific characterstics of the learning scheme implemented.

2.5.1 Performance metrics for classification

Accuracy: is a quantitative measure of the algorithm's correctness in predicting each class/group concerning the total size of the test set without taking into account any imbalance between the groups.

Sensitivity (recall or true positive rate): is the metric that evaluates a model's ability to predict the true positives of each available category.

Specificity (true negative rate): is the metric that evaluates a model's ability to predict true negatives of each available category.

Receiver operating characteristic curve (ROC): is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:

- True Positive Rate (TPR); and
- False Positive Rate (FPR)

False Positive Rate (FPR): is the ratio between incorrectly classified negative samples and the total number of negative samples.

A ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives. The following figure shows a typical ROC curve.

Area under the ROC curve (AUC): is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes. AUC and ROC curves are shown in Figure 2.9.

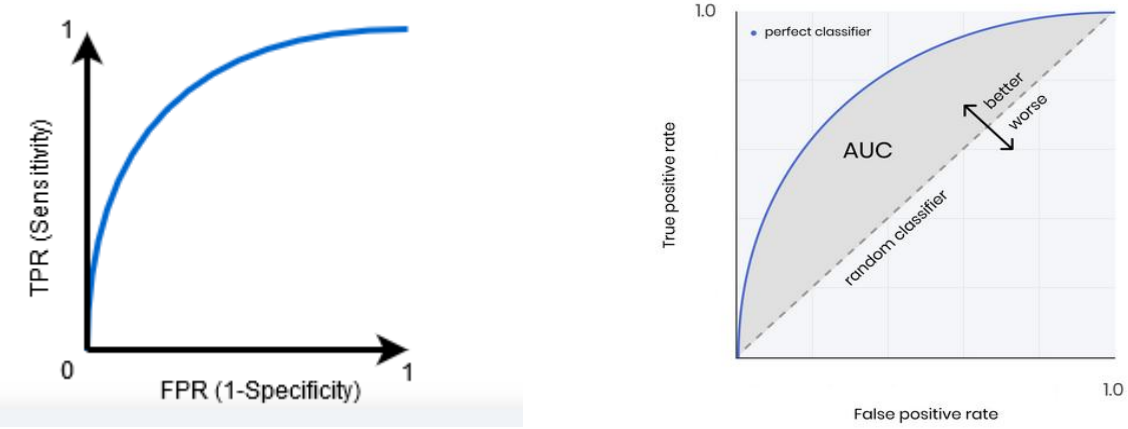


Figure 2.9: An Example of ROC curve and AUC a) a ROC curve, represented as a plot of TPR vs. FPR. b) an AUC score, which is shown as the area under the ROC curve.

2.5.2 Performance metrics for regression

In this study, mean squared error (MSE) and Pearson's correlation coefficient are used in a regression analysis we performed.

Mean squared error (MSE): is defined as $$MSE = \frac{1}{n} * \sum_1^n (y_i - \hat{y}_i)^2 \quad (2.5)$$

Pearson's correlation coefficient (r): $$r = \frac{\sum(x*y)}{\sqrt{(\sum x^2)*(\sum y^2)}} \quad (2.6)$$

Chapter 3

3. Challenges of application of deep learning in neuroimaging

Although the use of deep learning techniques to analyze neuroimaging data is significantly increasing with great promises, reliable application of deep learning for neuroimaging still remains in its infancy and many challenges remain(Zhu et al., 2019). The fact that medical images, also neuroimaging data, are often three-dimensional brings problems associated with memory and computation load. Other important challenges are related to data, interpretability, workflow integration and regularizations(Lundervold and Lundervold, 2019).

3.1 Scarcity of training data and overfitting

Deep neural networks are computationally intensive and complex multi-layered algorithms with parameters on the order of millions(Valliani and Soni, 2017). According to empirical studies suggestions, the convergence of these algorithms requires tenfold more training data relative to the number of parameters hence to produce an effective model. Due to the wide availability of images, videos, and free-form text on the internet, domains such as computer vision and natural language processing have been showing great progress(Valliani and Soni, 2017). Neuroimaging data on the contrary is usually very scarce due to privacy and data protection requirements related to medical data. Also, finding labeled data is very expensive and difficult to produce(Lundervold and Lundervold, 2019). Training a complex classifier with such a small dataset always carries the risk of overfitting(Zhu et al., 2019). Overfitting occurs when a model is able to perform well on data in the training set, but it shows a poor performance on the validation set hence unable to generalize well. This limits the applicability of deep learning systems to be bounded to certain patient demographics and prevents their usage across clinical contexts and the population at-large(Valliani and Soni, 2017).

Although the basic solution is being able to build large, public, labeled medical image datasets, privacy concerns, costs, assessment of ground truth, and the accuracy of the labels remain stumbling blocks(Chartrand et al., 2017).

Several studies tried to use different strategies to reduce overfitting, including regularization(Goodfellow et al., 2016b), early stopping(Prechelt, 1998), and drop out(Srivastava et al., 2014).

Early stopping

The training procedure of deep neural networks, involves iteratively updating the parameters of the model, until the pre-defined loss function reaches a minimum value. Both the training and the validation losses decrease in every iteration as far as the model is learning features that allow performing with a good generalization. Early stopping is a technique that stops model training when the validation loss starts to increase; hence the model starts to overfit the training data (Figure 3.1).

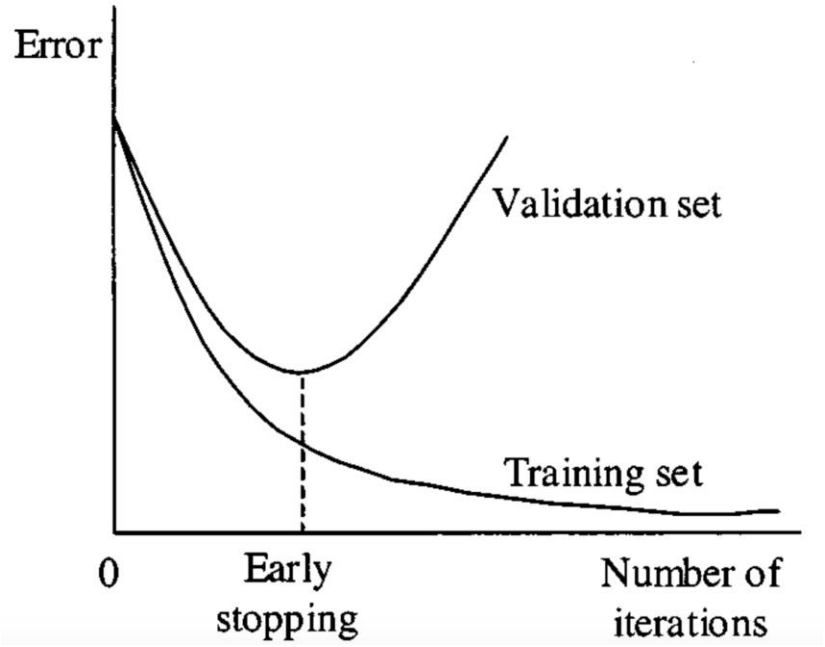


Figure 3.1:The training process is stopped when the validation loss starts to increase(Mustafeez, 2022).

Dropout

The depth of deep neural networks is a key factor in the ability of these models to extract hierarchical and latent features from complex high-dimensional data. However, large networks

consist of a large number of parameters, bringing the problem of overfitting. In addition, these networks are slow to use, making it difficult to deal with overfitting by combining the predictions of many different large neural networks at test time. Dropout is a regularization method that tries to address this problem by randomly dropping neurons out from the neural network along with all its incoming and outgoing connections during training (Figure 3.2). This prevents the model from adapting and overfitting too much of the training dataset(Srivastava et al., 2014). During the test time however, a model without dropout is used.

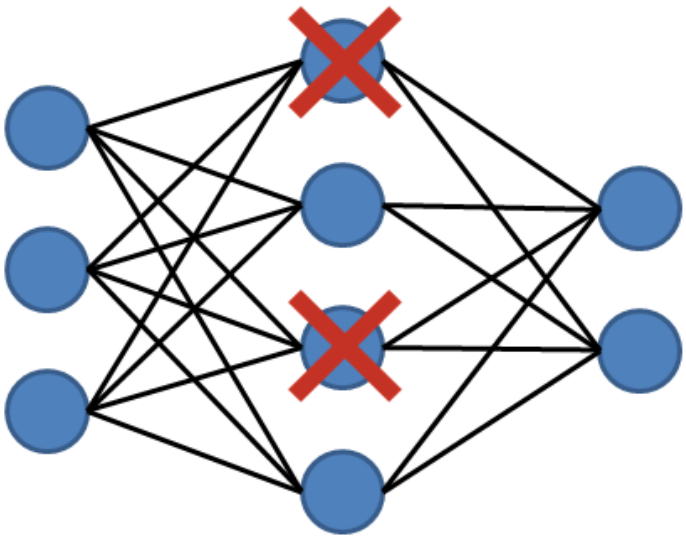


Figure 3.2: Illustration of the dropout technique to reduce the risk of overfitting. During training time, randomly selected neurons are turned off along with their connections(Dabbura, 2018).

Data augmentation

Augmentation is an alternative method to training with more data. It involves the generation of synthetic data using different techniques. Using data augmentation, a lot of similar images will be generated and the model is trained on multiple instances of the same class of objects. This increases the dataset size and as we add more and more data, the model is unable to overfit all the samples and is forced to generalize.

In the case of neuroimaging also, images in the training set are used, and modifications are applied to these samples to generate further representative samples which simulate changes in acquisition and anatomical variation of patients. Different techniques are used to generate new samples from training data.

The most common and simpler approaches involve the application of various operations to the original image, such as, affine transformations (rotation, zooming, cropping, flipping or translations), flipping, translations, scaling, cropping and shearing(Chlap et al., 2021, Nalepa et al., 2019). One limitation of these methods is that the generated images become much correlated to each other offering very few improvements for preventing overfitting and further generalization over unseen samples(Shin et al., 2018). Another drawback is that some operations like rotation and shearing might generate anatomically incorrect images(Nalepa et al., 2019).

Hence, rather than applying modifications to the original images another approach of data augmentation involving the generation of artificial images is very popular for generating medical images. Generative adversarial networks (GANs), which are a family of deep neural networks, are being exploited to augment neuroimaging datasets(Nalepa et al., 2019, Shorten and Khoshgoftaar, 2019, Han et al., 2019).

Generative adversarial networks (GANs)

Generative adversarial networks are types of deep neural networks that are used for generating artificial images. The GAN model architecture consist of two deep learning models, namely a generative model that captures data distribution and a discriminative model that tries to categorize the incoming input as a real or fake example(Goodfellow et al., 2016a). The learning procedure involves an adversarial process where the generator and the discriminator compete for one against the other. A basic GAN model is composed of a training dataset, random noise vector, generator and discriminator and is known by its iterative adversarial training procedure. Figure 3.3, illustrates the basic structure of a GAN network and the training process.

Training dataset (data): this is a dataset of real images that we want the generator to learn. For a sample x in the training dataset, the fixed distribution can be represented as $P_{data}(x)$.

Random noise vector (z): this is a raw input to the generator. The generator uses a simple random noise variable as a starting vector to generate synthetic images.

Generator: a generative model is a a neural network with parameters θ^g, that tries to estimate the training dataset distribution $P_{data}(x)$ as $P_g(x)$. It takes as an input a random noise vector z and

produces artificial images G(z) with a distribution of $P_g(z)$ which look like the images in the training dataset.

Discriminator: a discriminative model based on a neural network architecture that is used to identify the real images in training set from the synthetic images generated by the generator.

Training/learning GANs: the training procedure is an iterative process based on a backpropagation algorithm that propagates the classification error of the discriminator to modify the parameters of both the generator and the discriminator models. The generator is trained to make $P_g(x)$ and $P_{data}(x)$ as similar as possible(Lan et al., 2020). Hence, for the generator the target is to find G* represented by (3.1).

$$G^* = \arg\min Div(P_g, P_{data}) \quad (3.1)$$

In a GAN model the discriminator is used to measure the difference between $P_g(x)$ and $P_{data}(x)$. It is a neural network with parameter Θ^d, that performs binary classification, with a binary cross-entropy loss function (3.2), outputting 1 for a real sample x and 0 for an image generated by the generator(Goodfellow et al., 2014).

$$Loss = -(y \log(\hat{y}) + (1-y)\log(1-\hat{y})) \quad (3.2)$$

Where $\hat{y}$ is the probability that the model prediction sample is a positive example and y is the sample label. The value of y for a positive example is 1 and for a negative example is 0. By substituting the positive and negative cases in to P_{data} and P_g, the objective function becomes:

$$V(G,D) = E_x \sim P_{data}\,[\log D(x)] + E_x \sim P_{data}\,[\log(1-D(x))] \quad (3.3)$$

Combining (3.1) and (3.3), the objective function of a basic GAN is:

$$\min_G \max_D V(G,D) = \min_G \max_D E_x \sim P_{data}\,[\log D(x)] + E_x \sim P_{data}\,[\log(1-D(x))] \quad (3.4)$$

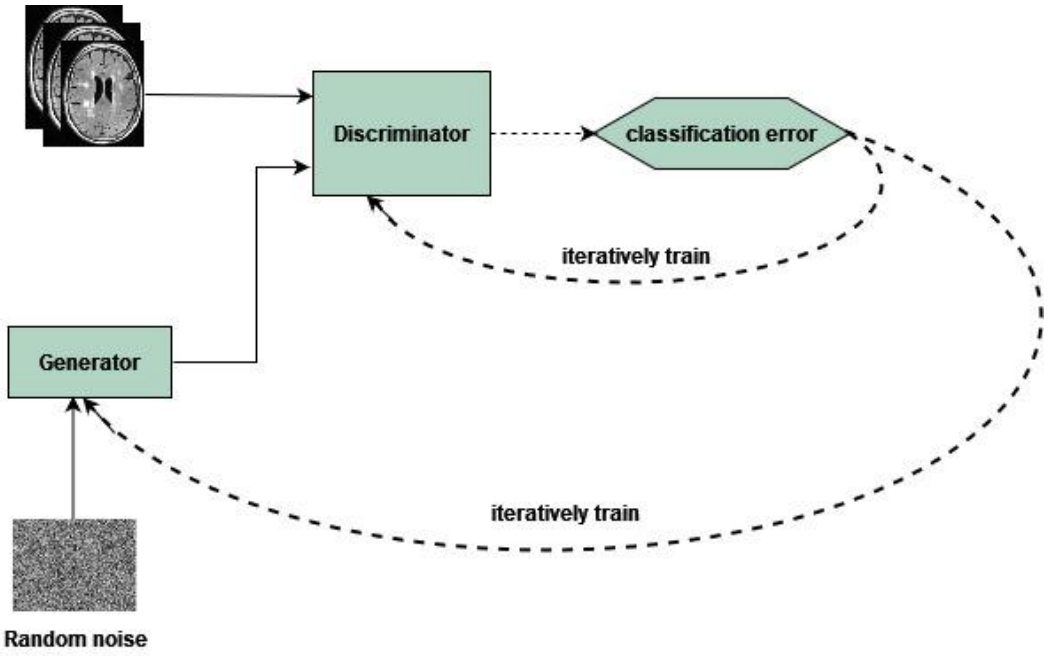


Figure 3.3: A general block diagram representing a GAN architecture. It consists of two neural network models, which are the generator and discriminator. The generator generates real-like images from a random noise vector and the discriminator tries to classify the incoming input as real or fake.

Since the originally proposed GAN models have limitations such as, vanishing gradients, difficulty in training and poor diversity(Wang et al., 2017a), several variants have been proposed to build GANs with better performance(Lan et al., 2020). A Deep Convolutional Generative adversarial network (DCGAN) is an architecturally modified version of GAN that replaces all fully connected layers of the basic GAN with deep convolutional networks(Radford et al., 2015). In addition the discriminator and generator are symmetrical to each other. Pooling layers and up-sampling layers are not included throughout the entire network. Batch normalization is used to solve the problem of vanishing gradients(Lan et al., 2020). Figure 3.4 shows a typical DCGAN, structured based on all convolutional layers.

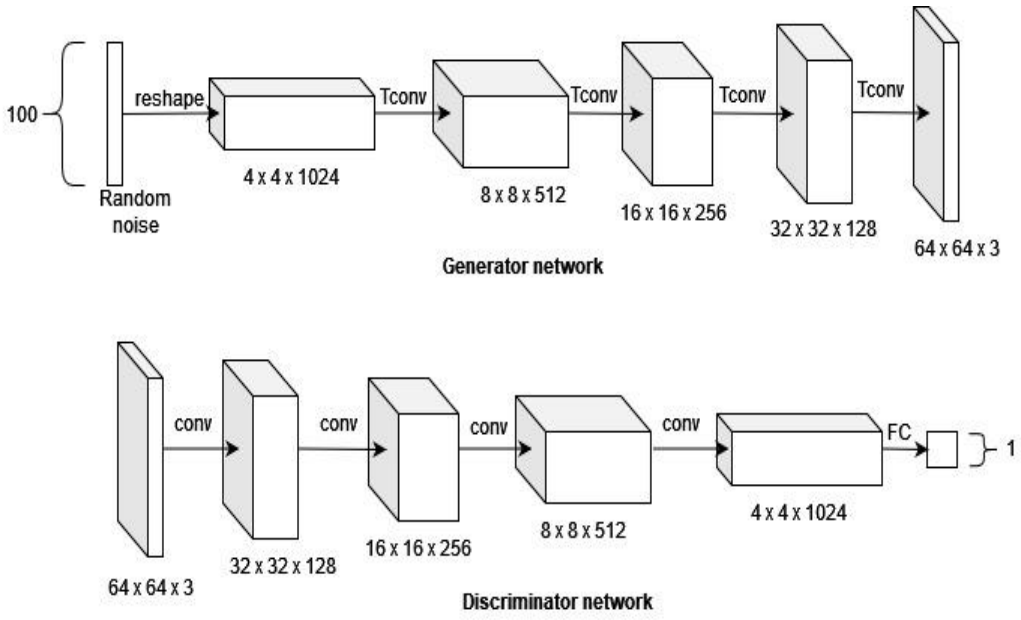


Figure 3.4: An example of a DCGAN that can generate an image with a resolution of 64x64x3. Both the generator and discriminator have a convolutional-based architecture. Starting from a random noise of dimension 100x1, the generator produces an image of resolution 64x64x3. The discriminator accepts both the real images in the training set and synthetic images from the generator to perform - binary classification of real versus fake labels. It takes an image of dimension 64x64x3 and through its de-convolution layers; the size of the 2D array is reduced at each level finally outputting as a real image with 100% probability and 0 for classifying the input as a fake image(Zhang et al., 2020).

Although DCGANs are the best GAN models in terms of model architecture, due to the use of binary crossentropy loss function, it has some limitations. The first drawback is the problem of model collapse, which occurs when the generator starts generating images of only one class while ignoring all other classes. Vanishing gradient is another problem related to using DCGANs. As the confidence-values of the discriminator is a single value that can only be b/w 0 and 1, and the goal is to get a value closer to 1 as much as possible, hence the calculated gradients approach to zero and as a result, the generator is not able to get much information and is not able to learn. So this may result in a strong discriminator, which will lead to a poor generator.

As a solution to these issues, another variant of GAN called Wasserstein generative adversarial network (WGAN, which modifies the loss function to make the training process more stable has been proposed(Arjovsky et al., 2017). It replaces the cross-entropy loss function (JS divergence),

which is not a stable loss metric for measuring the distance between distributions with disjoint parts, by a new distance measurement metric called Wasserstein loss that approximates Earth Mover's Distance (EMD).

$$W(P_g, P_{data}) = \inf_{p \in \prod(P_g, P_{data})} E_{(x, x') \sim p} \|x - x'\| \quad (3.5)$$

Where, $\prod(P_g, P_{data})$ is the set of all joint distributions p whose marginals are p_g and p_{data} respectively. p implies how much mass must be transported from one distribution to another.

EMD is the amount of effort needed to make one distribution to another distribution. In our case we want to make the generated image distribution equal to the real image distribution. By using WGAN, even in the case where two distributions do not overlap, it can still reflect their distance (Arjovsky et al., 2017) .

Since GAN is an emerging AI technology, fewer studies employed GANs to generate synthetic brain images. DCGAN(Kazuhiro et al., 2018), DCGAN with Wasserstien loss function(Rejusha and KS, 2021), WGAN(Han et al., 2018a), MI-GAN and MI-pix2pix(Alogna et al., 2020) are among the different variants of GANs which have been used to generate brain MRI images of different resolution. The quality of the images is usually measured by a human expert only qualitatively. Still, more efforts are needed to generate brain images of high resolution that could expand the dataset size and hence to train deep learning systems with a great prediction performance.

Transfer learning

Transfer learning is a machine learning technique used when there is scarcity in the training data. It aims to extract knowledge from one or more source tasks and applies the knowledge to a target task(Pan and Yang, 2010). The two important concepts related to transfer learning are domain D and task T.

A domain D consists of a feature space χ and a marginal probability P(X) over the feature space, where $X=x_1, \ldots, x_n \in \chi$. Given a domain, $D=\{\chi, P(X)\}$, a task T consist of a label space У and a conditional probability distribution $P(Y \mid X)$ that is typically learned from the training data consisting of pairs $x_i \in X$ and $y_i \in$ У.

For a source and target parameters (D_s and T_s) and (D_t, T_t) respectively, and assuming $D_s \neq D_t$ or $T_s \neq T_t$, by using transfer learning, the information gained from D_s and T_s is used to learn the target conditional probability distribution $P(Y_t \mid X_t)$(Pan and Yang, 2010). In most cases, the source domain is rich with samples, and a large dataset is available to train a deep neural network. While in the target domain, the dataset is small to learn a deep learning model from scratch. In this scenario, transfer learning allows using the knowledge acquired by a model trained on the source domain to improve the performance of carrying out the task in the target domain. Figure 3.5, illustrates the general usage of the transfer of knowledge from the source domain to the target domain.

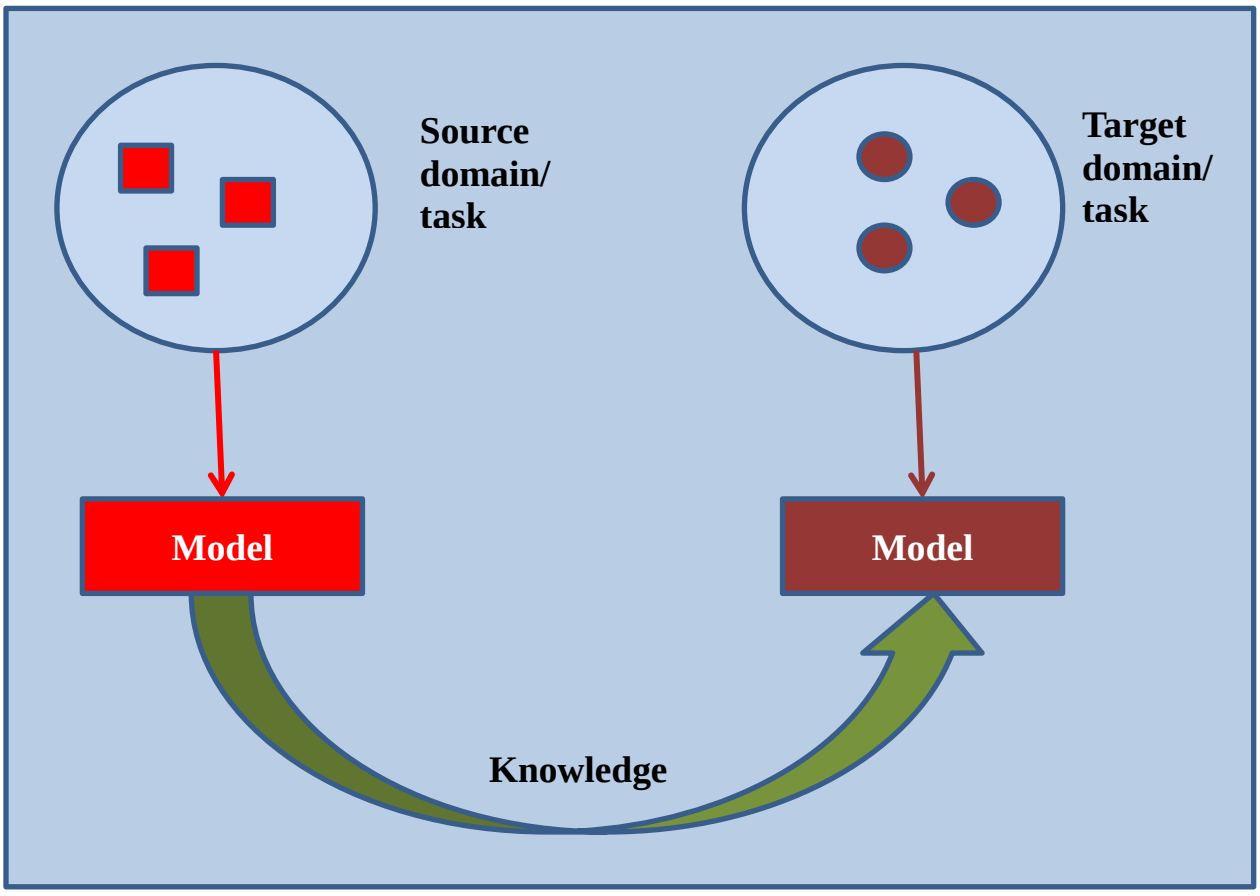


Figure 3.5: Flow of knowledge from the source domain to the target domain using transfer learning.

Considering CNNs, the commonly used approach of transfer learning to pretrain a deep CNN on a very large dataset and use that CNN either as an initialization or a fixed feature extractor for

the target task. ImageNet, which is one of the largest datasets of natural images which contains more than 14 million images and 1000 classes (Simonyan and Zisserman, 2015), has been used to train CNNs with very deep architectures. These models are often used to utilize the knowledge about the leaned features, in terms of parameter values (weights and biases), to improve the performance of models in the target domain where finding large datasets is difficult, such as neuroimaging. In recent times, many neuroimaging studies have employed transfer learning methods to achieve very good results, both for classification and regression tasks.

3.2 Data leakage

The second challenge of employing deep learning models for complex, high dimensional neuroimaging data is data leakage. Data leakage occurs due to the use of information in the model training that is not expected to be available at the prediction time. By introducing data leakage in to the model building process, we will end up with an overly optimistic model with very exciting results on the training and validation set, whereas performing very poorly on unseen test data. Although, neuroimaging literature nowadays is invaded by such a subtle problem producing non relevant models to the clinical scenario, much less attention has been given to following the correct practices to avoid this problem. One of the reasons for this is, even if the theory that data leakage inappropriately inflates the model's performance is known by researchers, the extent of model performance overestimation caused by data leakage has not been assessed very well. Hence, in one of our published papers, we quantitatively investigated the extent of model performance overestimation seen as a consequence of data leakage introduced by performing slice level dataset split while using 2D CNN models.

3.2.1 Effect of data leakage in brain MRI classification using 2D convolutional neural networks

Introduction

In recent years, the number of studies that apply AI tools for different neuroimaging analysis tasks, especially for classifying neuroimaging data has increased significantly. Moreover, most of these studies (Hatcher and Yu, 2018)(Goodfellow et al., 2016a)(Bahrami et al., 2016, Han, 2017, Li et al., 2014, Liu et al., 2018a, Vemulapalli et al., 2017, Zhu et al., 2018)(Chang, 2016,

Dou et al., 2016, Maier et al., 2015)(Liu et al., 2015, Plis et al., 2014)(Liu et al., 2015, Liu et al., 2014, Suk and Shen, 2013)(Kuang et al., 2014)(Vieira et al., 2017)reported very high accuracies in classifying patients with neurological diseases, such as Alzheimer's disease (AD) and Parkinson's disease (PD). For a binary classification of AD vs. healthy controls, Hon and Khan (Hon and Khan, 2017) reported accuracy up to 96.25% using a transfer learning strategy. Sarraf et al. (Sarraf et al., 2016)classified subjects as AD or healthy controls with a subject-level accuracy of 100% by adopting LeNet-5 and GoogleNet network architectures. In other studies, CNNs have been used for performing multi-class discrimination of subjects. Recently, Wu and colleagues (Wu et al., 2018) adopted a pre-trained CaffeNet and achieved accuracy of 98.71%, 72.04%, and 92.35% for a three-way classification between healthy controls, stable mild cognitive impairment (MCI), and progressive MCI patients, respectively. In another work by Islam and Zhang(Islam and Zhang, 2018), an ensemble system of three homogeneous CNNs were proposed, and average multi-class classification accuracy of 93.18% was found on the Open Access Series of Imaging Studies (OASIS) dataset. For the classification of PD, Esmaeilzadeh et al. (Esmaeilzadeh et al., 2018)classified PD patients from healthy controls based on MRI and demographic information (i.e., age and gender). With the proposed 3D model, they achieved 100% accuracy on the test set. In another study by Sivaranjini and Sujatha(Sivaranjini and Sujatha, 2019), a pre-trained 2D CNN AlexNet architecture was used to classify PD patients vs. healthy controls, resulting in an accuracy of 88.9%.

Although excellent performances have been shown by using deep learning for the classification of neurological disorders, there are still many challenges that need to be addressed, including complexity and difficulty in interpreting the results due to highly nonlinear computations, non-reproducibility of the results, and data/information and, especially, data overfitting (see Vieira et al. and Davatzikos for reviews).

Overly optimistic results may be due to data leakage – a process caused by the use of information in the model training that is not expected to be available at the prediction time. See Kaufman et al. (Kaufman et al., 2012) for further details on a formal definition of data leakage. Data leakage can be due to a target (class label) leakage or incorrect data split. For example, data leakage may occur when feature selection is performed based on the whole dataset before cross-validation(Reunanen, 2003, Varma and Simon, 2006). In this case, the target variable of samples

in the test sets may be erroneously used to improve the learning process. Several cases may be related to an incorrect data split. For example, when the data augmentation step is performed before dividing the test set from the training data (late split), the augmented data generated from the same original image can be seen in both training and test data, leading to incorrect inflated performance(Wen et al., 2020). Another form of train-test contamination that leads to data leakage is when the same test set is used to optimize the training hyperparameters and evaluate the model performance(Varma and Simon, 2006). Different use of information not available at prediction time occurs using longitudinal data, when there is a danger of information leaking from the future to the past. A particularly insidious form of data leakage may occur when information about the target inadvertently leaks into the input data, for example the presence of a ruler, markings or treatment devices in a medical image may correlate with the class label(Winkler et al., 2019, Oakden-Rayner et al., 2020, Narla et al., 2018).

While concluding that data leakage leads to overly optimistic results will surprise few practitioners, we believe that the extent to which this is happening in neuroimaging applications is mostly unknown, especially in small datasets. As we completed this study, we became aware of independent research by Wen et al. (Wen et al., 2020)that corroborates part of our conclusions regarding the problem of data leakage. They successfully suggested a framework for the reproducible assessment of AD classification methods. However, the architectures have not been trained and tested on smaller datasets typical of clinical practice, and they mainly employed hold-out model validation strategies rather than cross-validation (CV) – that gives a better indication of how well a model performs on unseen data(Blum et al., 1999, Yadav and Shukla, 2016). Moreover, the authors focused on illustrating the effect of data leakage on the classification of AD patients only.

Unfortunately, the problem of data leakage incurred by incorrect data split is not only limited to the area of AD classification but can also be seen in various neurological disorders. It is more common to observe the data leakage in 2D architectures, yet some forms of data leakage, such as late split, could be present in 3D CNN studies as well. Moreover, although deep complex classifiers are more prone to overfitting, also conventional machine learning algorithms may be affected by data leakage. A summary of these works with clear and potential data leakage is

given in Tables 3.1 and 3.2, respectively. Other works with insufficient information to assess data leakage are reported in Table 3.3.

Table 3.1:Summary of the previous studies performing classification of neurological disorders using MRI and with clear data leakage (see also Appendix 1 online for a detailed description).

Disorder	**Reference**	**Groups (number of subjects)**	**Machine learning model**	**Data split method**	**Type of data leakage**	**Accuracy (%)**
	Gunawardena et al., 2017	AD-MCI-HC (36)	2D CNN	4:1 train/test slice-level split	wrong split	96.00
	Hon & Khan, 2017	AD-HC (200)	2D CNN (VGG16)	4:1 train/test slice-level split	wrong split	96.25
	Jain et al., 2019	AD-MCI-HC (150)	2D CNN (VGG16)	4:1 train/test slice-level split	late and wrong split	95.00
	Khagi et al., 2019	AD-HC (56)	2D CNN (AlexNet GoogLeNet , ResNet50, new CNN)	6:2:2 train/validation/test slice-level split	wrong split	98.00
AD/MCI	Sarraf et al., 2017	AD-HC (43)	2D CNN (LeNet-5)	3:1:1 train/validation/test slice-level split	wrong split	96.85
	Wang et al., 2017	MCI-HC (629)	2D CNN	Data augmentation + 10:3:3 train/validation/test split by MRI slices	wrong split and augmentation before split	90.60
	Puranik et al., 2018	AD/EMCI-HC	2D CNN	17:3 train/test split by MRI slices	wrong split	98.40
	Basheera et al., 2019	AD-HC	2D CNN	4:1 train/test split by MRI slices	wrong split	90.47
	Nawaz et al., 2020	AD-MCI-HC	2D CNN	6:2:2 slice level split	wrong split	99.89

AD = Alzheimer's disease; HC = Healthy controls; MCI = Mild cognitive impairment.

Table 3.2:Summary of the previous studies performing classification of neurological disorders using MRI and suspected to have potential data leakage (see also Appendix 2 online for a detailed description).

Disorder	**Reference**	**Groups (number of subjects)**	**Machine learning model**	**Data split method**	**Type of data leakage**	**Accuracy (%)**
	Farooq et al., 2017	AD-MCI-LMCI-HC (355)	2D CNN (GoogLeNet and modified ResNet)	3:1 train/test (potential) slice-level split	wrong split	98.80
AD/MCI	Ramzan et al., 2019	HC-SMC-EMCI-MCI-LMCI-AD (138)	2D CNN (ResNet-18)	7:2:1 train/ validation/test (potential) slice-level split	wrong split	100
	Raza et al., 2019	AD-HC (432)	2D CNN (AlexNet)	4:1 train/test (potential) slice-level split	wrong split	98.74
	Pathak et al., 2020	AD-HC	2D CNN	3:1 (potential) slice level split	wrong split	91.75
ASD	Libero et al., 2015	ASD-TD (37)	Decision tree	unclear	entire data set used for feature selection	91.90
	Zhou et al., 2014	ASD-TD/HC (280)	Random tree classifier	4:1 train/test split	entire data set used for feature selection	100
PD	Sivaranjini, et al., 2019	PD-HC (182)	2D CNN	4:1 train/test split by MRI slices	wrong split	88.90
TBI	Lui et al., 2014	TBI-HC (47)	Multilayer perceptron	10-fold CV	entire data set used for feature selection	86.00
Brain tumor	Hasan et al., 2019	Tumor-HC (600)	MGLCM+ 2D CNN + SVM	10-fold CV	wrong split and entire data set used for feature selection	99.30

AD = Alzheimer's disease; ASD = Autism spectrum disorder; HC = Healthy controls; MCI = Mild cognitive impairment; PD = Parkinson's disease; SWEDD = scans without evidence of dopaminergic deficit; TBI = Traumatic brain injury; TD = Typically developing.

Table 3.3: Summary of the previous studies performing classification of neurological disorders using MRI and that provide insufficient information to assess data leakage (see also Appendix 3 online for a detailed description).

Disorder	**Reference**	**Groups (number of subjects)**	**Machine learning model**	**Data split method**	**Accuracy (%)**
	Al-Khuzaie et al., 2021	AD-HC (240)	2D CNN	(potential) slice-level split	99.30
AD/MCI	Wu et al., 2018	AD-HC	2D CNN	Data augmentation + 2:1 train/test split by MRI slices	97.58

AD = Alzheimer's disease; HC = Healthy controls; MCI = Mild cognitive impairment.

In this study, we addressed the issue of data leakage in one of the most common classes of deep learning models, i.e., 2D CNNs, caused by incorrect dataset split of 3D MRI data. Specifically, we quantified the effect of data leakage on CNN models trained on different datasets of T_1-weighted brain MRI of healthy controls and patients with neurological disorders using a nested CV scheme with two different data split strategies: a) subject-level split, avoiding any form of data leakage and b) slice-level split, in which different slices of the same subject are contained both in the training and the test folds (thus data leakage will occur). We focused our attention on both large (about 200 subjects) and small (about 30 subjects) datasets to evaluate a possible increase in performance overestimation when a smaller dataset was used, as is often the case in clinical practice. This paper expands on the preliminary results by Yagis et al.(Yagis et al.,

2019), offering a broader investigation of the issue. In particular, we performed the classification of AD patients using the following datasets: 1) OASIS-200, consisting of randomly sampled 100 AD patients and 100 healthy controls from the OASIS-1 study(Marcus et al., 2007), 2) ADNI, including 100 AD patients and 100 healthy controls randomly sampled from Alzheimer's Disease Neuroimaging Initiative (ADNI)(Petersen et al., 2010), and 3) OASIS-34, composed of 34 subjects (17 AD patients and 17 healthy controls) randomly selected from the OASIS-200 dataset. Given that the performance of a model trained on a small sample dataset could depend on the selected samples, we created ten instances of the OASIS-34 dataset by randomly sampling from the OASIS-200 dataset ten times independently. The subject IDs included in each instance are found in Appendix 4 online. Moreover, we generated a different dataset, called OASIS-random, where, for each subject of the OASIS-200 dataset, a fake random label of either AD patient or healthy control was assigned. In this case, the image data had no relationship with the assigned labels. Besides, we included two T_1-weighted images datasets of patients with de-novo PD: PPMI, including 100 de-novo PD patients and 100 healthy controls randomly chosen from the public Parkinson's Progression Markers Initiative (PPMI) dataset(Marek et al., 2018), and Versilia, a small-sized private clinical dataset of 17 patients with de-novo PD and 17 healthy controls. A detailed description of each dataset has been reported in the "Methods" section.

Materials and methods

Datasets

In this study, we adopted the scans collected by three public and international datasets of T_1-weighted images of patients with AD (the OASIS dataset (Marcus et al., 2007) and the ADNI dataset(Petersen et al., 2010)) and de-novo PD (the PPMI dataset (Marek, K. *et al., 2018*)). An additional private de-novo PD dataset, namely the Versilia dataset, has also been used. A summary of the demographics of the datasets used in this study is shown in Table 3.4. In the following sections, a detailed description of all datasets will be reported.

Table 3.4:Demographic features of subjects belonging to OASIS-200, ADNI, PPMI, and Versilia datasets. The same information for the OASIS-34 datasets has been reported in Appendix 5 online.

Dataset		**Patients**	**Healthy controls**
OASIS-200	Number of subjects	100	100
	Age (range, years)	62 – 96	59 – 94
	Age (mean ± SD, years)	76.70 ± 7.10	75.50 ± 9.10
	Gender (women/men)	59/41	73/27
ADNI	Number of subjects	100	100
	Age (range, years)	56 – 89	58 – 95
	Age (mean ± SD, years)	74.28 ± 7.96	75.04 ± 7.11
	Gender (women/men)	44/56	52/48
PPMI	Number of subjects	100	100
	Age (range, years)	34 – 82	31 – 83
	Age (mean ± SD, years)	61.71 ± 9.99	61.91 ± 11.52
	Gender (women/men)	40/60	36/64
Versilia	Number of subjects	17	17
	Age (range, years)	48 – 78	54 – 77
	Age (mean ± SD, years)	64 ± 7.21	64.00 ± 7.00
	Gender (women/men)	4/13	5/12

AD = Alzheimer's disease; ADNI = Alzheimer's Disease Neuroimaging Initiative; OASIS = Open Access Series of Imaging Studies; PD = Parkinson's disease; PPMI = Parkinson's Progression Markers Initiative; SD = standard deviation.

OASIS-200, OASIS-34, and OASIS-random datasets

We have used the T_1-weighted images of 100 AD patients [(59 women and 41 men, age 76.70 ± 7.10 years, mean ± standard deviation (SD)] and 100 healthy controls (73 women and 27 men, age 75.50 ± 9.10 years, mean ± SD) from the OASIS-1 study – a cross-sectional cohort of the OASIS brain MRI dataset(Marcus et al., 2007), freely available at https://www.oasis-brains.org/.

In particular, we have employed the same scans that were previously selected by other authors(Hon and Khan, 2017). We called this dataset OASIS-200. The subject identification numbers (IDs) and demographics of these subjects were specified in Appendix 6 online. No significant difference in age ($p = 0.15$ at t-test) was found between the two groups, while a significant (borderline) difference in gender was observed ($p = 0.04$ at χ^2-test).

In OASIS-1, AD diagnosis, as well as the severity of the disease, were evaluated based on the global Clinical Dementia Rating (CDR) score derived from individual CDR scores for the domains memory, orientation, judgment and problem solving, function in community affairs, home and hobbies, and personal case(Morris, 1993, Morris et al., 2001). Subjects with a global CDR score of 0 have been labeled as healthy controls, while scores 0.5 (very mild), 1 (mild), 2 (moderate), and 3 (severe) have been all labeled as AD.

All T_1-weighted images have been acquired on a 1.5 T MR scanner (Vision, Siemens, Erlangen, Germany), using a Magnetization Prepared Rapid Gradient Echo (MPRAGE) sequence in a sagittal plane [repetition time (TR) = 9.7 ms, echo time (TE) = 4.0 ms, flip angle = 10°, inversion time (TI) = 20 ms, delay time (TD) = 200 ms, voxel size = 1 mm × 1 mm × 1.25 mm, matrix size = 256 × 256, number of slices = 128](Marcus et al., 2007).

ADNI dataset

We considered the T_1-weighted MRI data of 100 AD patients (44 women and 56 men, age 74.28 ± 7.96 years, mean ± SD) and 100 healthy controls (52 women and 48 men, age 75.04 ± 7.11 years, mean ± SD). No significant difference in age ($p = 0.24$ at t-test) and gender ($p = 0.26$ at χ^2-test) was found between the two groups. AD patients have been randomly chosen from the ADNI 2 dataset (available at http://adni.loni.usc.edu/) – a cohort of ADNI that extends the work of ADNI 1 and ADNI-GO studies(Petersen et al., 2010). Led by Principal Investigator Michael W. Weiner, MD, ADNI was launched in 2003 to investigate if biological markers (such as MRI and PET) can be combined to define the progression of MCI and early AD. We have used MPRAGE T_1-weighted MRI scans acquired by 3 T scanners [6 Siemens (Erlangen, Germany) MRI scanners and 6 Philips (Amsterdam, Netherlands) scanners] in a sagittal plane (voxel size = 1 mm × 1 mm × 1.2 mm). The image size of the T_1-weighted data acquired from the Siemens and Philips scanners were 176 × 240 × 256 and 170 × 256 × 256, respectively. Since ADNI 2 is a

longitudinal dataset, more than one scan was available for each subject. The first scan of each participant has been chosen to produce a cross-sectional dataset. Appendix 7 provides subject IDs and the acquisition date of the specific scan used in our study. The MRI acquisition protocol for each MRI scanner can be found at http://adni.loni.usc.edu/methods/documents/mri-protocols/. In ADNI 2 dataset, subjects have been categorized as AD patients or healthy controls based on whether subjects have complaints about their memory and by considering a combination of neuropsychological clinical scores(Petersen et al., 2010).

PPMI dataset

We randomly selected 100 de-novo PD subjects (40 women and 60 men, age 61.71 ± 9.99, mean ± SD) and 100 healthy controls (36 women and 64 men, age 61.91 ± 11.52, mean ± SD) from the publicly available PPMI dataset (https://ida.loni.usc.edu/login.jsp?project=PPMI). No significant difference in age ($p = 0.44$ at t-test) and gender ($p = 0.56$ at χ^2-test) was found between the two groups. The criterion used to recruit de-novo PD patients, and healthy controls were defined by Marek and colleagues(Marek et al., 2018). Briefly, PD patients were selected within two years of diagnosis with a Hoehn and Yahr score < 3(Hoehn and Yahr, 1967), at least two of resting tremor, either bradykinesia or rigidity (must have either resting tremor or asymmetric bradykinesia) or a single asymmetric resting tremor or asymmetric bradykinesia and dopamine transporter (DAT) or vesicular monoamine transporter type 2 (VMAT-2) imaging showing a dopaminergic deficit. Healthy controls were free from any clinically significant neurological disorder(Marek et al., 2018).

The T_1-weighted scans were collected at baseline using MR scanners manufactured by Siemens (11 scanners at 3 T and five scanners at 1.5 T), Philips Medical Systems (10 scanners at 3 T and 11 scanners at 1.5 T), GE Medical Systems (11 scanners at 3 T and 24 scanners at 1.5 T) and another anonymous one (5 scanners at 1.5 T). We also found three subjects whose MRI protocol was missing. The details of the MRI protocols of all scanners can be found in Appendix 8.

Versilia dataset

Seventeen (4 women and 13 men, age 64 ± 7.21 years, mean ± SD) patients with de-novo parkinsonian syndrome consecutively referred to a Neurology Unit to evaluate PD over a 24-

month interval (from June 2012 to June 2014) were recruited in this dataset. More details about clinical evaluation can be found in(Tessa et al., 2019). Seventeen healthy controls (5 women and 12 men, age 64 ± 7 years, mean ± SD) with no history of neurological diseases and normal neurological examination were recruited as controls. No significant difference in age ($p = 0.95$ at t-test) and gender ($p = 0.70$ at χ^2-test) was found between the two groups.

All subjects underwent high-resolution 3D T_1-weighted imaging on a 1.5 T MR scanner system (Magnetom Avanto, software version Syngo MR B17, Siemens, Erlangen-Germany) equipped with a 12-element matrix radiofrequency head coil and SQ-engine gradients. The SQ-engine gradients had a maximum strength of 45 mT/m and a slew rate of 200 T/m/s. T_1-weighted MR images were acquired with an axial high resolution 3D MPRAGE sequence with TR = 1900 ms, TE = 3.44 ms, TI = 1100 ms, flip angle = 15^o, slice thickness = 0.86 mm, field of view (FOV) = 220 mm×220 mm, matrix size = 256×256, number of excitations (NEX) = 2, number of slices = 176.

T_1-weighted MRI data preprocessing

All T_1-weighted MRI data went through two preprocessing steps (see Figure 3.6). In the first stage, co-registration to a standard template space and skull stripping were applied to re-align all the images and remove non-brain regions. In the second stage, a subset of axial images has been collected using an entropy-based slice selection approach.

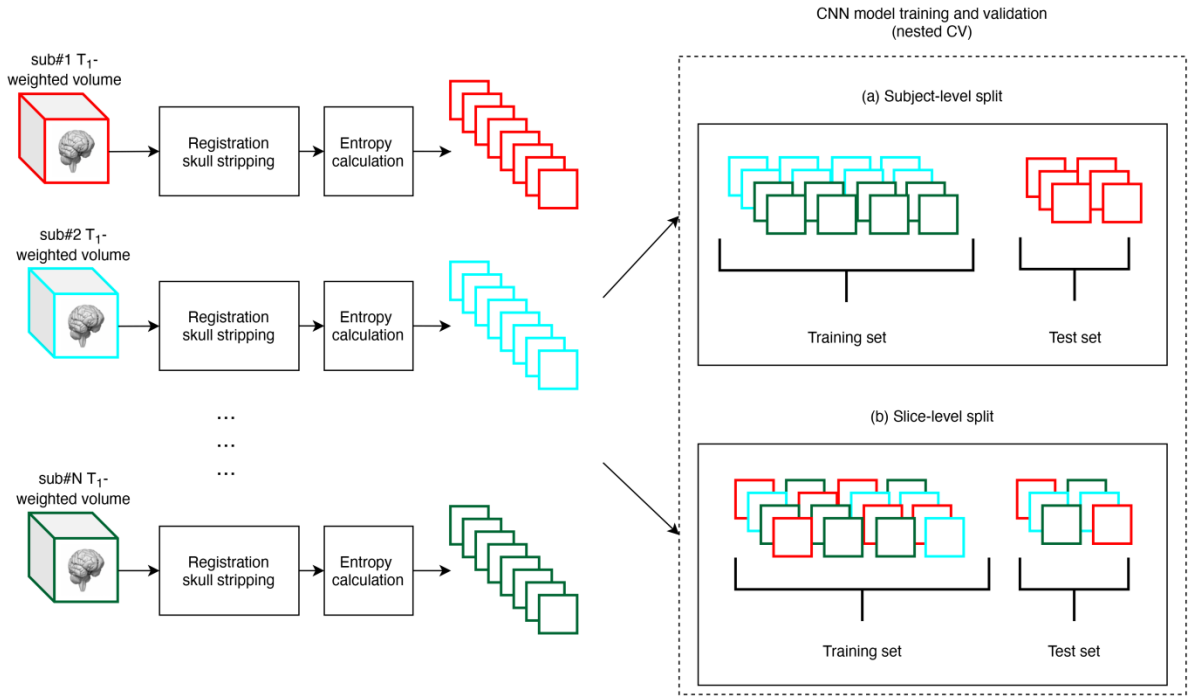


Figure 3.6: Schematic diagram of the overall T1-weighted MRI data processing and validation scheme. First, a preprocessing stage included co-registration to a standard space, skull-stripping and slices selection based on entropy calculation. Then, CNNs model's training and validation have been performed on each dataset in a nested CV loop using two different data split strategies: a) subject-level split, in which all the slices of a subject have been placed either in training or in the test set, avoiding any form of data leakage; b)slice-level split, in which all the slices have been pooled together before CV, then split randomly in to training and test set.

Co-registration to a standard template space and skull stripping

For the OASIS datasets, we used publicly available preprocessed data (gain-field corrected, brain masked, and co-registration)(Han et al., 2018b). Briefly, the brain masks from OASIS were obtained using an atlas-registration-based method, and their quality was controlled by human experts(Marcus et al., 2007), and each volume has been co-registered to the Talairach and Tournoux atlas. Each preprocessed T_1-weighted volume had a data matrix size of 176 × 208 × 176 and a voxel size of 1 mm × 1 mm × 1 mm(Han et al., 2018b).

For all other datasets, we have co-registered each individual T_1-weighted volume to the MNI152 standard template space (at 1 mm voxel size – available in the FSL version 6.0.3 package) by using the SyN algorithm included in ANTs package (version 2.1.0) with default parameters (Avants, B. B. *et al., 2011*). Then, the brain mask of the standard template space has been

applied to each co-registered volume. Each preprocessed T_1-weighted volume had a data matrix size of 182 × 218 × 182 and a voxel size of 1 mm × 1 mm × 1 mm. Figure 3.7 illustrates sample preprocessed T_1-weighted slices from OASIS-200, ADNI, PPMI, and Versilia datasets.

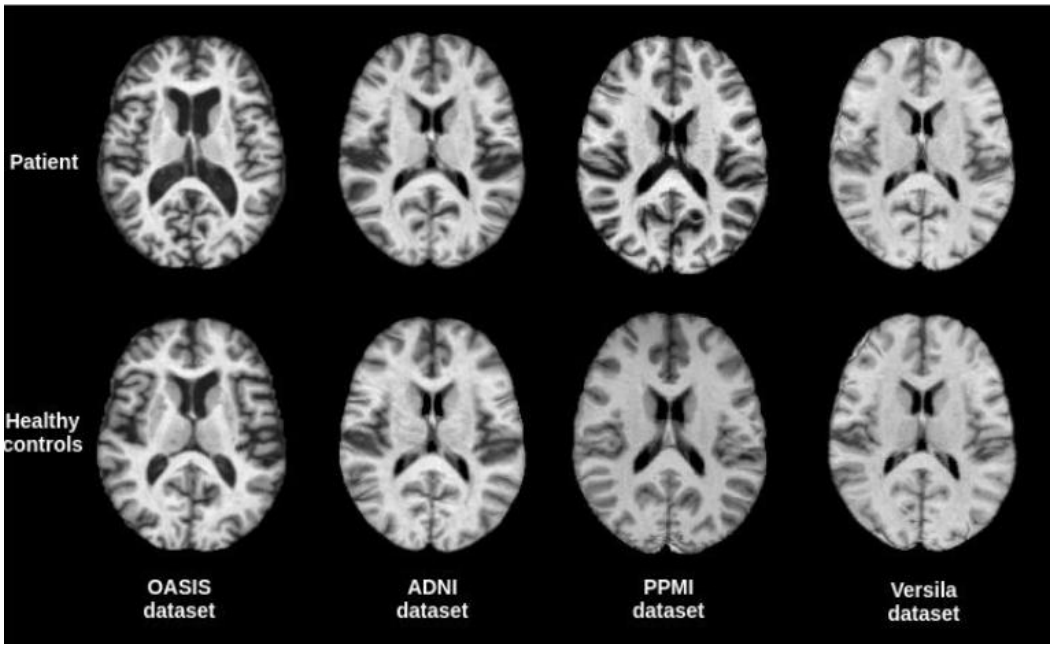


Figure 3.7: Sample preprocessed T1-weighted axial images from OASIS-200, ADNI, PPMI and Versilia datasets.

Entropy-based slice selection

Each T_1-weighted slice generally conveys a different amount of information. Given that we are interested in developing a 2D CNN model, we have performed a preliminary slice selection based on the amount of information. More specifically, for each T_1-weighted volume, the Shannon entropy E_S, representing the information content, was computed for each axial slice, as follows:

$$E_S = \sum_k p_k \log_2(p_k) \tag{3.6}$$

where k is the number of grayscale levels in the slice and p_k is the probability of occurrence, estimated as the relative frequency in the image, for the gray level k. Then, for each T_1-weighted volume, the slices were ordered in descending order based on their entropy scores, and, finally,

we selected only the eight axial slices that showed the highest entropy (Hon, M. & Khan, N., 2017).

To be consistent with the input sizes of the proposed 2D CNN models, all slices were resized to 224 × 224 pixels by fitting a cubic spline between the 4-by-4 neighborhood pixels[66]. Voxel-wise feature standardization has also been applied to make training the CNNs easier and achieve faster convergence, i.e., for each voxel, an average value of all grayscale values within the brain mask has been subtracted and scaled by the standard deviation (within the brain mask).

Model architectures

Since the number of subjects of each dataset may not be sufficient to train with high accuracy a 2D CNN model from scratch, we have used a machine learning technique called transfer learning that allows employing pre-trained models, i.e., model parameters previously developed for one task (source domain) to be transferred to target domain for weight initialization and feature extraction. In particular, CNN layers hierarchically extract features starting from the general low-level features to those specific to the target class, and using transfer learning, the general low-level features can be shared across tasks. Notably, we used pre-trained VGG16 (Simonyan and Zisserman, 2015) and ResNet-18 (He et al., 2015)models in this study, as detailed in the following sections. The transfer learning approach and VGG16 architectures used in this study are similar to those employed in (Hon and Khan, 2017) as their results triggered our investigation of data leakage.

VGG16-based models

VGG16 is one of the most influential architectures which explore network depth with very small (3x3) convolution filters stacked on top of each other. VGG16 consists of five convolutional blocks, with alternating convolutional and pooling layers and three fully-connected layers.

In transfer learning, the most common approach is copying the first *n* layers of the pre-trained network to the first *n* layers of a target network and then randomly initializing the remaining layers to be trained on the target task. Depending on the size of the target dataset and the number of parameters in the first *n* layers, these copied features can be left unchanged (i.e., frozen) or fine-tuned during the training of the network on a new dataset. It is well accepted that if the target dataset is relatively small, fine-tuning may cause overfitting, whereas if the target

dataset is large, then the base features can be fine-tuned to improve the model's performance without overfitting.

To investigate the effect of fine-tuning, we have tested two different variants of VGG16 architecture, namely VGG16-v1 and VGG16-v2 (Figure 3.8). The former model has been used as a feature extractor where the weights for all network layers are frozen except that of the final fully connected layer. Randomly initialized fully connected layers have replaced the three topmost layers with rectified linear unit (ReLU) activation. The weights are initialized according to the Xavier initialization heuristic (Glorot and Bengio, 2010) to prevent the gradients from vanishing or exploding.

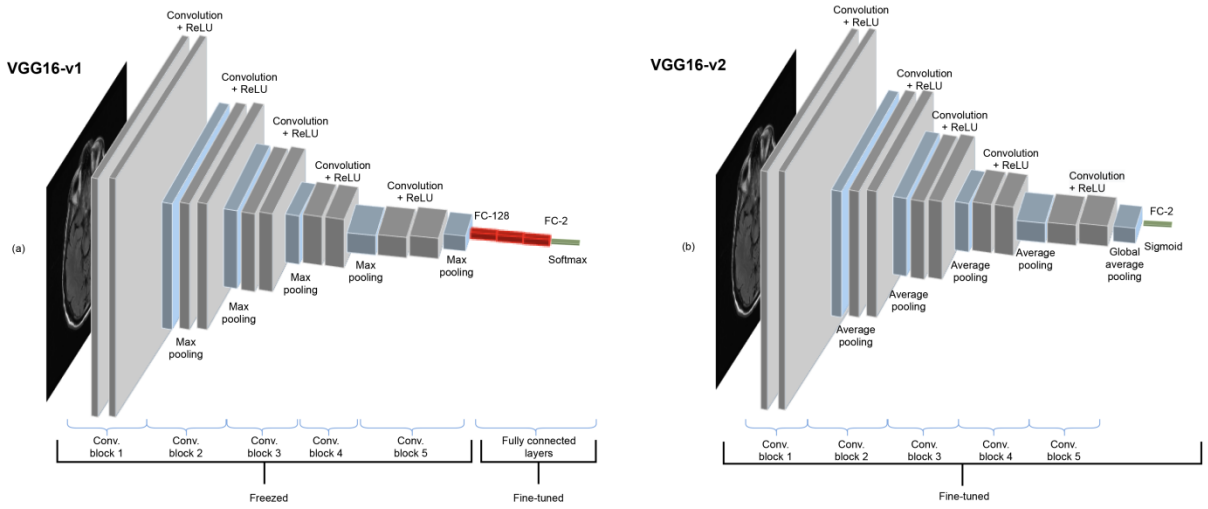


Figure 3.8: The two different networks based on the VGG16 architecture are shown. Each colored block of layers illustrates a series of convolutions. (a) The first model, named VGG16-v1 consists of five convolutional blocks followed by three fully connected layers. Only the last three fully connected layers are fine-tuned, b) On the other hand, the second model, VGG16-v2, has five convolutional blocks followed by a global average pooling layer, and all the layers are fine-tuned.

The VGG16-v2 model has been utilized as a weight initializer where the weights are derived from the pre-trained network and fine-tuned during training. We have replaced the fully connected layers with a randomly initialized global average pooling (GAP) layer suggested by Lin and colleagues (Lin, M. Et al., 2014) to reduce the number of parameters and, rather than freezing the CNN layers, we have fine-tuned all layers.

ResNet-18 based model

It has been long believed that deeper networks can learn more complex nonlinear relationships than shallower networks with the same number of neurons, and thus network depth is of great importance on model performance(Szegedy et al., 2015). However, many studies revealed that deeper networks often converge at a higher training and test error rate when compared to their shallower counterparts(He et al., 2015). Therefore, stacking more layers to the plain networks may eventually degrade the model's performance while complicating the optimization process. To overcome this issue, He and colleagues introduced deep residual neural networks and achieved top-5 test accuracies with their models on the popular ImageNet test set(He et al., 2015). The model was proposed as an attempt to solve the vanishing gradients and the degradation problems using residual blocks. With these residual blocks, the feature of any deeper unit can be computed as the sum of the activation of a shallower unit and the residual function. This architecture causes the gradient to be directly propagated to shallower units making ResNets easier to train.

There are different versions of residual neural network (ResNet) architecture with various numbers of layers. In this work, we used ResNet-18 architecture, an 18-layer residual deep learning network consisting of five stages, each with a convolution and identity block(He et al., 2015). In our model, one fully connected layer with sigmoid activation has been added at the end of the network – a common practice in binary classification tasks as it takes a real-valued input and squashes the output to a range between 0 and 1. Since the network is relatively smaller and has a lower number of parameters than VGG16, the weights and biases of all the transferred layers are fine-tuned while the newly added fully connected layer has been trained to start from randomly initialized weights. The architecture of our ResNet-18 model can be seen in Figure 3.9.

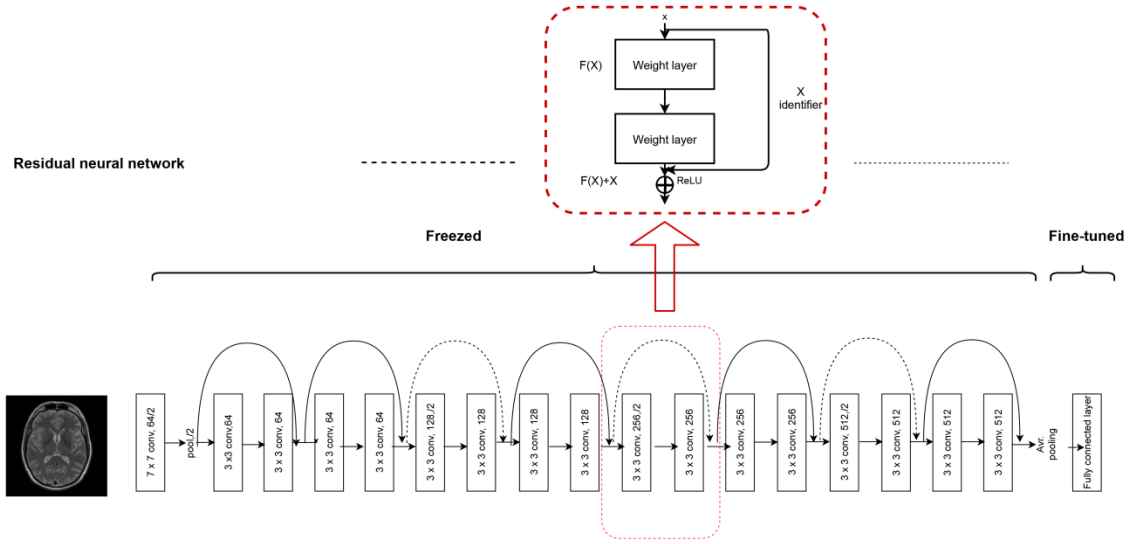


Figure 3.9: A modified ResNet-18 architecture with an average pooling layer at the end is shown. The upper box represents a residual learning block with an identity shortcut. Each layer is denoted as (filter size, # channels); layers labeled as "freezed" indicates that the weightes are not updated during backpropagation, whereas when they are labeled as "fine-tuned" they are updated. The identity shortcuts can be directly used when the input and output are of the same dimensions (solid line shortcuts) and when the dimensions increase (dotted line shortcuts). ReLU=rectified linear unit.

Model training and validation

Each 2D CNN model has been trained and validated using a nested CV strategy – a validation scheme that allows examining the unbiased generalization performance of the trained models along with performing, at the same time, hyperparameters optimization(Varma and Simon, 2006). It involves nesting two CV loops where the inner loop is used for optimizing model hyperparameters, and the outer loop gives an unbiased estimate of the performance of the best model. It is especially suitable when the amount of data available is insufficient to allow separate validation and test sets(Varma and Simon, 2006). A schematic diagram of the procedure is illustrated in Figure 2.8 of Section 2.3. It starts by dividing the dataset into *k* folds, and one-fold is kept as a test set (outer CV), while the other *k*-1 folds are split into inner folds (inner CV). The model hyperparameters are chosen from the hyperparameter space through a grid search based on the average performance of the model over the inner folds. In particular, we varied the learning rate in the set $\{10^{-5}, 3 \times 10^{-5}, 10^{-4}, 3 \times 10^{-4}, 10^{-3}\}$ and the learning rate decay in {0, 0.1, 0.3, 0.5}. The chosen model is then fitted with all the outer fold training data and tested on the

unseen test fold, resulting in an unbiased estimation of the model's prediction error. Specifically, we choose a 10-fold CV because it offers a favorable bias-variance tradeoff(Hastie et al., 2009, Lemm et al., 2011).

In all experiments, we used batch size = 128 and epoch number = 50. Due to its ability to adaptively updating individual learning rates for each parameter, an Adam optimizer was used(Kingma and Ba, 2014). Each selected slice of the 3D T_1-weighted volume has been classified independently and the final model's performance was stated using the mean slice-level accuracy, separately, on the training set and test set folds of the outer CV.

We thus conducted CNNs model's training and validation on each dataset in a nested CV loop using two different data split strategies: a) subject-level split, in which all the slices of a subject have been placed either in the training set or in the test set, avoiding any form of data leakage; b) slice-level split, in which all the slices have been pooled together before CV, then split randomly into training and test set. In this case, for each slice of the test set, a set of highly correlated slices coming from the MR volume of the same subject ended up in the training set, giving rise to data leakage, as shown pictographically in Figure 3.6.

CNN models were carried out using a custom-made software in Python language (version 3.6.8) using the following modules: CUDA v.9.0.176(Cook, 2014), TensorFlow-gpu v.1.12.0(2016), Keras v.2.2.4(Chollet and others, 2015), Scikit-learn v.0.20.2(Pedregosa et al., 2011), Nibabel v.2.3.3(Brett et al., 2019), and OpenCV v.3.3.0(Bradski and Kaehler, 2008). All the source code can be found in a Github repository at https://github.com/Imaging-AI-for-Health-virtual-lab/Slice-Level-Data-Leakage, and a Docker image can be downloaded at https://hub.docker.com/repository/docker/ai4healthvlab/slice-level-data-leakage. The training and validation of CNN models were performed on a workstation equipped with a 12 GB G5X frame buffer NVIDIA TITAN X (Pascal) GPU with 64 GB RAM, 8 CPUs, 3584 CUDA cores and 11.4 Gbps processing speed. The average computational time for CNN training on a dataset of 34 and 200 subjects were 5.68 hours (VGG16-v1), 5.63 hours (VGG16-v2), 2.94 hours (ResNet-18) and 33.93 hours (VGG16-v1), 33.82 hours (VGG16-v2), 14.12 hours (ResNet-18), respectively. The total computational time for this study was thus about 17 days.

Results

For AD classification, accuracy on the test set, using subject-level CV, was below 71% for large datasets (OASIS-200 and ADNI), whereas they were below 59% for smaller datasets (OASIS-34). Regarding de novo PD classification, they were around 50% for both large (PPMI) and small (Versilia) datasets. Conversely, slice-level CV erroneously produced very high classification accuracies on the test set in all datasets (higher than 94% and 92% on large and small datasets, respectively), leading to deceptive, over-optimistic results (Table 3.5).

The worst-case stemmed from the randomly labeled OASIS dataset, which resulted in a model with unacceptably high performances (accuracy on the test set more than 93%) using slice-level CV, whereas classification results obtained using a subject-level CV were about 50%, in accordance with the expected outcomes for a balanced dataset with completely random labels.

Table 3.5:Mean slice-level accuracy on the training and test set of the outer CV over 5-fold nested CV has been reported for three 2D CNN models (see "Methods" section), all datasets, and two data split methods (slice-level and subject-level). The difference between accuracy using slice-level and subject-level split in the test set has also been reported.

Dataset	Network architecture	Training set accuracy (%)		Test set accuracy (%)		
		Subject-level split	Slice-level split	Subject-level split	Slice-level split	Difference
OASIS-200	VGG16-v1	95.93	99.85	66.0	94.18	28.18
	VGG16-v2	95.13	100	66.13	96.99	30.86
	ResNet-18	100	100	68.87	98.96	30.1
OASIS-34	VGG16-v1	88.94	100	54.35	99.19	44.84
	VGG16-v2	96.94	100	54.34	99.33	44.99
	ResNet-18	100	100	57.49	98.96	41.47
OASIS-Random	VGG16-v1	63.38	100	53.37	95.93	42.56
	VGG16-v2	69.17	100	49.25	94.81	45.56
	ResNet-18	84.49	99.09	50.8	93.74	42.94
ADNI	VGG16-v1	91.09	100	70.12	95.31	25.19
	VGG16-v2	80.49	100	66.49	95.24	28.75
	ResNet-18	100	100	68.68	96.87	30.19
PPMI	VGG16-v1	76.8	100	48.24	93.99	45.75
	VGG16-v2	73.19	100	46.93	94.37	47.44
	ResNet-18	100	100	48.06	96.12	44.06
Versilia	VGG16-v1	99.72	100	53.86	95.97	42.11
	VGG16-v2	76.89	100	42.97	97.8	54.83
	ResNet-18	99.90	95.13	51.36	92.63	41.27

3.3 Interpretability

Interpretability is one of the major drawbacks of the application of deep learning systems in areas where the rationale for the model's decision is a requirement for trust, such as in the healthcare domain(Lipton, 2018). It describes how understandable is the link between the features used by a machine learning algorithm and the prediction it produces(Reyes et al., 2020). Although deep neural networks have been showing empirical success in medicine, including in neuroimaging, the complexity of these models that stem from the very deep architecture consisting of hundreds of layers and millions of parameters makes it difficult to understand the internal states of the model(Kimura and Tanaka, 2020). This is because the number and complexity of the model's features directly affect the interpretability of the model(Lipton, 2018). Due to these black-box properties, the clinical employment of these methods still needs building of further trust by incorporating more interpretability in to these systems. In medical imaging, also in neuroimaging analysis the desire to include interpretability in to the predictive systems is due to the reasons:

In order to adopt these tools for clinical applications, building trust between the model and the users is important, where the trust depends on the level of understanding of the internal states of these models. For example, in scenarios, where a model performed well during the model development phase and becomes less reliable when applied to real-world data, interpretability provides hints to judge if the model is trustworthy in a given scenario supported by expertise's knowledge(Sheu, 2020, Lipton, 2018).

Model interpretation helps to get more understanding and to discover novel aspects of the data(Sheu, 2020). For instance, in neuroimaging model visualization tools, which are one type of interpretability approaches, can help in finding new neural correlates associated with different brain disorders. Considering the more general legal side, model interpretability is explicitly stated as a requirement by the General Data Protection Regulation set by the European Union(Regulation, 2018).

To increase trust in the use of AI tools for medical imaging application and to incorporate these tools in a clinical setup, several AI explainability methods are being developed. Nevertheless, the larger number of studies employed attribution-based approaches for developing deep learning systems in medical imaging(Singh et al., 2020).

Attribution-based methods work by assigning a relevance to each input of a model to determine the contribution of each feature to the target neuron, which is often the output neuron of the correct class for a classification problem. Generally these methods produce attribution maps or heatmaps that represent the importance of different parts of the image to the model decision on a pixel-by-pixel basis. Usually positive attribution is marked in red and negative attribution is marked in blue(Singh et al., 2020, Huff et al., 2021). Most of attribution-based interpretation methods are implemented after building a model and are only meaningful when applied to a fully trained model. Hence these methods are known by a name, post-hoc explanations(Huff et al., 2021). The most commonly used attribution based explainability methods in medical imaging studies include:

Occlusion maps

The occlusion map, which was introduced first by Zeiler and Fergus(Zeiler and Fergus, 2014), is one of the simplest attribution-based interpretability methods, which is categorized as a perturbation-based approach(Singh et al., 2020). Perturbation techniques remove, mask or modify specific input features, then run the forward pass and compare the difference from the original output(Singh et al., 2020). To identify and highlight important regions on the input image, in the case of CNNs, part of the image is masked usually by a grey patch, and it is passed through the model (Figure 3.10). Parts of the image that strongly affect the output of the model when occluded are assigned high relevance and image parts that have less impact on the output rather assigned a low relevance(Huff et al., 2021). Since occluding each pixel in the image is very expensive, usually a patch of size, 5 x 5 or 10 x 10 is used to generate an occlusion map of a CNN. The major limitation of this approach is that it is computationally expensive as many forward passes are required to generate the occlusion maps.

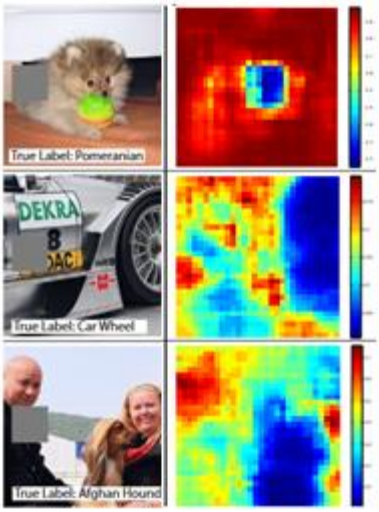


Figure 3.10: Occlusion map experiment by Zeiler and Fergus (Zeiler & Fergus, 2014) - was performed by occluding the images to the left and the generated occlusion heatmaps at the last classification layer.

Deep SHapley Additive exPlanations (SHAP)

Another perturbation-based approach is a SHAP explainer(Lundberg and Lee, 2017), which is an extension of a Shapley value in cooperative game theory, as a method for calculating the contribution of each feature in machine learning. Considering each feature as a player of a team, the contribution or relevance of each feature is represented by a SHAP value, where this value is computed by determining the difference between the predicted values with and without addition of each feature for all combinations of features and taking the average value. This allows determining the feature's influence on the output or prediction and whether the influence is positive or negative.

Saliency maps:

Saliency maps are another type of attribution-based explainability methods which are classified as back-propagation or gradient-based methods. Gradient-based methods rely on a computation of the gradient, when a test sample is forward propagated and back-propagated through a trained network(Simonyan et al., 2014).

Saliency maps specifically use the absolute value of the partial derivative of the target output neuron with respect to the input features to find the features which affect the output the most

with the least perturbation(Singh et al., 2020). Considering the computational time, these are very fast algorithms at the expense of a weaker relationship between the outcome and the variation of the output.

The main drawback of saliency map methods is the absence of indication as to whether a pixel provides evidence for or against a class, only that the classification is sensitive to that pixel(Huff et al., 2021). In addition, in the case of binary classification, saliency maps lose their class specificity, because if a feature is vital for distinguishing between two classes, it may be highlighted by a saliency map for both classes. In the equation, the heatmap $Sal_c(x)$ for a class *c* is computed directly as the derivative of the model output score $F_c(x)$ with respect to each pixel in the input image *x* through backpropagation(Huff et al., 2021):

$$Sal_c(x) = \frac{dF_c(x)}{dx} \tag{3.7}$$

Gradient-weighted class activation maps (GradCAM):

GradCAM explanations correspond to the gradients of the class score with respect to each feature map of the last convolutional unit(Selvaraju et al., 2017). These approaches focused on the features with a positive association with the class of interest. In formula(Huff et al., 2021):

$$a_k^c = \frac{dy_c}{df_k(x)} \tag{3.8}$$

Where, the weights a_k^c are the gradients of the score for class c y_c with respect to the kth feature maps $f_k(x)$ of the preceding convolutional layer:

$$GradCAM_c(x) = ReLU(\sum_k a_k^c f_k(x)) \tag{3.9}$$

Chapter 4

4. Development of interpretable, leakage-free and reproducible deep learning framework for analyzing neuroimaging data

To overcome the limitations seen in most of deep learning studies applied to neuroimaging data, a set of python functions (which is available at https://github.com/Imaging-AI-for-Health-virtual-lab/Slice-Level-Data-Leakage) have been developed, which allow to build 2D CNN-based models with a wide range of model development and validation options for both classification and regression tasks.

4.1 Main features of our deep learning framework

Wide model architecture choice

Our deep learning framework allows choosing a wide variety of model architectures. Most of the model architectures are of 2D CNN types. Since we believe that most of the openly available brain image datasets have relatively small size, training a deep CNN model from scratch would lead to overfitting and results in models with poor generalization performance (a more detailed explanation is found in Section 3.1). Hence in our framework, multiple pre-trained 2D models are provided and the user can adapt the models based on the desired task. The pre-trained models included in our software were taken from one of the top deep learning frameworks, called Keras. The pre-trained weights included are:

- VGG16
- DenseNet121
- Xception
- Resnet50 and ResNet18
- MobileNetV2
- InceptionResNetV2

Single input and multi-input CNNs:

Single input CNN: is a CNN that accepts only one type of input, basically a single type of brain image, such as slices of T1-weighted or T2-weighted FLAIR MRI data.

Multi-input CNN: a CNN, accepting two inputs. The pre-trained CNN models are adapted to accept two image inputs (by combining two different brain images eg., T1-weighted with T2-weighted or with DTI images) or they can be designed to take one image and another numerical data (demographic or clinical data).

2D and 3D models:

Although most of the models included have a 2D architecture, for the reason that to use widely available 2D pre-trained models, a 3D model architecture option is also available.

Flexibility of model training and validation options

All the models included in our framework are trained based on transfer learning techniques for both classification and regression problems. Each of the pre-trained models can be used either as feature extractor or as a weight initializer. The user can choose between the two options by setting the argument *'Nlfrez'* to determine the number of layers to freeze or to fine-tune.

Regarding model validation, three approaches are included.

- Holdout validation;
- K-fold CV; and
- Nested k-fold CV.

Although the choice of the validation technique to use can be made according to the theoretical principles described in Section 2.5, for a task requiring only model training and evaluation with hyperparameters specified by the user, either holdout or k-fold CV can be used. However, model selection is carried out in a nested k-fold CV loop (the implementation details will be shown in the next section, See Section 4.3).

No data leakage (Reliability)

All data preparation, data pre-processing and model training sub-tasks are designed carefully to avoid any form of data leakage. Basically the two useful considerations done in our deep learning tool are:

For both 2D and 3D models, feature scaling and normalization was performed on the statistics of the training set. Specifically mean and standard deviation of the training set was computed and these values were used to do feature scaling and normalization of the training, validation and test set. By doing so, the training dataset will not have knowledge about the distribution of the validation and test sets.

In the case of 2D CNNs, since the models take 2D MRI images, 3D MRI volumes have to be sliced in to 2D images using one of the anatomical planes (axial, coronal, or sagittal). During the model validation procedure, the dataset is split in to train, validation and test sets based on subject level to prevent the type of data leakage which was explained in Section 3.2.

In case of 3D architecture, since our models take the whole 3D volume of MRI, there is no chance of incurring data leakage while dividing the data in to training, validation and test sets.

Interpretability

As interpretability of the results of a deep learning system is a crucial component to getting trust of the predictive tool, as already highlighted in the previous chapter (section 3.3), multiple choices of CNN visualization methods are included in the predictive toolkit for both classification and regression tasks. The available visualization approaches for classification are:

- Saliency maps;
- GradCAM;
- Occlusion maps; and
- SHAP method.

While for regression,

- Saliency maps; and
- GradCAM

This feature of the toolkit is available for the validation techniques of holdout and k-fold CV. In addition, at the current level of development, the visualization techniques are included only for singleinput CNNs.

Presentation of results

The way the outputs of the deep learning analysis are presented depends on the task, type of analysis in terms of validation techniques and the type of CNN model architecture.

For a regression task, the available outputs include:

For a simple model training and evaluation, in the case of holdout and k-fold CV, the MRI image type, image IDs included in the training and test set, the model architecture, the evaluation outputs for each fold, including both training and validation MSE and Pearson's correlation coefficient are presented. In addition training and validation learning curves are drawn (Figure 4.1).

```
/home/vmci/MRI_images/T1_63VMCI_masked.nii.gz
processing ***** T1 ******images
opening volume 0
opening volume 1
opening volume 2
opening volume 3
opening volume 4
opening volume 5
```

(a)

```
image size: (928, 182, 218)
labels size (928,)
Data processing done!
Data processing time in minutes: 0.40747132698694866
Image IDs selected as test samples: [16 35 14 55 42  4 52 30 47  5  6 46 29 18 56 44 15 13  8 17 57  1 22  7
 43 53 40 27  0 33 26  9 23 45 39 24 32 49 11 20 54 37 41 31 38 19 12  2
 50 48 10 34  3 51 21 36 25 28]
Fold Number 0
all train data shape (832, 182, 218, 1)
all validation data shape (96, 182, 218, 1)
```

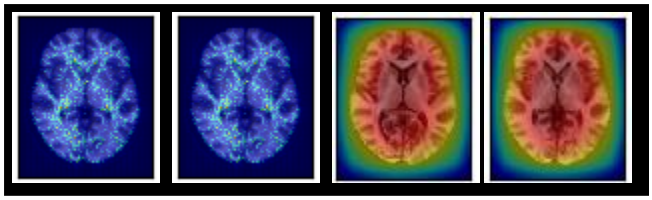
(b)

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         (None, 182, 218, 3)       0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 182, 218, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 182, 218, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 91, 109, 64)       0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 91, 109, 128)      73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 91, 109, 128)      147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 45, 54, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 45, 54, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 45, 54, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 45, 54, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 22, 27, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 22, 27, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 22, 27, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 22, 27, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 11, 13, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 11, 13, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 11, 13, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 11, 13, 512)       2359808
_________________________________________________________________
gap1 (GlobalAveragePooling2D (None, 512)               0
_________________________________________________________________
d1 (Dense)                   (None, 1)                 513
=================================================================
Total params: 14,715,201
Trainable params: 14,715,201
Non-trainable params: 0
_________________________________________________________________
training samples **************************** 832
```

(c)

```
fitting time in minutes: 73.28195561170578
THE AVERAGE CORRELATION COEFFICIENT OF THE 10 FOLDS OVER THE TRAINING SET IS 0.3475248619595053
THE AVERAGE CORRELATION COEFFICIENT OF THE 10 FOLDS OVER THE TEST SET IS 0.365740539226681
```

(d)



e)

Figure 4.1: Outputs presented for regression analysis. (a) path to the MRI dataset, (b) size of training and validation datasets, image indices selected as a validation set, and the fold number,

(c) the CNN model architecture, (d) the model's average performance on the important regions of the image for prediction, and (e) saliency and GradCAM heatmaps generated by the CNN visualization tools. See section 3.3 for a detailed explanation.

For parameter optimization, for each outer fold of nested CV, the list of the hyperparameter grid is reported at the beginning of the nested loop (see Figure 4.2). At the end of the execution, the best hyperparameter configuration with the performance evaluated on the chosen best configuration is printed out (Figure 4.3).

```
>>> Hyperas search space:

def get_space():
    return {
        'lr': hp.choice('lr', [0.0001,0.0003, 0.001,0.003]),
        'decay': hp.choice('decay', [0.5]),
        'weight_decay': hp.choice('weight_decay', [0.02,0.025,0.03]),
    }
```

Figure 4.2: The hyperparameter space to search for the best configuration of the analysis.

```
evaluation/ correlation coefficient on unseen test set: 0.2556931674480438
loss on unseen test set: 0.031767770648002625
Hyperparameter optimization is done!
Optimization time: 444.1223098794619
Processing time: 448.8055600444476
```

Figure 4.3: The output of one fold of nested CV loop.

For a classification problem also, similar outputs are presented except for few differences.

- Visualization of CNNs: in the case of classification, the following visualization images are displaced.

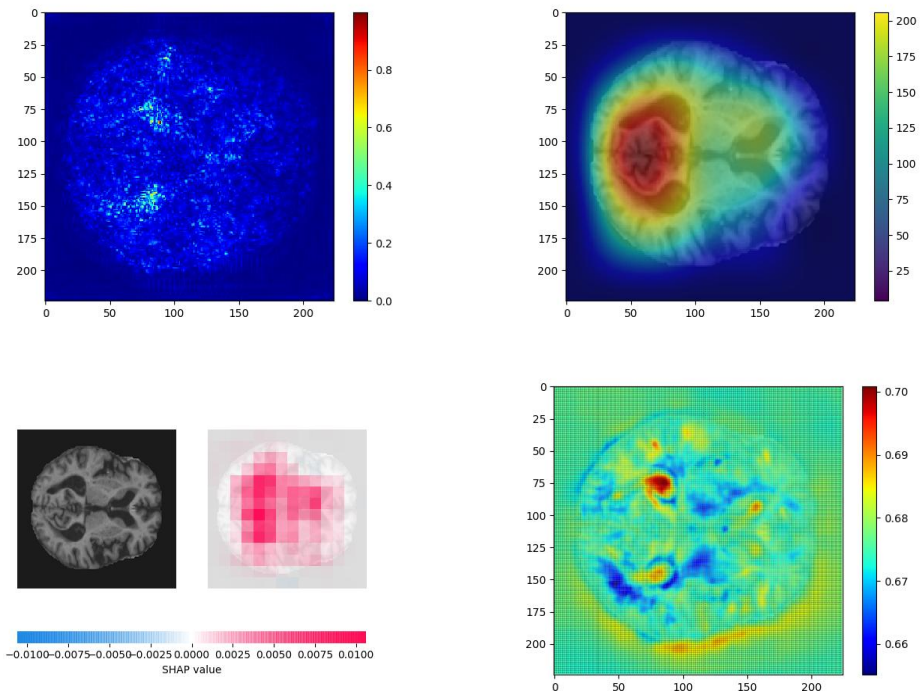


Figure 4.4: An example of visualization output of an AD slice classified correctly for binary classification of AD vs HC, whose learning curve is shown in Figure 4.5.

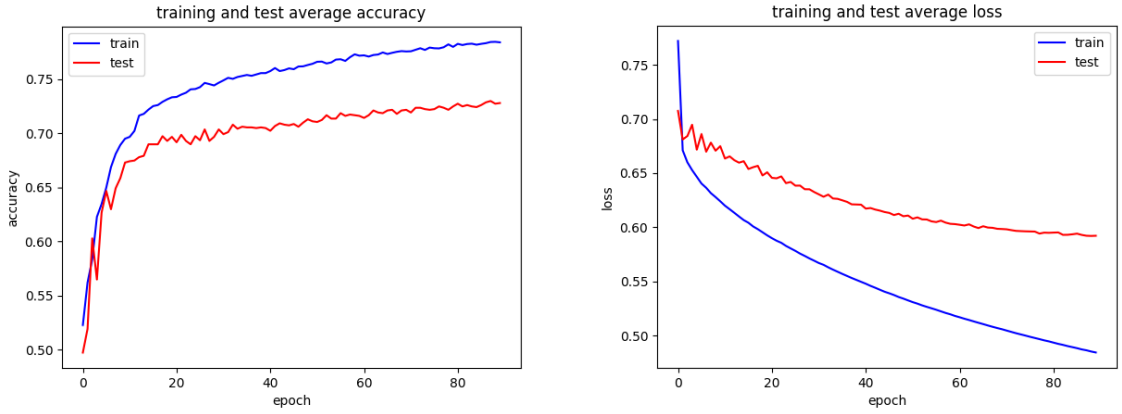


Figure 4.5: Learning curves for a binary classification task between AD and HC group.

4.2 General structure of the software

The structure of our deep learning framework is categorized in to three functions. The main functions included are:

all_net_train.py: this is the main analysis function to perform the model training and validation. The inputs to this function are:

- MRI dataset: is a 4D array in a neuroimaging informatics technology initiative (NIFTI) file format;
- Label data: a file consisting of subject IDs and the label assigned for each participant;
- Data_config: a configuration file that includes information needed for data preparation and pre-processing; and .
- Arch_config: a configuration file that contains information about the model architecture, training setup and validation method.

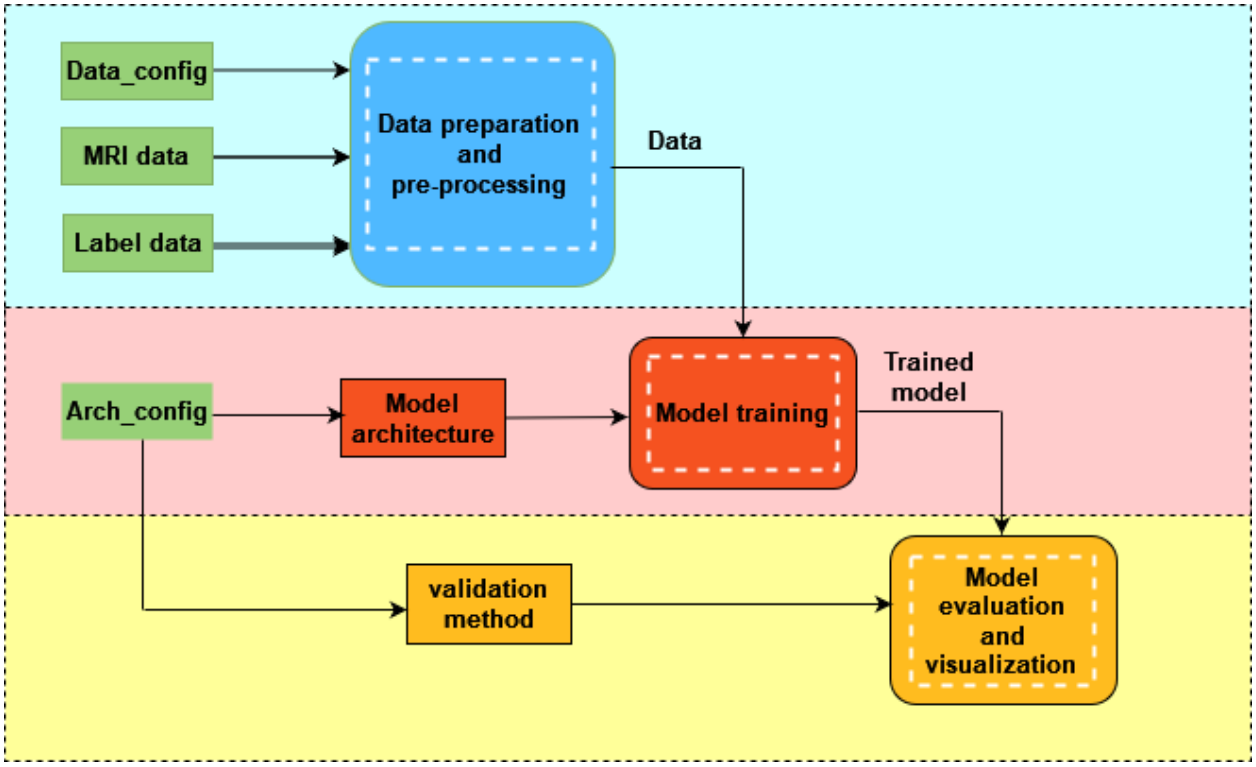


Figure 4.6: General overview of the deep learning framework.

By taking these input information, the all_net_train.py function. Performs:

1. Data pre-processing: the different stages of data pre-processing include:
    - ✓ slicing of the 3D MRI volume of each subject using either of the axial, coronal or sagittal planes
    - ✓ selecting a subset of slices based on entropy-based slice selection
    - ✓ dividing the 2D image dataset based on subject-level in to training-validation or training-validation-test sets to prevent data leakage as explained in section 3.2;
    - ✓ normalizing the 2D MRI slices using the training dataset statistics;
    - ✓ These sub-processes are executed to convert the raw neuroimaging data in to an appropriate format to be fed to the CNN model.
2. Model training: a pre-trained model of the user's choice is selected from many available pre-trained models, which is specified in the arch_config.json file, and by applying modifications of the user's interest, the model architecture is built, the model is then trained on the training dataset and validated on the validation set.
3. Model evaluation: the trained model is evaluated based on the defined metric function.

Visualize_cnn.py: is the function that is called up after the main analysis has been completed. It carries out model interpretation tasks to reason out the results of the main analysis. Multiple CNN visualization techniques are included for both classification (GradCAM, Saliency map, Shap and occlusion map) and regression (GradCAM, Saliency map) analysis.

This method accepts the trained model and test dataset samples, which are results of the all_net_train.py execution, and returns visualization output images highlighting the important brain regions on the test sample images.

gan_train.py: is the function used to generate synthetic brain MRI images.

4.3 Code development

The design and implementation of the code was founded on the basis of knowledge defined by following the suggestions on the web.

Data preparation

The input MRI data is imported from a NIFTI file format. It is a format defined by the data format working group (DFWG) in two meetings held at National Institute of Health (NIH). It was proposed to resolve the issue of the absence of orientation information in the previous data formats. It is adapted from the widely used ANALYZE format and uses the empty space in the ANALYZE header to add more important features. The new features include:

1. Affine coordinate definitions relating voxel index (*i, j, k*) to spatial location (*x, y, z*);
2. Codes to indicate spatio-temporal slice ordering for FMRI;
3. "Complete" set of 8-128 bit data types;
4. A standardized way to store vector-valued datasets over 1-4 dimensional domains;
5. Codes to indicate data "meaning";
6. A standardized way to add "extension" data to the header; and
7. Dual file (.hdr & .img) or single file (.nii) storage.

Each NIfTI file contains metadata and a voxel in up to 7 dimensions and supports a variety of data types. Usually NIfTI files have a .nii or .nii.gz extension containing both the header and the data. NIfTI files can be split into a binary header (.hdr) and image data (.img/.img.gz). NIfTI metadata provides additional information about the coordinate system and how to interpret the data of the image. This may include parameters such as intent, a description, or fMRI-specific metadata.

Data import is performed using a function included in the python library called “NiBabel” and it is loaded as a 4D Numpy array. The ‘Generate_data’ function in our framework slices the series of 3D MRI volumes in to 2D images and resizes them to the required image resolution. Other methods single2three_channel, generate iterator, holdout_validate and cv_validate perform conversion to threechannel (RGB) image, split data in to training/validation and test sets based on subject level normalize the image features.

Model training and validation

Our deep learning system allows choosing between three validation schemes holdout, k-fold CV, and nested k-fold CV. The holdout and k-fold CV validation methods are implemented using the

functionalities of the keras library, which is one of the popular deep learning framework. For both of these validation schemes, specifying a fixed set of hyperparameters is required.

In the case of a nested k-fold CV, it is implemented by nesting two k-fold CV loops. Starting from the outer loop, first a k-fold CV is applied to the whole dataset on the basis of subjects creating *N_outer* folds. Looping over the outer folds, while each fold will get a chance to be a test set, the remaining *N_outer* -1 folds are merged to be used for the model selection procedure performed by the inner k-fold CV. For a single input CNN, a method called GridSearchCV, provided by scikit-learn takes the training data and applies two loops of execution, the first iterating over the grid of hyperparameters and for each combination of hyperparameters performs an internal k-fold CV loop. For a multi-input CNN instead, a python library called Hyperas is used to perform hyperparameter tunning as it supports optimizing multi-input models. Overall, by running the nested CV, the inner loop selects the best model, and on the outer loop, the chosen model is evaluated on the unseen test set. For evaluating the goodness of the hyperparameter grid, a statistical metric accuracy (ACC) is used for classification tasks and Pearson's correlation coefficient for regression problems. This procedure of model selection and evaluation are summarized in Figure 4.7.

Model visualization

The visualization method also performs iteratively to generate a visualization map of each of the images included in the test or the validation set. For each test image, all heatmap images (occlusion map, saliency map, GradCAM heatmap and SHAP heatmap) are produced since all of the visualization functions are included in the single Visualize_cnn.py script. The out put images are saved as PNG files.

For binary classification, the test samples are categorized as True_positive, False_positive True_negative and False_negative to allow better analysis of the visualization results.

While the function for the occlusion map is implemented using the keras library, the visualize_saliency and visualize_cam functions provided by the 'keras-vis' python module are customized and integrated in to our system. For generating SHAP heatmaps also, the GradientExplainer function included in the SHAP python package is applied on the test brain images.

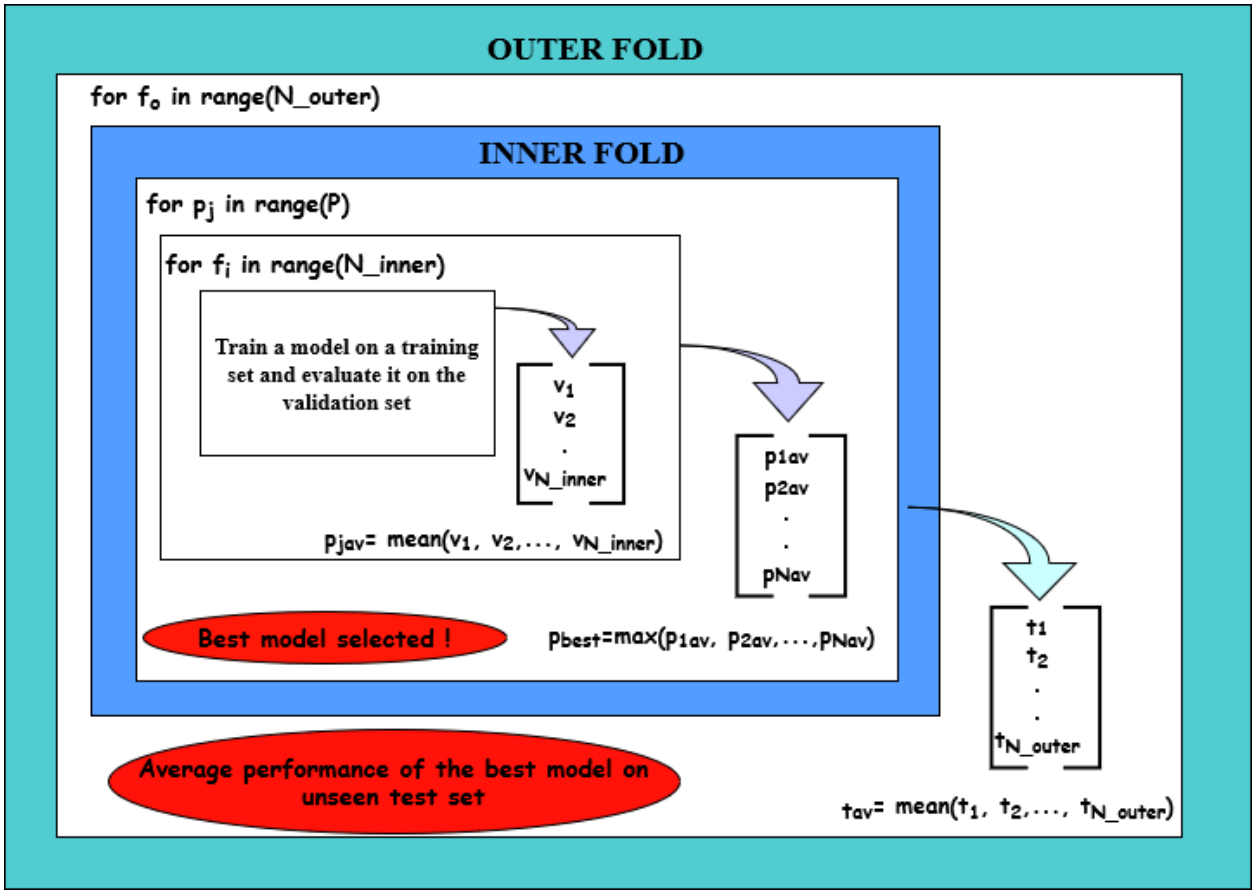


Figure 4.7: Schematic representing a nested CV. It involves three loops of execution, the outer k-fold CV, the iteration over the hyperparameter space and the inner k-fold CV. After the dataset is divided in to N_outer folds, for each of the outer folds fo, where $o \in \{1, 2, 3, ..., N\_outer\}$, model selection is performed by running the inner loop loop f_i , where $i \in \{1, 2, 3, ..., N_{\_inner}\}$ for eachc possible configuration of the hyperparameter p_i, where $j \in \{1, 2, 3, ..., P\}$.

Chapter 5

5. Applications of deep learning in neuroimaging

In this chapter, we discuss applications of the deep learning system we developed for analyzing neuroimaging data. The first project that has employed our deep learning system is the "VMCI Tuscany" study. In this study, we primarily aimed at developing a deep learning system capable of predicting a wide range of raw and demographically adjusted neuropsychological clinical scores of SVD subjects with MCI from neuroimaging data. In addition, we highlight the neural correlates that are influential in the cognitive degradation seen in SVD patients.

In the second project, we focused on developing a robust deep learning system that can diagnose Alzheimer's disease from neuroimaging data. The visualization results of our system also show important brain substrates associated with the cognitive decline caused by Alzheimer's disease.

5.1 Prediction of the cognitive performance in patients with small vessel disease and mild cognitive impairment: a deep learning approach

5.1.1 Introduction

Vascular mild cognitive impairment (MCI) is an intermediate state between normal status and dementia, with evidence of measurable cognitive impairment, but maintaining independence in activities of daily living (Moorhouse and Rockwood, 2008). Cerebral small vessel disease, which is the second most frequent cause of cognitive decline (Smith, 2017), is known as the main cause of vascular MCI (SVD) (Vasquez and Zakzanis, 2015). SVD is a brain disease characterized by several heterogeneous pathological changes affecting small arteries, arterioles, venules, and brain capillaries (Pantoni, 2010; Vasquez and Zakzanis, 2015). In patients with SVD and MCI, a wide range of brain, such as subcortical infarcts, lacunes, white matter (WM) T_2-hyperintensities, dilated perivascular spaces, micro-bleeds, and brain atrophy can be revealed by MRI (Wardlaw et al., 2013). Nevertheless, the impact of each of these features on cognitive abilities overall and in the single domains is not yet established (Pantoni et al., 2019).

Machine learning methods have been widely applied to neuroimaging and enabled better understanding of normal brain structure and functions and identifying signatures of different brain disorders (Davatzikos, 2019). Few machine learning studies (Ciulli et al., 2016; Pantoni et al., 2019; Shi et al., 2018) have addressed the analysis of the above MRI features with the cognitive status in patients with SVD and MCI. They assessed the ability of different conventional machine learning approaches by extracting MRI features from T_1-weighted and T_2-weighted fluid-attenuated inversion recovery (FLAIR) images and diffusion tensor imaging (DTI). Overall, these machine learning studies reported a good to excellent capability of the MRI features to predict comprehensive or domain-specific cognitive scores (Ciulli et al., 2016; Pantoni et al., 2019; Shi et al., 2018). However, machine learning methods require a hand-crafted and complex feature extraction procedure on manual or semiautomatic drawn regions of interest (ROIs). Recently, deep learning approaches have become very popular in medical image processing because they can extract complex patterns from high dimensional data while retaining spatial information without or with a reduced need for ROI definitions or editing for feature extraction (Plis et al., 2014). Deep learning methods are representation learning techniques based on artificial neural network architecture and known for their deep architecture that hierarchically extract complex levels of data abstractions using simple nonlinear functions(LeCun et al., 2015). Specifically, convolutional neural networks (CNNs) are convenient for analyzing MRI data (Plis et al., 2014) because their architecture is specialized in the extraction of latent patterns from structured data like multi-dimensional arrays, such as, for example, images(LeCun et al., 2015).

In this study, we assessed the ability of patients with SVD and MCI to predict the overall neuropsychological performance and, specifically, the performance in attention and executive functions tests – two of the cognitive domains typically earlier and more severely compromised (O'Brien et al., 2003) – using a deep learning approach. A multi-input CNN-based system is trained and evaluated in a 10-fold nested cross-validation loop. In particular, we separately fed T_1-weighted images, T_2-weighted FLAIR images, and DTI-derived mean diffusivity (MD) and fractional anisotropy (FA) maps into a CNN model along with demographic data. Since demographic information, such as education, age and sex are known to have a strong association with cognitive status(Casanova et al., 2020), we combined brain image and demographic features to predict the cognitive test scores. Through a transfer learning approach, the CNN-based system identified brain image patterns associated with each cognitive ability from each input image

modality to estimate individual neuropsychological scores. Hence, we assessed and compared the ability of different MRI-derived data (T_1-weighted images, T_2-weighted FLAIR images, MD, and FA maps) to predict the cognitive scores in patients with SVD and MCI.

1. Materials and methods

In this section, we explain the MRI dataset, the preprocessing image pipeline, CNN model training, and evaluation to predict neuropsychological test scores in patients with SVD and MCI.

Subjects

In this study, 58 patients [27 women and 31 men, aged 74.18 ± 6.98 years, mean ± standard deviation (SD)] with SVD and MCI from the VMCI-Tuscany study were considered (Poggesi et al., 2012). From 64 subjects, which were examined in a previous study (Pantoni et al., 2019), we removed six patients in whom data for one or more cognitive scores were missing. We thus included 58 patients with MCI according to an *ad hoc* operationalization of Winblad criteria (Salvadori et al., 2016; Winblad et al., 2004) and evidence on MRI of moderate-to-severe WM T_2 hyperintensities on the modified Fazekas scale(Pantoni, 2010)). Demographic data and descriptive statistics of neuropsychological scores of the dataset are shown in Table 5.1.

Cognitive evaluation

Each participant underwent a comprehensive neuropsychological evaluation through the VMCI-Tuscany neuropsychological battery – a comprehensive tool specifically developed for patients with SVD and MCI(Salvadori et al., 2015). The VMCI-Tuscany neuropsychological battery includes both global cognitive functioning tests and second-level tests covering different cognitive domains. Among the cognitive tests of the VMCI-Tuscany neuropsychological battery, we selected those sensitive to attention and executive dysfunctions for this experiment. This is because these are prominent features of subcortical vascular cognitive impairment (O'Brien et al., 2003). The cognitive tests included: 1) Montreal cognitive assessment (MoCA): a global efficiency test sensitive to attention and executive functions (score range 0–30: higher scores represent better performance); 2) trail making test part-A (TMT-A): a visual scanning and

Table 5.1: Demographic data and descriptive statistics of neuropsychological scores in the sample of 58 patients with SVD and MCI. mean ± SD (min – max).

Demographic data	
Number of patients	58
Sex (women/men)	27/31
Age (years)	74.18 ± 6.98 (59.80 – 89.03)
Education (years)	8.12 ± 4.17 (3.00 – 18.00)
Cognitive score	
MoCA	20.89 ± 4.42 (8.00 – 28.00)
SDMT	22.94 ± 11.55 (3.00 – 49.00)
TMT-A	89.08 ± 49.91 (25.60 – 238.00)
ROC-F immediate copy	21.25 ± 8.33 (2.00 – 36.00)
Stroop	51.41 ± 30.44 (12.00 – 169.00)
Visual search	30.63 ± 8.58 (10.00 –46.00)

tracking task for psychomotor speed (execution time in seconds: higher scores represent worse performance); 3) visual search (VS): a digit cancellation task for focused attention (score range 0–50: higher scores represent better performance); 4) symbol digit modalities test (SDMT): a symbol substitution task for processing speed and sustained attention (score range 0–110: higher scores represent better performance); 5) color-word stroop test (Stroop): a response inhibition task for selective attention (execution time in seconds: higher scores represent worse performance); 6) immediate copy of the Rey-Osterrieth complex figure (ROCF): a constructional praxis task whose complexity requires planning and organizational strategies related to executive functions (score range 0–36: higher scores represent better performance).

MRI acquisitions

All MR images were acquired using a 1.5 T scanner (Intera, Philips Medical Systems, Best, The Netherlands) with 33 mT/m gradients capability and a head coil with SENSE technology. T_1-

weighted images were collected using a turbo gradient echo sequence [repetition time (TR) = 8.1 ms, echo time (TE) = 3.7 ms, inversion time (TI) = 764 ms, flip angle = 8^o, field of view (FOV) = 256 mm × 256 mm, number of contiguous slices = 160, acquisition matrix = 256 × 256 with a slice thickness of 1 mm]. T_2-weighted axial FLAIR images were acquired using TR = 11, 000 ms, TE = 140 ms, TI = 2, 800 ms, flip angle = 90^o, FOV = 250 mm × 250 mm, acquisition matrix = 280 × 202, number of slices = 40, slice thickness = 3 mm, interslice gap = 0.6 mm. In addition, axial diffusion-weighted images were obtained with a single-shot echo-planar imaging sequence (TR = 9394 ms, TE = 89 ms, FOV = 256 mm × 256 mm, matrix size = 128 × 128, 50 slices, slice thickness = 3 mm, no gap, number of excitations (NEX) = 3, SENSE acceleration factor = 2) using diffusion sensitizing gradients applied along 15 non-collinear directions using b values of 0 (b_0 image) and 1000 s/mm^2.

MRI image processing

An overview of the imaging processing and computational approach is schematized in Figure 5.1.

Diffusion-weighted images (DWI) were corrected for head motion and eddy current distortions using the FMRIB's Diffusion Toolbox part of FSL 5.0.9 (Smith et al., 2004), and the rotational part of the affine transformation employed in this step was applied to the b-matrix (Leemans and Jones, 2009). Brain tissue was segmented using BET (Smith, 2002). Through a constrained nonlinear least-squares procedure implemented in the software CAMINO (Cook et al., 2006), a tensor model was later fitted to the DWI data. The diffusion tensor invariants of FA and MD indices were extracted from the tensor model using the package DTI-TK (Zhang et al., 2007).

Registration to the high-resolution MNI152 standard space was performed on the multi-modal MR images (T_1-weighted, T_2-weighted FLAIR, and DWI images) and derivatives (FA and MD maps). In detail, the T_1-weighted scans have been aligned to the standard template space through an affine and deformable transformation with linear interpolation included in the ANTs package (version 2.1.0) using default parameters (Avants et al., 2011). Later, the same affine and deformable transformation was applied to T_2-weighted FLAIR images and FA and MD maps, previously registered to the respective T_1-weighted scans by using a 12 degrees-of-freedom

linear transformation with spline interpolation implemented in the FSL *flirt* tool (Greve and Fischl, 2009; Jenkinson et al., 2002; Jenkinson and Smith, 2001).

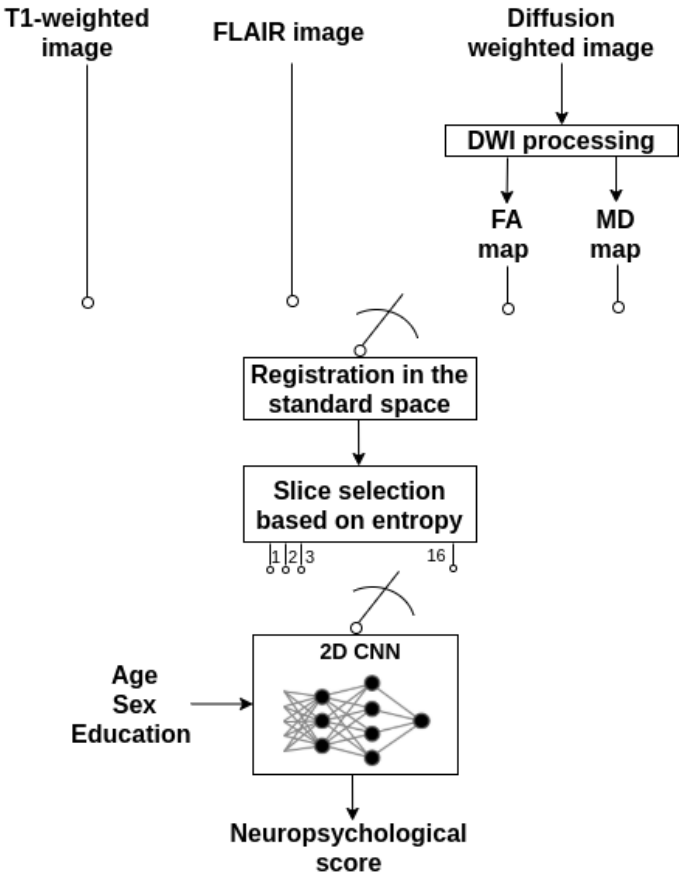


Figure 5.1: General overview of our method: each MRI data (T1-weighted, FLAIR, MD and FA) has passed through a preprocessing step. Then the adopted VGG16 model is trained on the training samples [MRI data and demographic variables (age, sex and years of education)] and the trained CNN is used to make a prediction of raw cognitive scores (MoCA, SDMT, TMT-A, ROC-F immediate copy, Stroop and visual search). Abbreviations: CNN, convolutionaal neural network; DWI, diffusion-weighted image; FA, fractional anaisotropy; MD, mean diffusivity)

A Dell PowerEdge T620 workstation equipped with two 8-core Intel Xeon E5–2640 v2, for a total of 32 CPU threads, and 128 GB RAM, using the Oracle Grid Engine batch-queuing system for parallel computing was used to perform all image processing tasks.

Since our proposed deep learning model is based on a 2D CNN architecture, we sliced the 3D MRI volumes in to a 2D gray scale images using an axial plane. However, not all axial slices contain important information for training the model for the prediction problem. Hence, we have

performed a preliminary slice selection based on the amount of information, retaining, for each volume, only a limited number of axial slices that showed the highest entropy(Hon and Khan, 2017). Mathematically, for a slice with k grayscale levels and with each gray level having a probability of occurrence p_k (estimated as its relative frequency in the image), the Shannon entropy E_S was computed as:

$$E_S = \sum_k p_k \log_2(p_k) \quad (5.1)$$

To decide the number of slices to choose, we selected a reference model that is trained on T1-weighted images to predict a MoCA score and this model is trained iteratively on different number of axial slices $\{1, 2, 3, \dots, 16\}$ based on a 10-fold CV approach. Then, we compared the results of the models trained on each number of slices and the model that produced the best performance on the test set was chosen. In this study thus, 16 most informative slices were considered. For all the other models we considered this number.

Considering that the pre-trained CNN model has been trained on color (3 channels) natural images, we repeated our grayscale images onto the three channels. Finally, voxel-wise feature standardization (for each voxel, an average value of all grayscale values within the brain mask has been subtracted and scaled by the standard deviation (within the brain mask)) has also been applied to make training the CNNs easier and to achieve faster convergence.

CNN models

CNN architecture

Since our dataset size is relatively small, we employed transfer learning technique by which knowledge gained by training a model in the source domain (i.e., a large dataset) can be transferred to the target domain (i.e., a small size dataset) to improve the performance of the model on the target task (Tan et al., 2018). Specifically, we have adapted the VGG16 architecture to build a multi-input CNN. VGG16 is one of the pre-trained 2D CNN models that won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), where deep CNNs are challenged to produce the best performance in classifying natural images on the ImageNet dataset. Our proposed multi-input CNN consists of two input branches, the CNN branch and the fully connected (FC) branch taking the image and demographic data, respectively. In the CNN branch, we reused the convolutional layers of VGG16 by replacing its pre-trained FC layers with

a GlobalAveragePooling (GAP) layer. While the FC branch, consists of a number of FC layers. The outputs from the two branches are then concatenated and the final FC layers produce a prediction output. The detailed architecture of our proposed architecture is shown in Figure 5.2.

Model training and evaluation

During the proposed 2D CNN training, all transferred convolutional layers were fine-tuned, and the newly added layers were trained from scratch. The training was performed using the Adam optimizer (Kingma and Ba, 2017) – an effective adaptive stochastic gradient descent optimization algorithm that adaptively updates individual learning rates for each parameter – using the first and second moments of the gradients to minimize the loss function and to get the best estimates of network parameters. Mean squared error (MSE) was used as a loss function to measure the error between the actual values of cognitive scores and the values predicted by the model.

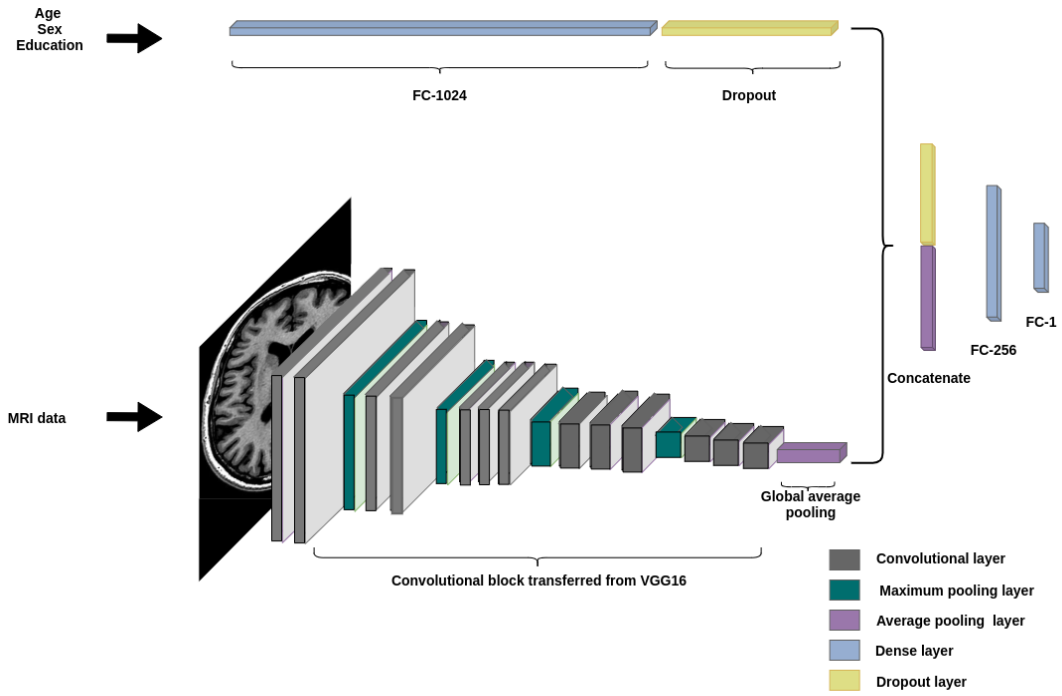


Figure 5.2: The adapted multi-input VGG16 model. Brain image data is processed by the convolutional blocks and demographic data is fed to the densely connected layers. The features are then concatenated and analyzed by the last fully connected layers (FC-256 and FC-1). Abbreviations: FC, fully connected; VGG, visual geometry group.

In this study, we incorporated two experiments. In the first experiment, to demonstrate the contribution of the demographic features in improving the model's prediction accuracy, two CNN models, one multi-input and another single-input CNN, were trained with and without demographic information respectively for predicting the MoCA score. The two models were trained on a 10-fold CV scheme using Adam optimizer with a learning rate of 1×10^{-4} and a learning rate decay of 0.5 and the model's performances on the validation set were compared to assess the contribution of demographic scores.

In the second experiment, we employed a 10-fold nested CV strategy for training and validating the multi-input CNN to get a prediction of the raw neuropsychological test scores (MoCA, SDMT, TMT-A, ROC-F immediate copy, Stroop and Visual search). In both experiments, our dataset was split into train-validation (CV in experiment 1) and train-validation-test sets (nested CV in experiment 2) on a subject-level basis, i.e., we incorporated all the 16 slices of a single subject in either the training, validation or test sets to prevent building an overly optimistic model caused by the presence of data leakage.

The final trained model's performance was evaluated using a regression metrics average slice-level Pearson correlation coefficient (r_{av}) between the actual values, and the model predicted values of the neuropsychological scores computed across the outer folds.

The deep learning scheme was set up on a workstation equipped with a 12 GB G5X frame buffer NVIDIA TITAN X (Pascal) GPU with 64 GB RAM, 8 CPUs, 3584 CUDA cores, and 11.4 Gbps processing speed. All CNN model development, training, validation and testing of the models have been implemented in Python (version 3.6.8) language using the following modules: GPU-TensorFlow 1.12.0, Keras 2.2.4 (Tensorflow backend), Sklearn 0.20.2, Nibabel 2.3.3, Hyperas 0.4.1, Hyperopt 0.2.2. The computational time needed for the training and validation of each CNN model was about 75 hours.

2. Results

In the first experiment, where a raw MoCA score is predicted using multiple MRI types, incorporating demographic features significantly improves the performance of all CNN models. While the models that are trained on image data only produced r=0.1463 using T_1-weighted images, r=0.2638 using MD, r=0.1061 using FA and r=0.2959 using T_2-weighted FLAIR images, including demographic data boostes the model's performance by producing (r=0.5120 on T_1-weighted images, r=0.6033 on MD, r=0.4269 on FA and r=0.5228 on T_2-weighted FLAIR images. Figure 3.2 demonstrates the comparison of the learning curves of the models trained with and without demographic data.

Table 5.2 also lists the performance obtained by the proposed CNN models for the correlation of the neuropsychological scores with different MRI features. Different neuropsychological scores were significantly predicted by different MRI features maps. The Montreal Cognitive Assessment (Pearson's correlation coefficient r=0.523), visual search (r=0.368), and Rey-Osterrieth complex figure (r=0.224) were best estimated using T_1-weighted, the symbol digit modalities test using T_2-weighted FLAIR (r=0.569) images, and the trail making test part-A (r=0.513) and the Stroop (r=0.460) using MD maps.

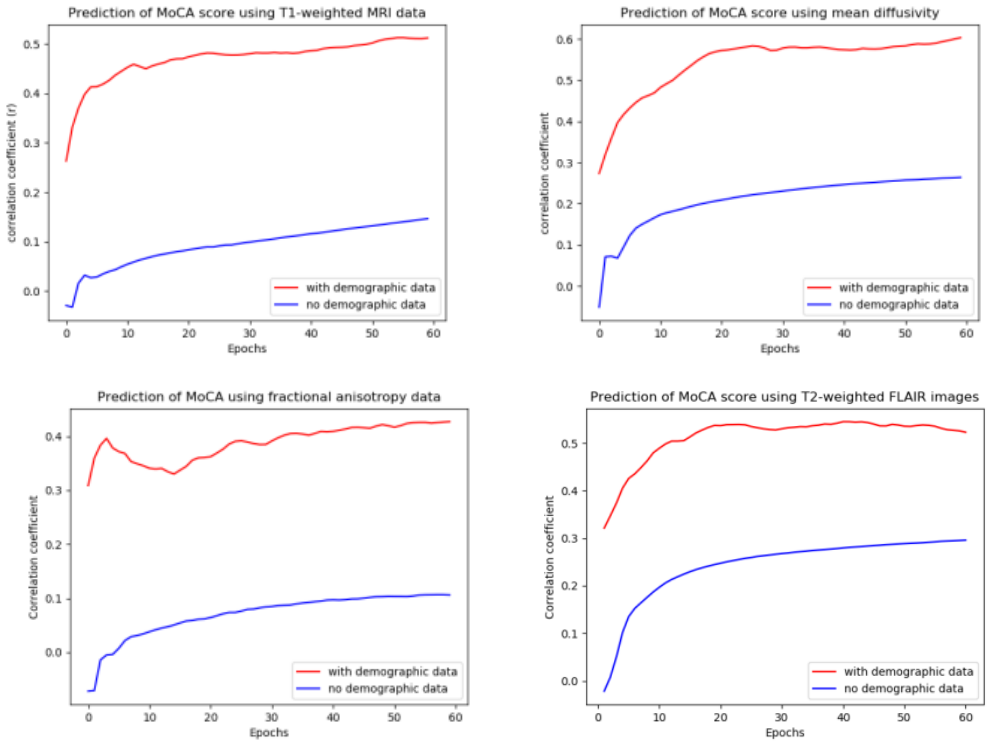


Figure 5.3 Comparison of a MoCA score prediction on the test set with and without incorporating demographic variables. For all MRI types, including demographic data significantly improves the prediction accuracy of the CNN model.

Table 5.2: Average Pearson's correlation coefficient over 10-fold nested CV on outer fold test samples.

Cognitive test	Mean correlation coefficient			
	T_1	FLAIR	MD	FA
MoCA	0.523	0.435	0.504	0.375
SDMT	0.505	0.569	0.246	0.505
TMT-A	0.438	0.451	0.522	0.504
ROC-F immediate copy	0.224	0.203	0.145	0.207
Stroop	0.419	0.429	0.460	0.412
Visual search	0.368	0.166	0.011	0.117

5.2 3D Convolutional Neural Networks for Diagnosis of Alzheimer's Disease via structural MRI

5.2.1 Introduction

Alzheimer's Disease (AD) is the most common type of dementia, which is caused by the deterioration of cognitive and memory functions(Hague et al., 2005). Pathologically, AD is characterized by the accumulation of extracellular β-amyloid (Aβ) plaques and cytopla00000mic neurofibrillary tangles (NFTs) which have a microtubule-associated protein called tau(Braak and Braak, 1991). In healthy neurons, tau protein normally stabilizes the microtubules(Weingarten et al., 1975). However, abnormal changes in brain chemistry cause tau protein molecules to detach from microtubules and form neurofibrillary tangles destroying the brain cells' ability to communicate with other cells(Grundke-Iqbal et al., 1986). Some recent studies reveal that AD may begin 20 years or more before any symptoms appear and the disease is clinically diagnosed(Villemagne et al., 2013, Reiman et al., 2012, Jack Jr et al., 2009, Bateman et al., 2012, Braak et al., 2011). Only after a certain stage, patients may experience diagnostic symptoms such as deterioration in memory and decline in cognitive abilities when irreversible neurological damage already occured. Therefore, an early and accurate diagnosis of AD is crucial and may be possible via computer-assisted analytical techniques. Receiving an early diagnosis of AD will enable patients to benefit from various treatments, plan their future, and maximize their life quality. As AD progresses, the structure of the brain undergoes some changes, such as the shrinkage of the cerebral cortex and hippocampus and the expansion of ventricles(Lehericy et al., 1994, Bobinski et al., 1999). Through numerous medical imaging techniques like magnetic resonance imaging (MRI), positron emission tomography (PET) and computed tomography (CT), some of these changes can be detected earlier. Notably, a T1-weighted MRI scan of the brain reveals high-resolution structural information of the brain and can be used to identify atrophic changes in the temporal lobes(Mortimer et al., 2004).

Throughout the last decade, multiple studies have been focusing on the automatic diagnosis of AD using different methods(Alam et al., 2017, Liu et al., 2013, Gray et al., 2013). Among those, deep learning (DL) has come to the fore as one of the most promising tools to address AD diagnosis and prognosis. In DL models, discriminative features may be extracted automatically from the raw data resulting in end-to-end learning design.

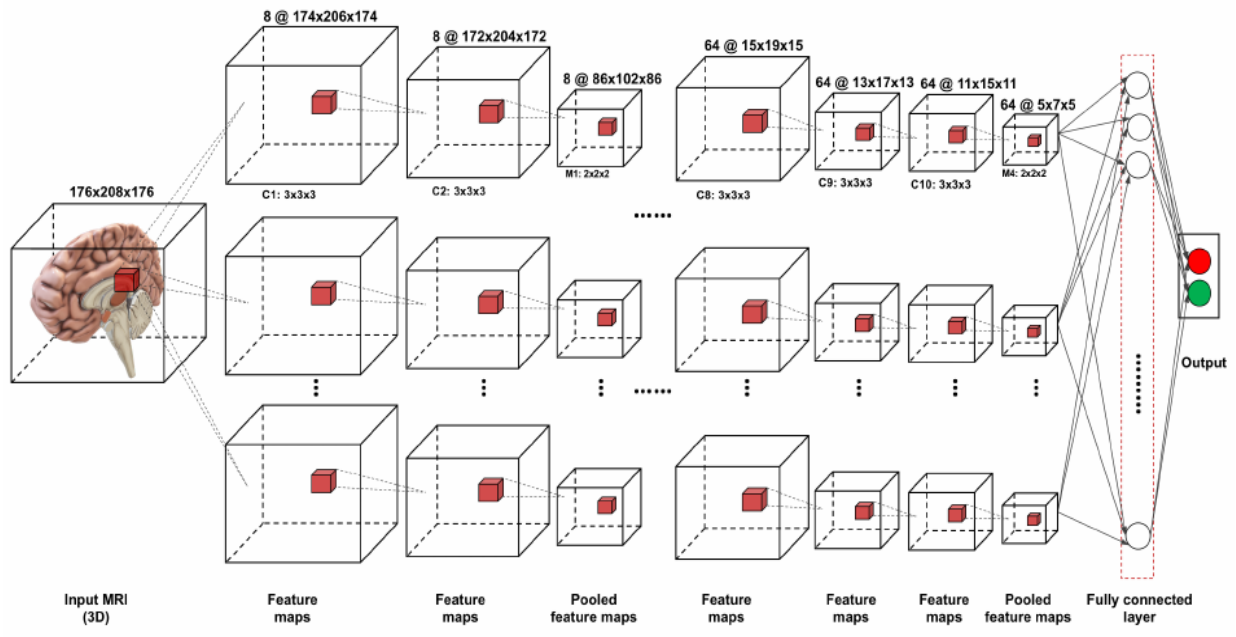


Figure 5.4: Overview of the 3D convolutional neural network (CNN) architecture. 3D boxes show input and feature maps.

In this work, we propose an end-to-end AD classifier, which takes T1 weighted MRI as input. We implemented a 3D VGG (a deep neural network model implemented by Oxford Visual Geometry Group (VGG)) variant convolutional neural network (CNN) to overcome the limitations regarding the feature extraction from brain MRI and preserve spatial relations. Figure 5.3 provides an illustration of the network architecture.

The paper is organized as follows: after this introduction, a brief of related work is given in Section 5.2.2. Section 5.2.3 provides the details of the proposed model, including the dataset and classification algorithm of CNN. Experimental results are presented in Section 5.2.4. Finally, Section 5.2.5 concludes the paper with some final remarks.

5.1.2 Related Work

DL has become a popular and powerful technique with the advance of the computational power of GPU clusters and big data analytics, and as a result, rapidly expanded into various fields. In medical image analysis, several neuroimaging studies have utilized DL models for diagnosis of

AD(Huang et al., 2019, Oh et al., 2019, Payan and Montana, 2015, Rieke et al., 2018, Korolev et al., 2017, Wen et al., 2020).

Various studies used a set of 2D slices extracted from the MRI volume as input to the 2D CNN architectures(Farooq et al., 2017, Gunawardena et al., 2017, Hon and Khan, 2017, Islam and Zhang, 2018, Valliani and Soni, 2017, Wang et al., 2018, Yagis et al., 2019). Farooq et al. (Farooq et al., 2017)used a 2D CNN model for 4-way classification of Alzheimer's into AD, MCI (Mild Cognitive Impairment), LMCI (Late Cognitive Impairment) and HC (Healthy Control) using structural MRI images. Sarraf et al. (Sarraf et al., 2016)utilized CNN and the famous architecture LeNet-5 to classify functional MRI data of AD's patients from healthy controls. In [24], Hon et al. used VGG16 and Inception V4 to classify AD using transfer learning. Finally, in 2019, Jain et al. (Jain et al., 2019)presented the CNN model for the 3-way AD classification. However, in most of these studies, it is not clear if data division was done at the subject-level, calling into question the validity of the results due to potential data leakage(Wen et al., 2020, Yagis et al., 2019, Fung et al., 2019). Another possible problem in the 2D approach is the loss of information from 3D MRI when sliced and analyzed by 2D convolutional filters.

Some studies addressed 3D networks to solve the issue of insufficient information in the 2D slice-level approach(Huang et al., 2019, Oh et al., 2019). Even though these models are computationally more expensive, they have a higher capability to extract discriminative features from three-dimensional MRI data. Korolev et al. (Korolev et al., 2017)used 3D residual neural network architecture together with several regularization techniques for AD classification. In 2018, Hosseini-Asl et al.(Hosseini-Asl et al., 2016a), utilized a pre-trained 3D- Adaptive CNN classifier with used scans from the CADDe-mentia dataset for the classification of AD vs. HC. However, the details regarding CV methodology and classification decisions are not presented in this study. Wang et al. (Wang et al., 2017b)proposed an ensemble of 3D densely connected convolutional networks (3D-DenseNets) for three-class AD, MCI, and HC diagnosis. In their model, MRI scans of the same patients that are over three years apart are employed as different samples, incorporating information from test data into the learning process. Rieke et al. (Rieke et al., 2018)trained a 3D CNN for AD classification accuracy. At the end of their visualization efforts, they showed that the model focuses on the medial temporal lobe. Yang

et al. (Yang et al., 2018)also provided visual explanations regarding the AD from deep 3D CNNs. They utilized 3D VGGNet together with 3D-ResNet. Finally, in 2019, Oh et al. (Oh et al., 2019)developed a volumetric CNN-based approach for the AD classification task It should be noted that the classification performances of these studies are hard to compare as they have trained and tested the models with different sets of participants. The studies also differ in terms of the pre-processing stages, hyperparameter selection, cross-validation (CV) procedure, and evaluation metrics.

5.1.3 Methodology

In this section, the main components of our framework are presented. We briefly describe the datasets used in the experiments in Subsection 5.2.3-A, explain the preprocessing steps of T1-weighted MRI data in 3.2.3-B and finally show the architectures of the model in 3.2.3-C.

A. Datasets

In this study, we use two primary publicly available datasets on AD and Related Dementia: the Alzheimer's Disease Neu- roimaging Initiative (ADNI) dataset (Petersen et al., 2010) and Open Access Series of Imaging Studies (OASIS)(Marcus et al., 2007). These datasets are described in detail below. The characteristics of the subjects included in this study are given in Table 3.4 in section 3.2.1

1) ADNI: ADNI1 is a research initiative that brings together researchers to collect, validate, and utilize several types of data such as clinical, genetic, MRI, PET, and biospecimen to validate biomarkers for AD(Petersen et al., 2010). ADNI was formed in 2004 and launched three different phases so far, namely ADNI 1, ADNI GO/2, and now ADNI 3. In addition to the first phase, ADNI 2 contains information from 150 elderly controls, 100 EMCI subjects, 150 late mild cognitive impairment (LMCI) subjects, and 150 mild AD patients.
In this work, we used a subset of ADNI 2 dataset with 200 structural T1-weighted MRI scans. From ADNI 2 dataset, we randomly picked 200 subjects, 100 of whom were chosen from the AD group (44 women and 56 men, age 74.28 ± 7.96 years, mean ± SD), while the other 100 from the HC group (52 women and 48 men, age 75.04 ± 7.11 years, mean ± SD). Only the first scan of each patient has been added to the dataset.
Patients with a CDR score of 0 are labeled as HC subjects, whereas the ones whose CDR rating

is higher than 0 are considered AD subjects. MPRAGE T1-weighted MRI images have been acquired using 3 T scanners, and consisted of 176 × 240 × 256 (Siemens) and 170 × 256 × 256 (Philips) voxels with a size of approximately 1 mm × 1 mm × 1.2 mm.
2) OASIS: OASIS2 is a project that is intended to promote future discoveries in AD by providing neuroimaging datasets freely to the scientific community. The project released data in three different phases: OASIS 1-Cross-sectional, OASIS 2- Longitudinal, and OASIS-3-Longitudinal. OASIS 1 includes overall 416 subjects (316 HC and 100 AD) aged 18 to 96. For our experiments, T1-weighted MRI scans of 100 healthy subjects [73 women and 27 men, age 75.5 ± 9.1 years, mean ± SD] and 100 AD patients (59 women and 41 men, age 76.7 ± 7.1 years, mean ± SD) have been selected to create a subset of OASIS-1 dataset. Again, the CDR score was 0 for the HC subjects, 0.5 (very mild), 1 (mild), 2 (moderate), and 3 (severe) were for the AD subjects. MPRAGE T1-weighted MRI images have been acquired using a 1.5 T Siemens scanner. They are in the size of 256 × 256 × 128 with voxel size 1 mm × 1 mm × 1.25 mm.

B. Data pre-processing

Even though CNN models do not require any preprocessing beforehand, an accurate image preprocessing stage could be key to increasing the effectiveness of learning and help to achieve a good classification performance, particularly in the domain of MRI(Cuingnet et al., 2011, Lu and Weng, 2007). We transformed all the data into a standardized structure by performing co-registration with a standard template and skull stripping. For ADNI, each T1-weighted image has been co-registered with the SyN method using standard T1-weighted template MNI152 at 1 mm(Avants et al., 2011). After co-registration, the brain mask of the standard space was applied to each volume to remove extracranial tissues. The final size of the ADNI T1-weighted MRI volumes is 182 x 218 x 182 with 1mm x 1mm x 1mm voxel size.
When it comes to the OASIS dataset, we used the data which was already gain-field corrected. An additional brain masking and re-sampling operations are performed. The final dimension of the 3D volume is 176 x 208 x 176 with 1mm x 1mm x 1mm voxel size(Han et al., 2018b). The sample MRI slices from ADNI and OASIS datasets after the pre-processing stage can be seen in Figure 5.4.

C. CNN Models: 3D Convolutional Networks

We created a 3D CNN model inspired by VGG-16 architecture. The model has four onvolutional

blocks, among which the first two contain two convolutional layers each, and the latter two have three convolutional layers followed by a pooling layer with filter size a 2x2x2. The overview of

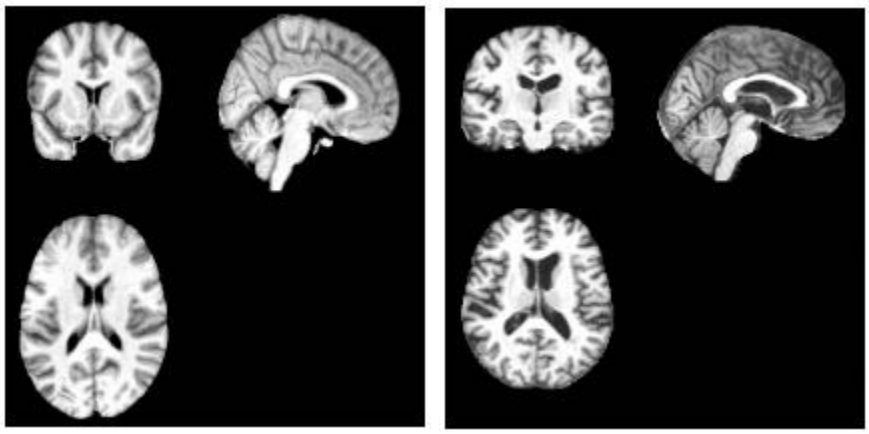


Figure 5.5: Example of six Magnetic resonance imaging (MRI) slices of two Alzheimer's Disease (AD) subjects from ADNI and OASIS databases (Petersen, et al., 2010; Marcus, et al., 2007). a) A sample T_1-weighted MRI slices of an Alzheimer's disease (AD) patient from ADNI dataset after pre-processing – in coronal, sagittal, and axial view (left, right and bottom respectively). b) Sample of T_1-weighted MRI slices of an Alzheimer's disease patient from the OASIS dataset after pre-processing processing – in coronal, sagittal, and axial view (left, right and bottom respectively)

the 3D CNN architecture is shown in Figure 5.5. A convolutional and a pooling layer has several feature maps, and in most cases, the number of feature maps increases as layers grow. The calculation of the jth feature map is given by:

$$y^j = f\left(w_j * x + b_j\right) \tag{5.2}$$

where yj be the 3D array of the jth feature map in a hidden layer, x be the 3D array of the input, bj be the scalar bias and W_j be the 3D filter with a size of w×h×d. f corresponds to an activation function, and $*$ stands for the convolution operation. The convolution operation [Wj $*$ x](m, p, q), is represented as follows:

$$\sum_{u=0}^{w-1}\sum_{v=0}^{h-1}\sum_{k=0}^{d-1} W_j(w-u, h-v, d-w)x(m+u, p+v, q+k) \tag{5.3}$$

After the convolutional blocks, a dropout layer with a probability of 0.5 is applied to avoid overfitting. It is followed by three fully connected layers with 128, 64, and 2 neurons,

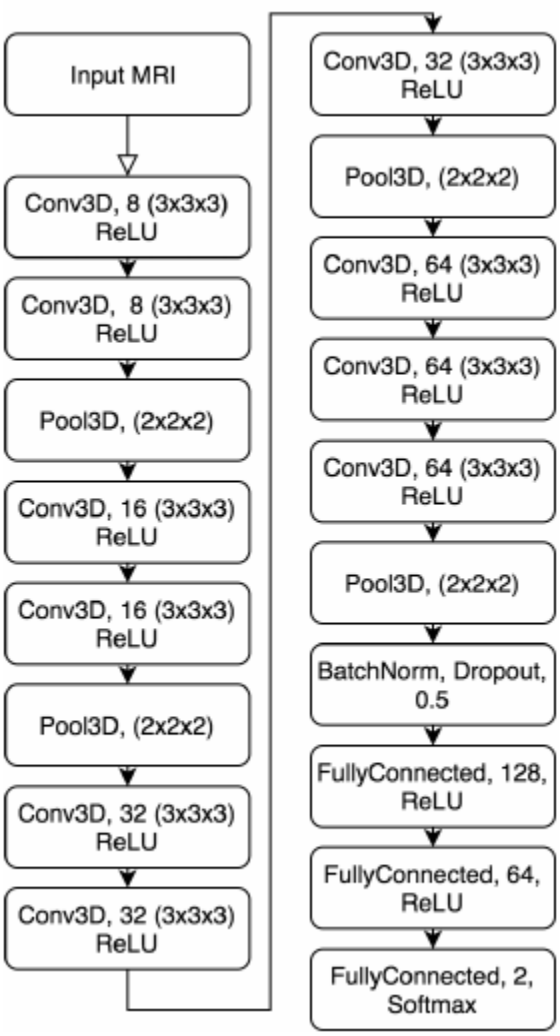


Figure 5.6: The architecture of the convolutional neural network (CNN) model used in our AD classification tasks.

respectively. The last fully-connected layer with softmax activation provides the output label. The model has been trained with categorical cross-entropy loss and the Adam optimizer with a learning rate of 0.0001 and a batch size of 2 for 200 epochs. Binary cross-entropy loss is computed as:

$$L(y,p) = -(y log\, p + (1-y) log(1-p)) \quad (5.4)$$

where y is the actual label and p is the predicted label. Training and validation of our proposed models were performed on a NVidia RTX2080 GPU.

5.1.4 Results

The model has been evaluated using five-fold CV. The average accuracy is obtained by repeating 5 times the full 5-fold cross-validation starting from five different splits of the data into folds. The architecture was built using Keras with TensorFlow backend(Chollet and others, 2015, 2016).

The model was tested on two different test sets, each of which contained 40 subjects. Using 5-fold CV, the model achieves 73.4%± 0.04 (mean, standard deviation) on ADNI dataset and 69.9%± 0.06 (mean, standard deviation) classification accuracy on the OASIS dataset. The results are comparable to other studies that use different convolutional models for AD vs. HC classification. In addition, the dataset is divided by subjects, and only one screening of a patient is included in the dataset in order to prevent possible data leakage. For instance, Rieke et al. (Rieke et al., 2018)reported 78% ± 0.04 classification accuracy with a similar architecture using ADNI 1 datasets, which contains MRI scans of the subjects up to three-time points (screening, 12 and 24 months; sometimes multiple scans per visit). Following such procedure may cause the scans of the same subject to be in both testing and training set, which could affect the model performance.

5.2 Development of an interpretable deep learning system for the classification of Alzheimer's disease

5.3.1 Introduction

Alzheimer disease is the most commonly occurring neurodegenerative disorder (Selkoe and Lansbury, 1999) that causes memory impairment at its initial stage and advances to a cognitive decline that can affect behavior, speech, visuospatial orientation, and motor system(Kelley and Petersen, 2007). Early diagnosis is important to plan treatment strategies that could slow down the disease progression and enhance the quality of life(Small et al., 1997). Diagnosis of AD needs a follow-up of patient's medical history by a physician by performing clinical assessment and neuropsycological tests(Small et al., 1997). Neuroimaging tools, such as structural MRI, functional MRI, and positron emission tomography (PET) are also used to confirm that the cognitive decline caused by AD is altering the brain structure.

In the past, traditional machine learning methods have been used to analyze neuroimaging data. Nevertheless, due to the need to extract hand-crafted features, designing a machine learning system becomes a very long process and is not appropriate for non-expert users. Another approach called deep learning, which is a family of machine learning methods that has the ability to automatically extract features from complex data(LeCun et al., 2015), 2015) overcomes the limitations of traditional machine learning approaches and hence becomes the current state of the art technology in medical imaging, including neuroimaging.

Convolutional neural networks (CNNs), are a special type of deep learning models that are used specifically for image processing applications(LeCun et al., 2015). A basic CNN model consists of convolutional layers, pooling layers and fully connected layers. Numerous studies have employed CNNs for classifying between sMRI of AD patients and healthy subjects(Liu et al., 2020, Oh et al., 2019, Qiu et al., 2020, Wen et al., 2020, Feng et al., 2020, Yagis et al., 2019). Most of these studies used MRI data obtained from ADNI dataset and other few studies (Yagis et al., 2019, Yagis et al., 2021, Tufail et al., 2020, Puente-Castro et al., 2020, Saratxaga et al., 2021, Mehmood et al., 2020, Massalimova and Varol, 2021) applied deep learning techniques on the OASIS collection of brain images.

Apart from their success in many applications, deep learning approaches have been criticized for producing highly non-interpretable models(Linardatos et al., 2020). Interpretability is a requirement in many applications in which crucial decisions are made by users relying on a model's outputs, such as in medical applications(Lipton, 2018).

CNN visualization methods help understand the reasoning behind the model's decisions. A number of recent neuroimaging studies have integrated explainability tools in their CNN models to classify different neurological disorders(Gao et al., 2021, Jimeno et al., 2022, Zhang et al., 2021, Qiu et al., 2020, Tang et al., 2019, Lu et al., 2022, Oh et al., 2019, Iizuka et al., 2019, Sánchez Fernández et al., 2020). Regarding AD classification, few studies (Lu et al., 2022, Oh et al., 2019, Qiu et al., 2020) employed CNN visualization techniques to highlight the features used by the model to make decisions. All of these studies used a public brain dataset of AD and healthy individuales, namely Alzheimer's Disease Neuroimaging Initiative (ADNI).

In this study, we proposed an interpretable CNN for classifying structural MRI (sMRI) scans obtained from a public OASIS dataset. The CNN model is trained based on a transfer learning technique by utilizing the weights of a pre-trained VGG16 network. Unlike the previous deep learning studies classifying the OASIS collection of brain images, our proposed model includes a wide range of visualization methods to illustrate that the models are focusing on the clinically defined AD pathologies.

5.3.2 Methods

In this section, the datasets used in our study, the model architecture, training and validation schemes, and finally CNN visualization methods applied to the trained model and their interpretation are discussed.

Subjects

In this study we used a publicly available dataset of AD patients and healthy control (HC) subjects called Open Access Series of Imaging studies (OASIS)(Marcus et al., 2007). The dataset consists of across sectional collection of MRI scans of 416 right-handed subjects aged between 18 and 96. The scans were acquired using a 1.5 T vision scanner. In the dataset both men and women are included. 100 AD patients [(59 women and 41 men, age 76.70 ± 7.10 years, mean ± standard deviation (SD)] and 100 HC subjects (73 women and 27 men, age 75.50 ± 9.10 years,

mean ± SD) who have been previously selected by Hon and Khan (Hon and Khan, 2017) are included in our experiment (refer Section 3.2.1.2 for inclusion criteria and scanner parameters and Table 3.4 for demographic information).

Data pre-processing

The OASIS dataset publicly provides a pre-processed data, where gain-field correction, brain masking and atlas-based co-registration(Han et al., 2018b) were applied to the raw MRI images resulting in a data matrix size of 176 × 208 × 176 and a voxel size of 1 mm × 1 mm × 1 mm(Han et al., 2018b). We performed 2D image processing on such partially pre-processed 3D MRI volumes. The 2D image processing involves slicing the MRI volumes in to 2D gray scale images using an axial anatomical plane, performing slice selection based on entropy values (as explained in Section 3.2.1.2), splitting the data in to training and validation sets based on a 5-fold CV scheme and lastly applying feature scaling based on the training feature statistics (using mean and SD values). From each MRI scan 10 slices are selected based on their entropy values (Hon and Khan, 2017) producing a total of 2000 (1000 AD and 1000 HC) gray scale images. As compared to the number of parameters for building a CNN model, the size of the image dataset is insufficient to effectively train a CNN from scratch. To prevent overfitting of a CNN model caused due to limited training samples, we employed a technique called transfer learning by starting from a pre-trained VGG16 model and finetuning the model parameters on the MRI dataset. Since VGG16 is trained on colored RGB images, the gray scale MRI slices were converted to threechannel images by repeating the 2D image on to the three channels. By applying these pre-processing operations, we end up with an array of 2000 x 176 x 208 x 3.

CNN model

The CNN model architecture is customized from the pre-trained VGG16 model. The fully connected (FC) layers of VGG16 are removed and replaced by a global average pooling (GAP) layer and a last FC classification layer with a ‘sigmoid’ activation is added (Figure 5.6). During model training three convolutional blocks were freezed to reduce the number of trainable parameters and to avoid overfitting. The rest two blocks of convolutional layers were finetuned along with the newly added FC layer.

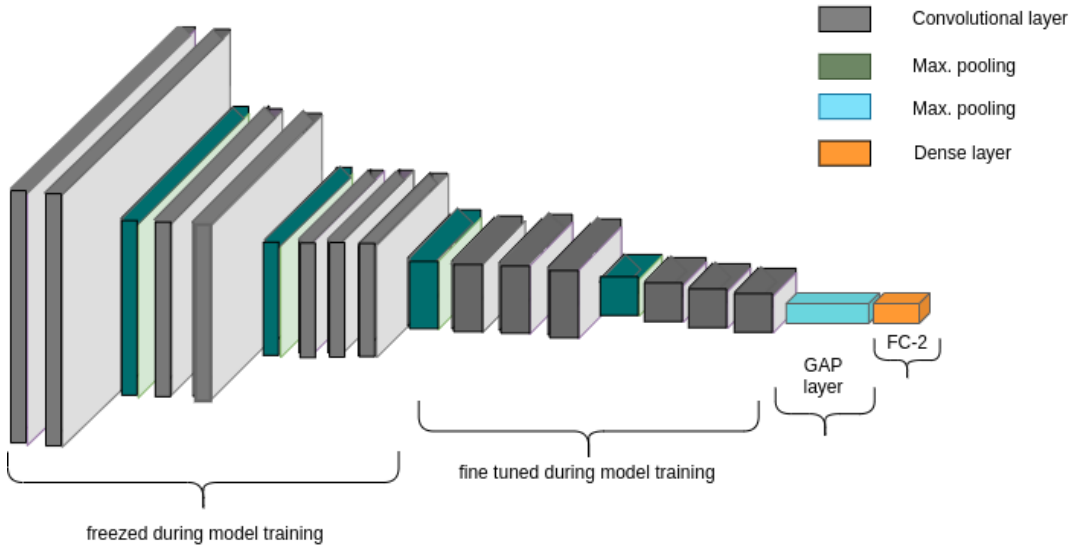


Figure 5.7: A customized VGG16 model consists of: a convolutional that which is transferred from the pre-trained VGG16 model, a GAP (global average pooling layer) and two FC layers (FC-256 and FC-2).

Model training was performed based on a 5-fold CV scheme with an Adam optimizer with a *learning_rate* of 1 x 10^{-4} and a learning rate *decay* of 0.5. '*categorical_corssentropy*' is used as a loss function. The model is trained for 90 epochs and with a batch size of 128 images. Three classification metrics: 1) balanced accuracy, 2) sensitivity and 3) specificity were used to measure the performance of the model. The final results of the trained model are reported based on the average accuracy computed over the 5 folds on the validation set.

CNN visualization

Model visualization methods are tools that enable understanding the rationale behind a deep learning model's decisions. For a CNN model, these interpretability approaches are applied on a trained model to see which image regions or features are given high importance for the prediction analysis. In this study, we employed 4 attributebased interpretability techniques (two gradient-based approaches, saliency maps and GradCAM and two perturbation-based methods, SHAP and Occlusion maps) for a classification problem of AD vs. HC subjects. To emphasize the importance of these visualization tools, we performed two experiments. In the first

experiment, a model is trained to classify subjects as AD and HC, and visualization heatmaps highlight the brain regions that are used by the model to identify AD brain scans from healthy MRI images and to check if our results are in line with the neural correlates of AD, which are defined in the previous AD studies.

While, the aim of the second experiment is to highlight the potential of these visualization tools for identifying biased models producing highly inflated performances. Data-leakage caused by slice-level split is one of the methodological pitfalls of applying 2D CNNs for the classification of 3D MRI data that result in a biased model outputting overestimated performance on the test set(Yagis et al., 2021, Yagis et al., 2019, Wen et al., 2020). Following similar procedures as in the experiment explained in Section 3.2.1, in this study also we trained two architecturally similar models using two data split methods. While the first model is trained by applying subject-level split, hence without data leakage, the second model is trained on data that is divided based on MRI slices introducing data leakage. Correctly classified AD test samples are then passed through the trained models and visualization heatmaps generated from the two models are compared to check if reliabile features are used by the two CNN models.

5.3.3 Results

The results of Experiment 1 and Experiment 2 are presented in this section. In Experiment 1, the performance of our interpretable CNN model as measured by the average accuracy, sensitivity and specificity values computed over the five folds on the test set are reported in table 5.3. The learning curve is also shown in Figure 5.7. An example of the visualization hetmap images generated by passing MRI images of AD patients which are predicted by the model taken from the test set can be seen in Figure 5.8.

In the second Experiment, the model which is trained introducing data leakage achieved a test set accuracy of 95.12% (Table 5.4). Figure 5.8 and 5.10 illustrate the learning curve and the visualization heatmaps of the trained model respectively.

Table 5.3: Average model's performance computed over the five folds on the test set.

	sensitivity	specificity	accuracy
training set	0.8146	0.7686	0.7916
test set	0.7185	0.7273	0.7162

Table 5.4: Average accuracy computed over the five folds on the validation set.

	sensitivity	specificity	accuracy
training set	0.9996	0.9810	0.9923
test set	0.9592	0.9450	0.9512

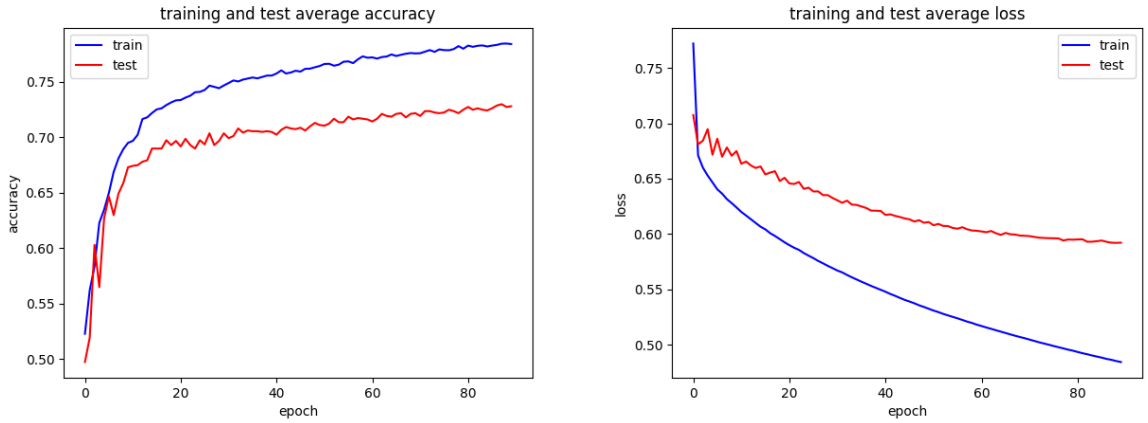


Figure 5.8: Learning curves of the model on both the training and validation samples.

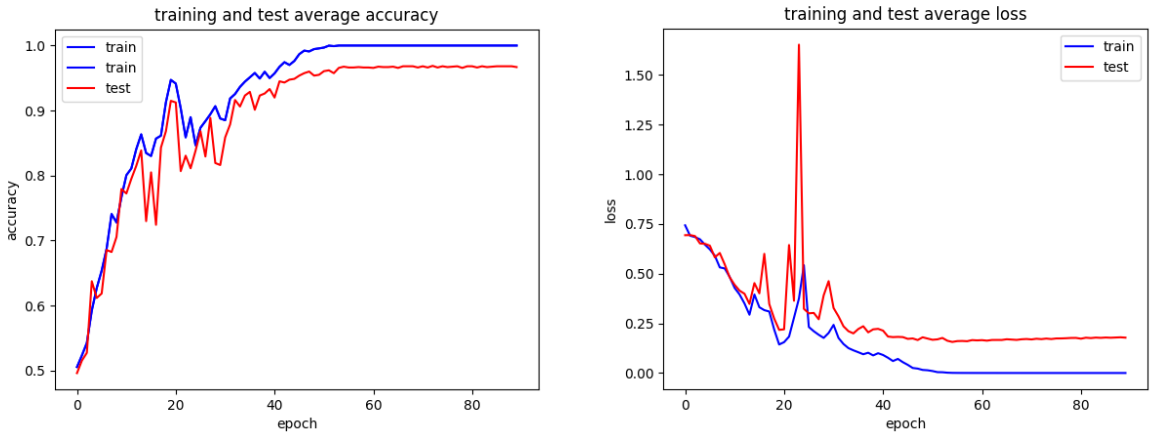


Figure 5.9: the learning curve of the biased model trained with data leakage.

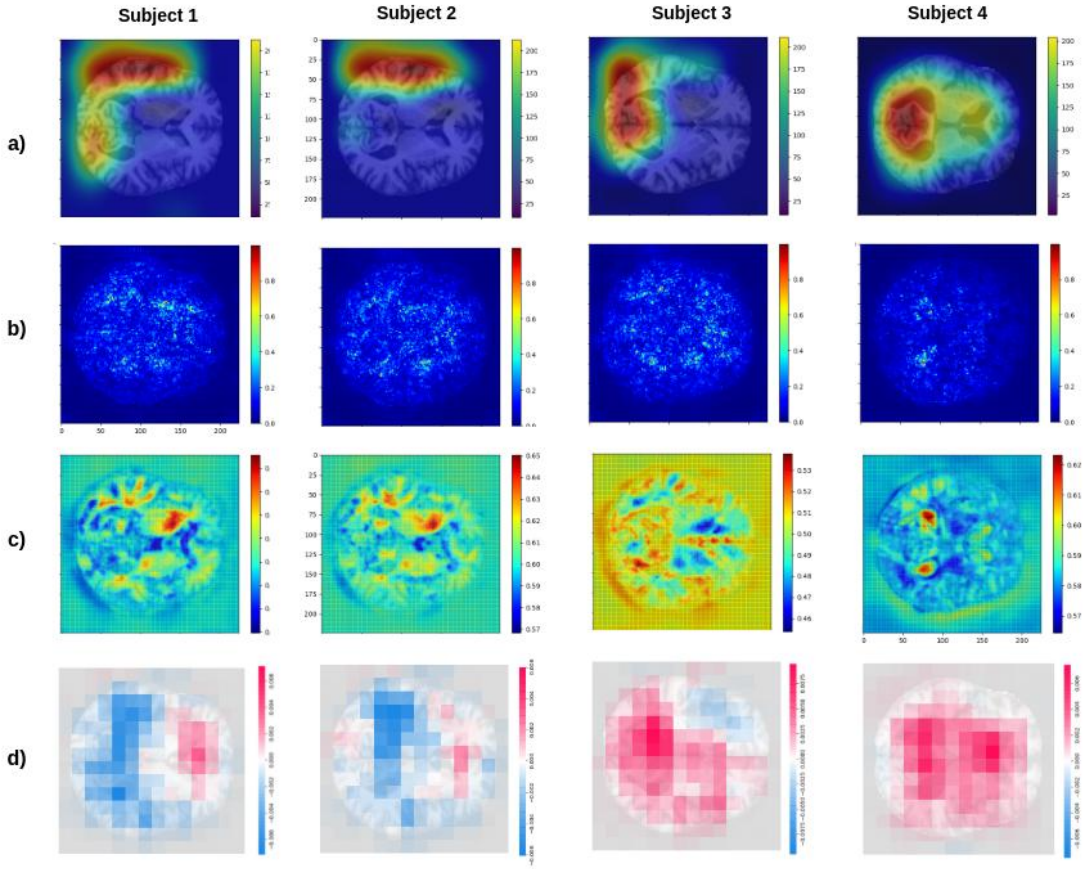


Figure 5.10:CNN visualization heatmaps of MRI slices taken from AD patients, which the CNN model correctly classifies. a) represents Grad-CAM images, b) saliency maps, c)occlusion maps and d) SHAP heatmaps.

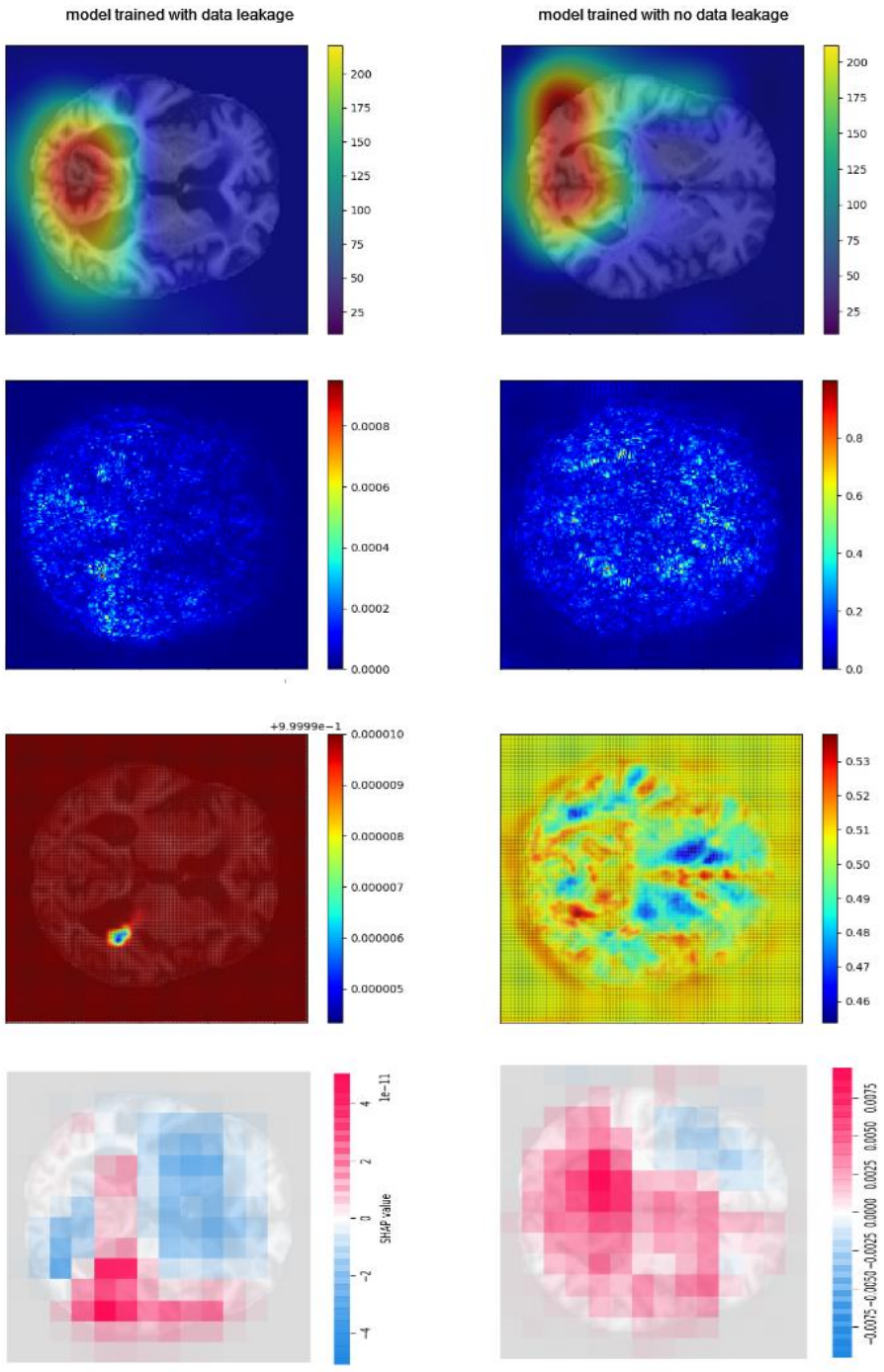


Figure 5.11: CNN visualization heatmaps give an indication of a model producing a biased performance due to the presence of data leakage. Heatmaps on the left side are generated by the model which is trained on data split based on slices (with data leakage). For CAM, occlusion map and SHAP, the heatmap represents a very low number (probability close to 0), capturing the biased model. While, Grad-CAM fails to identify the biased model.

Chapter 6

6. Discussion

6.1 Effect of data leakage in brain MRI classification using 2D convolutional neural networks

In this study, we quantitatively assessed the extent of the overestimation of the model's classification performance caused by an incorrect slice-level CV, which is unfortunately adopted in neuroimaging literature (see Tables 3.1, 3.2, 3.2. More specifically, we showed the performance of three 2D CNN models (two VGG variants and one ResNet-18, see section 3.2.1.2 of chapter 3) trained with subject-level and slice-level CV data splits to classify AD and PD patients from healthy controls using T_1-weighted brain MRI data. Our results revealed that pooling slices of MRI volumes for all subjects and then dividing randomly into training and test set leads to significantly inflated accuracies (in some cases from barely above chance level to about 99%). In particular, slice-level CV erroneously increased the average slice level accuracy on the test set by 40–55% on smaller datasets (OASIS-34 and Versilia) and 25–45% on larger datasets (OASIS-200, ADNI, PPMI). Moreover, we also conducted an additional experiment in which all the labels of the subjects were fully randomized (OASIS-random dataset). Even under such circumstances, using the slice-level split, we achieved an erroneous 95% classification accuracy on the test set with all models, whereas we found 50% accuracy using a subject-level data split, as expected from a randomized experiment. This large (and erroneous) increase in performance could be due to the high intra-subject correlation among T_1-weighted slices, resulting in a similar information content present in slices of the same subject(Murad et al., 2020).

In AD classification, three previous studies(Hon and Khan, 2017, Sarraf et al., 2016, Farooq et al., 2017), using similar deep networks (VGG16, ResNet-18 and LeNet-5, respectively), reported higher classification accuracies (92.3%, 98.0% and 96.8%, respectively) than ours. However, there is a strong indication that these performances are massively overestimated due to a slice-level split. In particular, in one of these works(Hon and Khan, 2017), the presence of data

leakage was further corroborated by the source code accompanying the paper and confirmed by our data. In fact, when we used the same dataset of Hon and Khan(Hon and Khan, 2017) (OASIS-200 dataset), our VGG16 models achieved only 66% classification accuracy with subject-level split, whereas they boosted to about 97% with a slice-level split. Similar findings were presented by Wen et al.(Wen et al., 2020), who used an ADNI dataset with 330 healthy controls and 336 AD patients. Indeed, using baseline data, they reported a 79% of balanced accuracy in the validation set with a subject-level split which increased up to 100% with a slice-level split.

One of the main issues in the classification of neurological disorders using deep learning is data scarcity(Suk et al., 2015). Not only because labeling is expensive but also because privacy reasons and institutional policies make acquiring and sharing large sets of labeled imaging data even more challenging(Kobayashi et al., 2018). To show the impact of data size on model performance, we created 10 small subsets from the OASIS dataset (OASIS-34 datasets). As expected, when we reduced the data, we obtained lower classification accuracies with all the networks using the subject-level data split method. However, when the slice-level method was used, the models erroneous achieved better results on OASIS-34 than on the OASIS-200 dataset. Similarly, models trained on the Versilia dataset (34 subjects) produced inflated results with the slice-level split. Overall, these results indicate that data leakage is highly relevant, especially when small datasets are used, which may, unfortunately, be common in clinical practice.

It is well-known that data leakage leads to inflating performance—and this phenomenon is not specific to brain MRI or deep learning, but it can occur in any machine learning system. Nevertheless, the degree of overestimation quantified through our experiments was surprising. Unfortunately, in the literature, the precise application of CV is frequently not well-documented, and the source code is not available, although we have observed these issues mostly in manuscripts that were either not peer-reviewed or not rigorously peer-reviewed (see Tables 3.1, 3.2, 3.3). Overall, this situation leaves the neuroimaging community unable to trust the (sometimes) promising results published. Regardless of the network architecture, the number of subjects, and the level of complexity of the classification problem, all experiments that applied slice-level CV yielded very high classification accuracies on the test set as a result of incorporating different slices of the same subject in both the training and test sets. Considering

classifications on 2D MRI images, we showed that it is crucial that the CV split be done based on the subject-level to prevent data leakage and get trustable results. This assures that the training and validation sets to be completely independent and confirms that no information is leaking from the test set into the training set during the development of the model. Additionally, employing 3D models for 3D data with subject-level train-test split should be encouraged as 2D models do not effectively capture 3D features. The high computational complexity of 3D models may be tackled using image patches or sub-images, and parallel processing on multiple GPUs, or, in some cases, by image downsampling.

With recent advances in machine learning, more and more people are becoming interested in applying these techniques to biomedical imaging, and there is a real and growing risk that not all researchers pay sufficient attention to this serious issue. We also emphasize the need to document how the CV is implemented, the architecture used, how the different hyperparameter choices/tunings are made and include their values where possible. Besides, we advocate reproducibility and encourage the community to take a step towards transparency in deep/machine learning in medical image analysis by publicly releasing code, including containers and a link to open datasets(Celi et al., 2019). Moreover, a blind evaluation on external test sets—i.e., within open challenges—is highly recommended.

One limitation of this study is due to the substantial overfitting we observed while applying a subject-level split for training our models. This overfitting is manifested by the very high accuracy in training sets compared to that observed in test sets (Table3.4). Focussing our efforts on alleviating overfitting may have improved performance in the test set, thus reducing the extent of the faulty boost due to the slice-level split. Moreover, in this study, we have not assessed all data leakage types, including late split and hyperparameters optimization in the test set—that may also be present in 3D CNN studies. We have found evidence of all these data leakage issues in the recent literature (see Tables3.1, 3.2, 3.3), and we plan to quantify their effect in our future work systematically.

In conclusion, training a 2D CNN model for analyzing 3D brain image data must be performed using a subject-level CV to prevent data leakage. The adoption of slice-based CV results in very optimistic model performances, especially for small datasets, as the extent of the overestimation due to data leakage is severe.

6.2 An interpretable, leakage free and reproduciable deep learning framework for analyzing neuroimaging data

As part of this work, a python function tool was developed, which uses the Tensorflow library as backbone, dedicated to deep learning analysis and more specifically aimed at classification and regression analysis of neuroimaging data, explicitly brain MRI data with the features of leakage-free pre-processing, transfer-learning based model training to prevent overfitting, reproducibility and interpretability of the results. The capabilities of our algorithm were put in to practice on real application problems according to different predictive schemes, demonstrating usability, versatility, flexibility and computational efficiency. In the technological landscape regarding the implementation of solutions of tools for deep learning analysis, there are several approaches and software tools similar to each other but with specific peculiarities. The most popular software tools based on Tensorflow library include:

NiftyNet: is a Tensorflow based opensource CNN platform for research in medical image analysis and aimed at sharing networks and pre-trained models which are used for classification and segmentation tasks(Gibson et al., 2018). It is developed with features of customizable interfaces of network components incorporating comprehensive evaluation metrics for segmentation task, support for different dimensional inputs of 2D, 2.5D, 3D and 4D, providing efficient training with multiple GPU support and more others.

Deep learning toolkit for Medical imaging (DLTK): this is also a toolkit written in python on top of Tensorflow framework(Pawlowski et al., 2017), which supports classification, segmentation, regression and super-resolution tasks. It is developed to enable fast prototyping and to ensure reproducibility in image analysis applications, especially medical imaging.

Nobrainer: is a deep learning framework for 3D image processing. It includes several 3D CNNs, methods for loading and augmenting volumetric data, losses and metrics for 3D data and simple utilities for model training, evaluation, prediction and transfer learning(Kaczmarzyk et al.). It also provides pre-trained models for brain extraction, brain segmentation, brain generation and other analysis problems.

All of the above toolkits are popular deep learning frameworks, which provide many openly available pre-trained CNN models for multiple analysis problems. However, all these tools do not incorporate interpretability features and they require integrating AI explainability tools to show the results achieved are reasonable. Our deep learning tool has been successfully used for the study of VMCI in patients with SVD using multiparameteric brain MRI data and for classification analysis of AD patients versus HC subjects employing T1-weighted MRI data. The results obtained from the VMCI analysis were exhibited at the 35th International conference of Computer Assisted Radiology and Surgery in 2021.

6.3 Prediction of the cognitive performance in patients with small vessel disease and mild cognitive impairment: a deep learning approach

In this first study, we demonstrated that deep learning methods could be useful in predicting neuropsychological scores in patients with SVD and MCI by automatically extracting information from different MRI and MRI-derived maps. The prediction outcomes of our CNN models showed marked correlations with cognitive scores. SDMT (0.569), MoCA (0.523), TMT-part A (0.513), and Stroop (0.485) are among the tests that have been predicted with good accuracy. Although Shi and colleagues found better results for the prediction of a MoCA score (r=0.77 – 0.80) using a combination of 8 demographic and neuroimaging features, the reported values were on the validation set rather than the unseen test set. In addition, a larger sample size was used to train an SVR model (Shi et al., 2018). Instead, smaller values were obtained in the same population using a least absolute shrinkage and selection operator (LASSO) regression trained on 13 demographic and neuroimaging features (coefficient of correlation = 0.354) (Pantoni et al., 2019). However, in that study, DTI-derived indices were not available and the model predicted demographically adjusted TMT-A scores, rather than raw scores.

TMT score, which mainly measures the psychomotor speed, was best predicted from DTI-derived features MD and FA with a correlation coefficient of 0.51 and 0.5 respectively. This is in accordance with the fact that cognitive deficits associated with information processing speed are results of white matter damage and deteriorations (Papp et al., 2014) which are well explained by changes in DTI-derived quantitative measures(Ciulli et al., 2016, Pasi et al., 2016).

The contribution of demographic variables in improving the prediction accuracy of the CNN models is demonstrated by the learning curves generated in experiment 1 (Figure 3.2). Level of education, age and sex are identified as the best predictors of the cognitive status among different demographic variables (Casanova, et al., 2020). Methodologically, our approach takes care of the common issues that may exist in applying deep learning methods for analyzing neuroimaging data. We implemented a nested 10-fold CV, which allows getting an unbiased estimate of the selected models. 10-fold CV, which is a common choice for model validation procedure, is sufficient for model selection(Breiman and Spector, 1992). In addition, it balances the computational cost of the evaluation procedure and unbiased estimate of model performance by providing a good bias-variance balance(Hastie et al., 2009, Lemm et al., 2011).

Moreover, we ensured that we did not introduce data leakage which is caused by performing slice-level data split during the train/validation/test data division procedure(Wen et al., 2020). Instead, we split the data based on 3D MRI volumes (subject-level split): For each CV loop, a sub-set of subjects was used for training the CNN models, another subset for validating the model, and the rest for testing the chosen best model. This assures that all 2D slices of each patient are included in only one of the sample datasets (training/validation/test) hence preventing models from overfitting due to data leakage.

As a key limitation, the size of our dataset is relatively small to train deep learning models with good predictive accuracy. Although we tried to solve the problem by using the transfer learning method, yet larger datasets would produce models with better generalization and more accurate prediction scores by fine-tuning the pre-trained models well.

6.4 Development of interpretable deep learning system for the classification of Alzheimer disease

In this study, a deep learning model customized from VGG16 is proposed for a binary classification problem of AD and HC subjects. The proposed CNN was trained by employing transfer learning techinique to prevent model overfitting caused by the small size of the training data. The model was trained on a brian image collection of the OASIS dataset achieving an average accuracy of 71.6% on the test set. Comparing to previous studies employing OASIS dataset(Yagis et al., 2019, Yagis et al., 2021), our model classifies AD and HC subjects with a better accuracy. Although, in the other few studies (Tufail et al., 2020, Saratxaga et al., 2021, Massalimova and Varol, 2021) the authors reported higher accuracies, these results are due to the use of larger number of subjects, multimodality and the application of data augmentation to improve the performance of the model. Apart from repoting the model's performance, neither of these studies included model visualization tools to make sure that the models are focusing on meaningful brain regions to perform the predictive analysis. While, our proposed model incorporates four different visualization methods which allows strengthening the reliability of our system.

The results also showed that the interpretation techniques highlight features located around the frontal lobe, the parital lobe, cerebral cortex and areas around the thalamus. The SHAP method outperform the other methods in localizing the frontal lobe. While the cortical atrophy and alterations around the thalamus were captured by the Grad-CAM method. The visualization outcomes by the CAM-based technique are very much distributed and the GradCAM method has a better localization ability.

Regarding the role of visualization techniques in identifying a biased model, such as a model trained by introducing data leakage, although the occlusion map method outperforms the other approaches, SHAP and CAM also perform well by producing heatmaps with a probability value of close to 0. Rather the Grad-CAM method fails by producing goodlooking heatmaps.

Since each interpretability approach has its own limitation, incorporating multiple visualization methods helps better understand deep learningbased predictive systems.

Chapter 7

7. Conclusion

This work includes two parts. In the first part of the research, a literature review was performed to identify the challenges and pitfalls of employing deep learning predictive systems for analyzing neuroimaging data and based on our observations, we found out that the exisiting results in a considerable number of research papers were associated with some kind of methodological bias. Focusing on a data-leakage problem caused by splitting the 3D brain MRI dataset based on slice-level, we assessed and quantified the model performance overestimation seen in 2D CNN models developed for the classification of AD versus HC and PD versus HC subjects. Our results confirmed that slice-level data leakage, which is seen in a significant number of studies, results in overly optimistic models producing falsy good results and the effect being worse when using datasets of small size.

Since, the first part of this study led us to the conclusion that there is a need for designing a deep learning system that alleviates the pitfalls seen in analyzing neuroimaging data, in the second part of this work, a deep learning predictive system, which has features of leakage-free, interpretability, and reproducibility, was developed in a python language based on Tensorflow backend Keras and Scikit-learn libraries aimed at conducting classification and regression analysis with flexible options of employment starting from simple model training and evaluation based on holdout and k-fold CV strategy to the complex procedure of hyperparameter optimization and model validation according to a nested k-fold crossvalidation scheme.

The ability of the predictive system to predict the overall neuropsychological performance as assessed by MoCA, SDMT, TMT-A, ROCF, Stroop and Visual Search scoring in patients with SVD and MCI was demonstrated using multimodal MRI and DTI and patient demographic data. The results have also confirmed the importance of deep learning approaches in the evaluation of the cognitive performance status based on neuroimaging and demographic data

The tool kit was also employed for the classification of Alzheimer disease, both in 2D and 3D model architectures scenarios applied to MRI data. In conducting the studies, the advantages of CNN visualization techniques that are available in the case of 2D models, were highlighted, which allowed interpreting the decisions produced by the model and to see relevant image regions or features that are given higher importance by the model's prediction.

From the clinical point of view, knowledge of the disease in SVD patients with MCI has been expanded, highlighting the brain substrates underlying the alterations of psychomotor speed. The results confirm that cognitive deficits associated with information processing speed are results of white matter damage and deteriorations and this is well explained by the changes in DTI-derived quantitative measures. This also demonstrate also the possibility of using machine learning strategies in understanding the neural substrates of cognitive impairment in studies with larger sample size.

Possible feature developments also include:

1. greater automation of some changes to the configuration of predictive schemes that can be set in the toolkit, such as the number of parameters that can be optimized.
2. expanding the system to support ensembles of CNNs
3. The application of the system to other neurological pathologies, especially those where bigger dataset of the pathologies are available.

8. References

Abrol, A. et al., 2021. Deep learning encodes robust discriminative neuroimaging representations to outperform standard machine learning. *Nature communications,* Volume 12(1), pp. 1-17.

Al-Khuzaie, F., Bayat, O. & Duru, A. D., 2021. Diagnosis of Alzheimer disease using 2D MRI slices by convolutional neural network.. *Applied Bionics and Biomechanics.*

Alam, S., Kwon, G. R., Kim, J. I. & Park, C. S., 2017. Twin SVM-based classification of Alzheimer's disease using complex dual-tree wavelet principal coefficients and LDA.. *Journal of healthcare engineering.*

Alogna, E., Giacomello, E. & Loiacono, D., 2020. Brain Magnetic Resonance Imaging Generation using Generative Adversarial Networks.. *In 2020 IEEE Symposium Series on Computational Intelligence (SSCI),* pp. 2528-2535.

Alpaydin, E., Murphy, K. P. & Bishop, C. M., 2010. *Machine Learning. The New AI..* s.l.:MIT Press.

Anon., 2019. [Online]
Available at: https://www.tutorialandexample.com/artificial-neural-network-interview-questions

Anon., 2022. *machine learning group.* [Online]
Available at: https://www.cosmos.esa.int/web/machine-learning-group/convolutional-neural-networks-introduction

Arjovsky, M., Chintala, S. & Bottou, L., 2017. Wasserstein generative adversarial networks.. *In International conference on machine learning,* pp. 214-223.

Avants, B. B. et al., 2011. A reproducible evaluation of ANTs similarity metric performance in brain image registration.. *Neuroimage,* Volume 54(3), pp. 2033-2044.

Bahrami, K. et al., 2016. Reconstruction of 7T-Like Images From 3T MRI.. *IEEE Trans Med Imaging,* Volume 35, pp. 2085-2097.

Basheera, S. & Ram, M. S., 2019. Convolution neural network–based Alzheimer's disease classification using hybrid enhanced independent component analysis based segmented gray matter of T2 weighted magnetic resonance imaging with clinical valuation. *Alzheimer's & Dementia: Translational Research & Clinical Interventions,* Volume 5, pp. 974-986.

Bateman, R. et al., 2012. Clinical and biomarker changes in dominantly inherited Alzheimer's disease.. *N Engl J Med,* Volume 367, p. 795.

Bengio, Y., 2009. Learning deep architectures for AI, Found.. *Trends Mach. Learn,* Volume 2, pp. 1-127.

Bengio, Y., Lamblin, P., Popovici, D. & Larochelle, H., 2007. Greedy layer-wise training of deep networks. *in Advances in Neural Information Processing Systems,* p. 153.

Bermudez, C. et al., 2018. Learning implicit brain MRI manifolds with deep learning.. *In Medical Imaging: Image Processing International Society for Optics and Photonics.,* Volume 10574, p. 105741L.

Billones, C., Demetria, O. L., Hostallero, D. E. & Naval, P. C., 2016. DemNet: a convolutional neural network for the detection of Alzheimer's disease and mild cognitive impairment.. *In 2016 IEEE region 10 conference (TENCON),* pp. 3724-372.

Blum, A., KalaI, A. & Langford, J., 1999. Beating the hold-out: Bounds for k-fold and progressive cross-validation.. *In Proceedings of the twelfth annual conference on Computational learning theory*, pp. 203-208.

Bobinski, M. et al., 1999. The histological validation of post mortem magnetic resonance imaging-determined hippocampal volume in Alzheimer's disease.. *Neuroscience,* Volume 95(3), pp. 721-725.

Braak, H. & Braak, E., 1991. Neuropathological stageing of Alzheimer-related changes.. *Acta neuropathologica,* Volume 82(4), pp. 239-259.

Braak, H., Thal, D. R., Ghebremedhin, E. & Del Tredici, K., 2011. Stages of the pathologic process in Alzheimer disease: age categories from 1 to 100 years.. *Journal of Neuropathology & Experimental Neurology,* Volume 70(11), pp. 960-969.

Bradski, G. & Kaehler, A., 2008. Learning OpenCV: Computer vision with the OpenCV library.. *" O'Reilly Media, Inc."*..

Breiman, L. & Spector, P., 1992. Submodel selection and evaluation in regression. The X-random case.. *International statistical review/revue internationale de Statistique,* pp. 291-319.

Brett, M. et al., 2019. nipy/nibabel: 2.3.3. Zenodo.

Bussola, N. et al., 2021. AI slipping on tiles: Data leakage in digital pathology.. *In International Conference on Pattern Recognition,* Issue Springer, Cham., pp. 167-182.

Casanova, R. et al., 2020. Investigating predictors of cognitive decline using machine learning. *The Journals of Gerontology,* pp. 75(4),733-742.

Cawley, G. C. & Talbot, N. L., 2010. On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. *The Journal of Machine Learning Research,* Volume 11, p. 2079–2107.

Celi, L., Citi, L., Ghassemi, M. & Pollard, T. J., 2019. The PLOS ONE collection on machine learning in health and biomedicine: Towards open code and open data.. *PloS one,* Volume 14(1), p. e0210232.

Chang, P., 2016. Fully convolutional deep residual neural networks for brain tumor segmentation.. *In International workshop on Brainlesion: Glioma, multiple sclerosis, stroke and traumatic brain injuries*, pp. 108-118.

Chartrand, G. et al., 2017. Deep learning: a primer for radiologists.. *Radiographics,* Volume 37(7), pp. 2113-2131.

Ching, T. et al., 2018. Opportunities and obstacles for deep learning in biology and medicine.. *Journal of The Royal Society Interface,* Volume 15(141), p. 20170387.

Chlap, P. et al., 2021. A review of medical image data augmentation techniques for deep learning applications.. *Journal of Medical Imaging and Radiation Oncology.*.

Chollet, F. & others, 2015. Keras..

Ciulli, S. et al., 2016. Prediction of Impaired Performance in Trail Making Test in MCI Patients With Small Vessel Disease Using DTI Data. *IEEE J Biomed Health Inform,* Volume 20, p. 1026–1033.

Ciulli, S., Luca , C., Emilia , S. & Raffaella , V., 2016. Prediction of impaired performance in trail making test in MCI patients with small vessel disease using DTI data. *IEEE journal of biomedical and health informatics,* Volume 20(4), pp. 1026-1033.

Cook, P. A. et al., 2006. Camino: Open-Source Diffusion-MRI Reconstruction and Processing.. *14th Scientific Meeting of the International Society for Magnetic Resonance in Medicine,* p. 2759.

Cook, S., 2014. CUDA Programming: a Developer's Guide to Parallel Computing with GPUs.. *Elsevier Science.*

Cuingnet, R. et al., 2011. Automatic classification of patients with Alzheimer's disease from structural MRI: a comparison of ten methods using the ADNI database.. *neuroimage,* Volume 56(2), pp. 766-781.

Dabbura, I., 2018. *Towards Data Science.* [Online]
Available at: https://towardsdatascience.com/coding-neural-network-dropout-3095632d25ce

Davatzikos, C., 2019. Machine learning in neuroimaging: Progress and challenges. *Neuroimage,* Volume 197, p. 652–656.

Developers., T., 2021. TensorFlow.. *(Zenodo).*

Dongare, A., Kharde, R. R. & Kachare, A. D., 2012. Introduction to artificial neural network.. *International Journal of Engineering and Innovative Technology (IJEIT),* Volume 2(1), pp. 189-194.

Dou, Q. et al., 2016. Dou, Q. et al. Automatic Detection of Cerebral Microbleeds From MR Images via 3D Convolutional Neural Networks.. *IEEE Transactions on Medical Imaging,* Volume 35, pp. 1182-1195.

Duchi, J., Hazan, E. & Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization.. *Journal of machine learning research,* p. 12(7).

Duchi, J., Hazan, E. & Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization.. *Journal of machine learning research,* Volume 12(7).

Esmaeilzadeh, S., Yang, Y. & Adeli, E., 2018. End-to-end parkinson disease diagnosis using brain mr-images by 3d-cnn.. *arXiv preprint arXiv,* p. 1806.05233.

Farooq, A., Anwar, S., Awais, M. & Rehman, S., 2017. A deep CNN based multi-class classification of Alzheimer's disease using MRI.. *in 2017 IEEE International Conference on Imaging Systems and Techniques (IST),* pp. 1-6.

Feng, W. et al., 2020. Automated MRI-based deep learning model for detection of Alzheimer's disease process.. *International Journal of Neural Systems,* Volume 30(06), p. 2050032.

Fukushima, K. & Miyake, S., 1982. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition.. *In Competition and cooperation in neural nets,* pp. 267-285.

Fung, Y. et al., 2019. Alzheimer's disease brain mri classification: Challenges and insights.. *arXiv preprint arXiv:1906.04231*..

G.R., C., Khazaei, J., Ghobadian, B. & Goudarzi, A. M., 2007. Prediction of process and product parameters in anorange juice spray dryer using artificial neural networks.. *J.Food Eng,* Volume 84(4), pp. 534-543.

Gao, J. et al., 2021. Multisite autism spectrum disorder classification using convolutional neural network classifier and individual morphological brain networks. *Frontiers in Neuroscience,* Volume 14, p. 1473.

Gibson, E. et al., 2018. NiftyNet: a deep-learning platform for medical imaging. *Computer Methods and Programs in Biomedicine.*

Glorot, X. & Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. *In Proceedings of the thirteenth international conference on artificial intelligence and statistics . JMLR Workshop and Conference Proceedings.,* pp. 249-256.

Glorot, X., Bordes, A. & Bengio, Y., 2011. Deep Sparse Rectifier Neural Networks,. , *in: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. Presented at the Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics,* p. 315–323.

Goodfellow, I., Bengio, Y. & Courville, A., 2016. *Deep learning.*. s.l.:The MIT Press.

Goodfellow, I., Bengio, Y. & Courville, A., 2016. Regularization for deep learning. . In: *Deep learning.* s.l.:MIT Press, pp. 216-261.

Goodfellow, I. et al., 2014. Generative adversarial nets.. *Advances in neural information processing systems,* Volume 27.

Gray, K. et al., 2013. Random forest-based similarity measures for multi-modal classification of Alzheimer's disease.. *NeuroImage,* Volume 65, pp. 167-175.

Greenspan, H., van Ginneken, B. & Summers, R. M. E. D. L. i. M. I., 2016. Overview and Future Promise of an Exciting New Technique.. *IEEE Trans. Med. Imaging,* Volume 35, p. 1153–1159.

Greve, D. N. & Fischl, B., 2009. Accurate and robust brain image alignment using boundary-based registration. *NeuroImage,* Volume 48, p. 63–72.

Grundke-Iqbal, I. et al., 1986. Abnormal phosphorylation of the microtubule-associated protein tau (tau) in Alzheimer cytoskeletal pathology.. *Proceedings of the National Academy of Sciences,* Volume 83(13), pp. 4913-4917.

Gunawardena, K. A. N. N. P., Rajapakse, R. N. & Kodikara, N. D., 2017. Applying convolutional neural networks for pre-detection of alzheimer's disease from structural MRI data.. *In 2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP),* pp. 1-7.

Gunawardena, K., Rajapakse, R. N. & Kodikara, N. D., 2017. Applying convolutional neural networks for pre-detection of alzheimer's disease from structural MRI data.. *In 2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP),* pp. 1-7.

Gupta, A., Ayhan, M. & Maida, A., 2013. Nural image bases to represent neuroimaging data. *in International Conference on Machine Learning (Atlanta, GA),* p. 87–994.

Hague, S., Klaffke, S. & Bandmann, O., 2005. Neurodegenerative disorders: Parkinson's disease and Huntington's disease.. *Journal of Neurology, Neurosurgery & Psychiatry*, Volume 76(8), pp. 1058-1063.

Han, C. et al., 2018. GAN-based synthetic brain MR image generation.. *In 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pp. 734-738.

Han, C. et al., 2019. Combining noise-to-image and image-to-image GANs: brain MR image augmentation for tumor detection.. *IEEE Access*, Volume 7, pp. 156966-156977.

Han, X., 2017. MR-based synthetic CT generation using a deep convolutional neural network method.. *Med. Phys*, Volume 44, p. 1408–1419.

Han, X. et al., 2018. Brain extraction from normal and pathological images: a joint PCA/image-reconstruction approach.. *NeuroImage*, Volume 176, pp. 431-445.

Hasan, A. et al., 2019. Combining Deep and Handcrafted Image Features for MRI Brain Scan Classification.. *IEEE Access* , Volume 7, p. 79959–79967.

Hastie, T., Tibshirani, R. & Friedman, J., 2009. The elements of statistical learnin.. p. 33.

Hatcher, W. G. & Yu, W., 2018. A Survey of Deep Learning: Platforms, Applications and Emerging Research Trends.. *IEEE* , Volume 6, p. 24411–24432.

He, K., Zhang, X., Ren, S. & Sun, J., 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,. *in: 2015 IEEE International Conference on Computer Vision (ICCV). Presented at the IEEE International Conference on Computer Vision (ICCV)*, p. 1026–1034.

Hinton, G. E. et al., 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580 [cs]*.

Hoehn, M. M. & Yahr, M. D., 1967. Parkinsonism: onset, progression and mortality.. *Neurology*, Volume 17, pp. 427-442.

Hon, M. & Khan, N. M., 2017. Towards Alzheimer's disease classification through transfer learning.. *In 2017 IEEE International conference on bioinformatics and biomedicine (BIBM)*, pp. 1166-1169.

Hosseini-Asl, E. et al., 2018. Alzheimer's Disease Diagnostics by a 3D Deeply Supervised Adaptable Convolutional Network.. *Frontiers in bioscience (Landmark edition)*, Volume 23, pp. 584-596.

Hosseini-Asl, E., Gimel'farb, G. & El-Baz, A., 2016. Alzheimer's disease diagnostics by a deeply supervised adaptable 3D convolutional network.. *arXiv preprint arXiv:1607.00556*.

Hosseini-Asl, E., Keynton, R. & El-Baz, A., 2016. Alzheimer's disease diagnostics by adaptation of 3D convolutional network.. *In 2016 IEEE international conference on image processing (ICIP)*, pp. 126-130.

Huang, Y. et al., 2019. Diagnosis of Alzheimer's disease via multi-modality 3D convolutional neural network.. *Frontiers in Neuroscience*, Volume 13, p. 509.

Hubel, D. & Wiesel, T. N., 1968. Receptive fields and functional architecture of monkey striate cortex.. *The Journal of physiology*, Volume 195(1), pp. 215-243.

Hubel, D. & Wiesel, T. N., 1968. Receptive fields and functional architecture of monkey striate cortex.. *The Journal of physiology*, Volume 195(1), pp. 215-243.

Huff, D., Weisman, A. J. & Jeraj, R., 2021. Interpretation and visualization techniques for deep learning models in medical imaging.. *Physics in Medicine & Biology*, Volume 66(4), p. 04TR01.

Hutson, M., 2018. *Artificial intelligence faces reproducibility crisis*, s.l.: s.n.

Ibarretxe-Bilbao, N., Tolosa, E., Junque, C. & Marti, M. J., 2009. MRI and cognitive impairment in Parkinson's disease.. *Movement Disorders*, Volume 24(S2), pp. S748-S753.

Iizuka, N. et al., 2003. Oligonucleotide microarray for prediction of early intrahepatic recurrence of hepatocellular carcinoma after curative resection.. *The lancet*, Volume 361(9361), pp. 923-929.

Iizuka, T., Fukasawa, M. & Kameyama, M., 2019. Deep-learning-based imaging-classification identified cingulate island sign in dementia with Lewy bodies.. *Scientific reports*, Volume 9(1), pp. 1-9.

Islam, J. & Zhang, Y., 2018. Brain MRI analysis for Alzheimer's disease diagnosis using an ensemble system of deep convolutional neural networks.. *Brain Inform*, Volume 5, p. 2.

Islam, J. & Zhang, Y., 2020. GAN-based synthetic brain PET image generation.. *Brain informatics*, Volume 7(1), pp. 1-12.

Jack Jr, C. et al., 2009. Serial PIB and MRI in normal, mild cognitive impairment and Alzheimer's disease: implications for sequence of pathological events in Alzheimer's disease.. *Brain*, Volume 132(5), pp. 1355-1365.

Jain, A., Mao, J. & Mohiuddin, K. M., 1996. Artificial neural networks :A tutorial.. *Computer*, Volume 29(3), pp. 31-44.

Jain, R., Jain, N., Aggarwal, A. & D, J., 2019. Convolutional Neural Network based Alzheimer's Disease Classification from Magnetic Resonance Brain Images.. *Cognitive Systems Research*, Volume 57.

Jenkinson, M., Bannister, P., Brady, M. & Smith, S., 2002. Improved optimization for the robust and accurate linear registration and motion correction of brain images.. *Neuroimage,* Volume 17, p. 825–841.

Jenkinson, M. & Smith, S., 2001. A global optimisation method for robust affine registration of brain images.. *Med Image Anal,* Volume 5, p. 143–156.

Jimeno, M. M. et al., 2022. ArtifactID: Identifying artifacts in low-field MRI of the brain using deep learning.. *Magnetic resonance imaging,* 89(., Ogbole, G., & Geethanath, S.), pp. 42-48.

Jo, T., Nho, K. & Saykin, A. J., 2019. Deep learning in Alzheimer's disease: diagnostic classification and prognostic prediction using neuroimaging data.. *Frontiers in aging neuroscience,* Volume 11, p. 220.

Kaczmarzyk, J. et al., n.d. https://doi.org/10.5281/zenodo.4995077.

Kalogirou, S., 2000. Applications of artificial neural-networks for energy systems.. *Applied energy,* Volume 67(1-2), pp. 17-35.

Karasawa, H., Liu, C. L. & Ohwada, H., 2018. Deep 3d convolutional neural network architectures for alzheimer's disease diagnosis.. *In Asian conference on intelligent information and database systems,* Issue Springer, Cham, pp. 287-296.

Karthik, R., Menaka, R., Johnson, A. & Anand, S., 2020. Neuroimaging and deep learning for brain stroke detection-A review of recent advancements and future prospects. *Computer Methods and Programs in Biomedicine,* p. 105728.

Kassubek, J., 2017. The application of neuroimaging to healthy and diseased brains: present and future.. *Frontiers in neurology,* Volume 8, p. 61.

Kaufman, S., Rosset, S., Perlich, C. & Stitelman, O., 2012. Leakage in data mining: Formulation, detection, and avoidance.. *ACM Transactions on Knowledge Discovery from Data (TKDD),* pp. 1-21.

Kazuhiro, K. et al., 2018. Generative adversarial networks for the creation of realistic artificial brain magnetic resonance images.. *Tomography,* Volume 4(4), pp. 159-163.

Kelley, B. J. & Petersen, R. C., 2007. Alzheimer's disease and mild cognitive impairment.. *Neurologic clinics,* Volume 25(3), pp. 577-609.

Khagi, B., Lee, B., Pyun, Y. J. & Kwon, G. R., 2019. CNN models performance analysis on MRI images of oasis dataset for distinction between healthy and Alzheimer's patient.. *In 2019 International Conference on Electronics, Information, and Communication (ICEIC),* pp. 1-4.

Kimura, M. & Tanaka, M., 2020. New perspective of interpretability of deep neural networks.. *In 2020 3rd International Conference on Information and Computer Technologies (ICICT)*, pp. 78-85.

Kingma, D. & Ba, J., 2014. Adam: A method for stochastic optimization.. *arXiv preprint arXiv*, p. 1412.6980.

Kobayashi, S., Kane, T. B. & Paton, C., 2018. Kobayashi, S., Kane, T.B. and Paton, C., 2018. The privacy and security implications of open data in healthcare.. *Yearbook of medical informatics*, Volume 27(01), pp. 041-047.

Korolev, S., Safiullin, A., Belyaev, M. & Dodonova, Y., 2017. Residual and plain convolutional neural networks for 3D brain MRI classification.. *In 2017 IEEE 14th international symposium on biomedical imaging (ISBI 2017)*, pp. 835-838.

Krogh, A., 2008. What are artificial neural networks?. *Nature biotechnology*, Volume 26(2), pp. 195-197.

Kuang, D. et al., 2014. Discrimination of ADHD based on fMRI data with deep belief network. *In International Conference on Intelligent Computing*, pp. 225-232.

Lan, L. et al., 2020. Generative adversarial networks and its applications in biomedical informatics.. *Frontiers in Public Health*, Volume 8, p. 164.

LeCun, Y., Bengio, y. & Hinton, g., 2015. Deep learning.. *Nature*, Volume 521(7553), pp. 436-444.

LeCun, Y. et al., 1989. Handwritten digit recognition with a back-propagation network.. *Advances in neural information processing systems*, Volume 2.

Leemans, A. & Jones, D. K., 2009. The B-matrix must be rotated when correcting for subject motion in DTI data. *Magn Reson Med*, Volume 61, p. 1336–1349.

Lehericy, S. et al., 1994. Amygdalohippocampal MR volume measurements in the early stages of Alzheimer disease.. *American Journal of Neuroradiology*, 15(5)(., Deweer, B., Dubois, B. and Marsault, C.,), p. 929.

Lemm, S., Blankertz, B., Dickhaus, T. & Müller, K. R., 2011. Introduction to machine learning for brain imaging.. *Neuroimage*, Volume 56(2), pp. 387-399..

Libero, L. E. et al., 2015. Multimodal neuroimaging based classification of autism spectrum disorder using anatomical, neurochemical, and white matter correlates.. *Cortex*, Volume 66, pp. 46-59.

Linardatos, P., Papastefanopoulos, V. & Kotsiantis, S., 2020. Explainable ai: A review of machine learning interpretability methods.. *Entropy,* Volume 23(1), p. 18.

Lin, M., Chen, Q. & Yan, S., 2014. Network In Network.. *arXiv:1312.4400 [cs]*.

Lipton, Z., 2018. The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery.. *Queue,* Volume 16(3), pp. 31-57.

Li, R. et al., 2014. Deep learning based imaging data completion for improved brain disease diagnosis.. *in MICCAI International Conference on Medical Image Computing and Computer-Assisted Intervention,* Volume 17, p. 305–312.

Liu, F. et al., 2018. Deep Learning MR Imaging–based Attenuation Correction for PET/MR Imaging.. *Radiology,* Volume 286, pp. 676-684.

Liu, F. & Shen, C., 2014. Learning deep convolutional features for MRI based Alzheimer's disease classification.. *arXiv preprint arXiv,* p. 1404.3366.

Liu, J. et al., 2018. Applications of deep learning to MRI images: A survey. *Big Data Mining and Analytics,* Volume 1(1), pp. 1-18.

Liu, M., Cheng, D., Wang, K. & Wang, Y., 2018. Multi-modality cascaded convolutional neural networks for Alzheimer's disease diagnosis.. *Neuroinformatics,* Volume 16(3), pp. 295-308.

Liu, M. et al., 2020. A multi-model deep convolutional neural network for automatic hippocampus segmentation and classification in Alzheimer's disease. *Neuroimage,* Volume 208, p. 116459.

Liu, S. et al., 2015. [69Multimodal Neuroimaging Feature Learning for Multiclass Diagnosis of Alzheimer's Disease.. *IEEE Trans. Biomed. Eng.,* Volume 62, pp. 1132-1140.

Liu, S. et al., 2014. Early diagnosis of Alzheimer's disease with deep learning.. *In IEEE 11th international symposium on biomedical imaging (ISBI),* pp. 1015-1018.

Liu, X. et al., 2013. Locally linear embedding (LLE) for MRI based Alzheimer's disease classification.. *Neuroimage,* Volume 83, pp. 148-157.

Long, J., Shelhamer, E. & Darrell, T., 2015. Fully convolutional networks for semantic segmentation. *in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).,* p. 3431–344.

L, P., 1998. Early stopping—but when?. *In Neural Networks: Tricks of the trade* , Volume Springer, pp. 55-69.

Lu, D. & Weng, Q., 2007. A survey of image classification methods and techniques for improving classification performance.. *International journal of Remote sensing,* Volume 28(5), pp. 823-870.

Lui, Y. et al., 2014. Classification algorithms using multiple MRI features in mild traumatic brain injury.. *Neurology,* Volume 83, pp. 1235-1240.

Lundberg, S. & Lee, S. I., 2017. A unified approach to interpreting model predictions.. *In Proceedings of the 31st international conference on neural information processing systems,* pp. 4768-4777.

Lundervold, A. & Lundervold, A., 2019. An overview of deep learning in medical imaging focusing on MRI.. *Zeitschrift für Medizinische Physik,* Volume 29(2), pp. 102-127.

Lu, P. et al., 2022. A Two-Stage Model for Predicting Mild Cognitive Impairment to Alzheimer's Disease Conversion.. *Frontiers in Aging Neuroscience,* Volume 14.

Maier, O. et al., 2015. Classifiers for Ischemic Stroke Lesion Segmentation: A Comparison Study.. *PLOS ONE,* Volume 10, p. e0145118.

Marcus, D. et al., 2007. Open Access Series of Imaging Studies (OASIS): cross-sectional MRI data in young, middle aged, nondemented, and demented older adults. *J Cogn Neurosci,* Volume 19, p. 1498–1507.

Marek, K. et al., 2018. The Parkinson's progression markers initiative (PPMI)–establishing a PD biomarker cohort.. *Annals of clinical and translational neurology,* 5(12)(Coffey, C.S., Caspell-Garcia, C., Simuni, T., Jennings, D., Tanner, C.M., Trojanowski, J.Q. and Shaw, L.M.,), pp. 1460-1477.

Massalimova, A. & Varol, H. A., 2021. Input Agnostic Deep Learning for Alzheimer's Disease Classification Using Multimodal MRI Images.. *In 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC),* pp. 2875-2878.

Mehmood, A., Maqsood, M., Bashir, M. & Shuyuan, Y., 2020. A deep Siamese convolution neural network for multi-class classification of Alzheimer disease.. *Brain sciences,* Volume 10(2), p. 84.

Micheli-Tzanakou, E., 2011. Artificial neural networks: an overview.. *Network: Computation in Neural Systems ,* Volume 22(1-4), pp. 208-230.

Moorhouse, P. & Rockwoo, K., 2008. Vascular cognitive impairment: current concepts and clinical developments.. *Lancet Neuro,* Volume 7, p. 246–255.

Morris, J. C., 1993. The Clinical Dementia Rating (CDR): current version and scoring rules.. *Neurology,* Volume 43, p. 2412–2414.

Morris, J. et al., 2001. Mild cognitive impairment represents early-stage Alzheimer disease.. *Arch. Neurol.,* Volume 58, pp. 397-405.

Mortimer, J. et al., 2004. Delayed recall, hippocampal volume and Alzheimer neuropathology: findings from the Nun Study.. *Neurology,* Volume 62(3), pp. 428-432.

Mrudang, D. P., Parth, D. S. & Sunil, J., 2019. Medical image diagnosis for disease detection: A deep learning approach. *In U-Healthcare Monitoring Systems,,* p. 37–60.

Murad, M. et al., 2020. Efficient Reconstruction Technique for Multi-Slice CS-MRI Using Novel Interpolation and 2D Sampling Scheme.. *IEEE Access,* Volume 8, p. 117452–117466.

Mustafeez, A. Z., 2022. *Educative.* [Online]
Available at: https://www.educative.io/answers/what-is-early-stopping

Nalepa, J., Marcinkiewicz, M. & Kawulok, M., 2019. Data augmentation for brain-tumor segmentation: a review.. *Frontiers in computational neuroscience,* Volume 13, p. 83.

Namatēvs, I., 2017. Deep Convolutional Neural Networks: Structure, Feature Extraction and Training. *Information Technology and Management Science* , Volume 20.

Narla, A. et al., 2018. Automated Classification of Skin Lesions: From Pixels to Practice.. *Journal of Investigative Dermatology,* Volume 138, p. 2108–2110.

Nawaz, A. et al., 2020. Deep Convolutional Neural Network based Classification of Alzheimer's Disease using MRI Data.. *In 2020 IEEE 23rd International Multitopic Conference (INMIC),* pp. 1-6.

O'Brien, J. T. et al., 2003. Vascular cognitive impairment.. *Lancet Neurol,* p. 89–98.

O'Sullivan, S. et al., 2020. Developments in AI and Machine Learning for Neuroimaging.. *Artificial Intelligence and Machine Learning for Digital Pathology,* pp. 307-320.

Oakden-Rayner, L., Dunnmon, J., Carneiro, G. & Re, C., 2020. Hidden stratification causes clinically meaningful failures in machine learning for medical imaging.. *in Proceedings of the ACM Conference on Health, Inference, and Learning,* p. 151–159.

Oh, K. et al., 2019. Classification and visualization of Alzheimer's disease using volumetric convolutional neural network and transfer learning.. *Scientific Reports,* Volume 9(1), pp. 1-16.

Pandya, M., Shah, P. D. & Jardosh, S., 2019. Medical image diagnosis for disease detection: A deep learning approach.. *In U-Healthcare Monitoring Systems,* Issue Academic Press, pp. 37-60.

Pan, S. & Yang, Q., 2010. A survey on transfer learning.. *IEEE Transactions on knowledge and data engineering,* Volume 22(10), pp. 1345-1359.

Pantoni, L., 2010. Cerebral small vessel disease: from pathogenesis and clinical characteristics to therapeutic challenges.. *Lancet Neurol,* Volume 9, p. 689–701.

Pantoni, L. et al., 2019. Fractal dimension of cerebral white matter: A consistent feature for prediction of the cognitive performance in patients with small vessel disease and mild cognitive impairment.. *Neuroimage Clin,* Volume 24, p. 101990.

Panwar, H. et al., 2020. Application of deep learning for fast detection of COVID-19 in X-Rays using nCOVnet.. *Chaos, Solitons & Fractals,* 138(Gupta, P.K., Siddiqui, M.K., Morales-Menendez, R. and Singh, V.), p. 109944.

Papp, K. V. et al., 2014. Processing speed in normal aging: Effects of white matter hyperintensities and hippocampal volume loss. *Aging, Neuropsychology, and Cognition,* Volume 21(2), pp. 197-213.

Pasi, M. et al., 2016. White matter microstructural damage on diffusion tensor imaging in cerebral small vessel disease: clinical consequences. *Stroke,* Volume 47(6), pp. 1679-1684.

Pathak, K. & Kundaram, S. S., 2020. Accuracy-Based Performance Analysis of Alzheimer's Disease Classification Using Deep Convolution Neural Network.. *In Soft Computing: Theories and Applications,* pp. 731-744.

Pawlowski, N. et al., 2017. DLTK: State of the Art Reference Implementations for Deep Learning on Medical Images. *arXiv preprint arXiv:1711.06853.*

Payan, A. & Montana, G., 2015. Predicting Alzheimer's disease: a neuroimaging study with 3D convolutional neural networks.. *arXiv preprint arXiv,* p. 1502.02506.

Payan, A. & Montana, G., 2015. Predicting Alzheimer's disease: a neuroimaging study with 3D convolutional neural networks.. *arXiv preprint arXiv:1502.02506.*

Pedregosa, F. et al., 2011. Scikit-learn: Machine Learning in Python.. *Journal of Machine Learning Research,* Volume 12, p. 2825–2830.

Petersen, R. et al., 2010. Alzheimer's disease neuroimaging initiative (ADNI): clinical characterization.. *Neurology,* Volume 74(3), pp. 201-209.

Plis, S. M. et al., 2014. Deep learning for neuroimaging: a validation study.. *Frontiers in neuroscience,* Volume 8, p. 229.

Poggesi, A. et al., 2012. Risk and Determinants of Dementia in Patients with Mild Cognitive Impairment and Brain Subcortical Vascular Changes: A Study of Clinical, Neuroimaging, and Biological Markers-The VMCI-Tuscany Study: Rationale, Design, and Methodology.. *Int J Alzheimers Dis,* Volume 608013.

Prechelt, L., 1998. Early stopping-but when?. *In Neural Networks: Tricks of the trade,* Issue Springer, Berlin, Heidelberg., pp. 55-69.

Puente-Castro, A., Fernandez-Blanco, E., Pazos, A. & Munteanu, C. R., 2020. Automatic assessment of Alzheimer's disease diagnosis based on deep learning techniques.. *Computers in Biology and Medicine,* Volume 120, p. 103764.

Puranik, M., Shah, H., Shah, K. & Bagul, S., 2018. Intelligent Alzheimer's Detector Using Deep Learning.. *in 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS),* pp. 318-323.

Qian, N., 1999. On the momentum term in gradient descent learning algorithms.. *Neural networks,* Volume 12(1), pp. 145-151.

Qiu, S. et al., 2018. Fusion of deep learning models of MRI scans, Mini–Mental State Examination, and logical memory test enhances diagnosis of mild cognitive impairment.. *Alzheimer's & Dementia: Diagnosis, Assessment & Disease Monitoring,* Volume 10, pp. 737-749.

Qiu, S. et al., 2020. Development and validation of an interpretable deep learning framework for Alzheimer's disease classification.. *Brain,* Volume 143 (6), pp. 1920-1933.

Radford, A., Metz, L. & Chintala, S., 2015. Unsupervised representation learning with deep convolutional generative adversarial networks.. *arXiv preprint arXiv,* p. 1511.06434.

Ramzan, F. et al., 2019. A Deep Learning Approach for Automated Diagnosis and Multi-Class Classification of Alzheimer's Disease Stages Using Resting-State fMRI and Residual Neural Networks.. *J Med Syst,* Volume 44, p. 37.

Raschka, S., 2018. Model evaluation, model selection, and algorithm selection in machine learning.. *arXiv preprint arXiv,* Volume 1811.12808.

Rawat, W. & Wang, Z., 2017. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Comput,* Volume 29, p. 2352–2449.

Raza, M. et al., 2019. Diagnosis and monitoring of Alzheimer's patients using classical and deep learning techniques.. *Expert Systems with Applications,* Volume 136, pp. 353-364.

Regulation, P., 2018. General data protection regulation.. *Intouch,* Volume 25.

Reiman, E. et al., 2012. Brain imaging and fluid biomarker analysis in young adults at genetic risk for autosomal dominant Alzheimer's disease in the presenilin 1 E280A kindred: a case-control study.. *The Lancet Neurology,* Volume 11(12), pp. 1048-1056.

Rejusha, T. & KS, V. K., 2021. Artificial MRI Image Generation using Deep Convolutional GAN and its Comparison with other Augmentation Methods.. *In 2021 International Conference on Communication, Control and Information Sciences (ICCISc),* Volume 1, pp. 1-6.

Reunanen, J., 2003. Overfitting in Making Comparisons Between Variable Selection Methods.. *J. Mach. Learn. ,* Volume 3, p. 1371–1382.

Reyes, M. et al., 2020. On the interpretability of artificial intelligence in radiology: challenges and opportunities.. *Radiology: Artificial Intelligence,* Volume 2(3), p. 190043.

Rieke, J. et al., 2018. Visualizing convolutional networks for MRI-based diagnosis of Alzheimer's disease.. *In Understanding and Interpreting Machine Learning in Medical Image Computing Applications,* Issue Springer, Cham, pp. 24-31.

Ripley, B., 1993. Statistical aspects of neural networks.. *Networks and chaos-statistical and probabalistic aspects,* pp. 40-123.

Rodriguez, J., Perez, A. & Lozano, J. A., 2009. Sensitivity analysis of k-fold cross validation in prediction error estimation.. *IEEE transactions on pattern analysis and machine intelligence,* Volume 32(3), pp. 569-575.

Rosenblatt, F., 1958. The perceptron: a probabilistic model for information storage and organization in the brain.. *Psychological review,* Volume 65(6), p. 386.

Ruder, S., 2016. An overview of gradient descent optimization algorithms.. *arXiv preprint arXiv,* p. 1609.04747.

Rumelhart, D., Hinton, G. E. & Williams, R. J., 1986. Learning representations by back-propagating errors.. *Nature,* Volume 323(6088), pp. 533-536.

S.A., K., 2000. Applications of artificial neural networksfor energy systems.. *Appl. Energy,* Volume 67, pp. 17-35.

Salvadori, E. et al., 2015. Development and Psychometric Properties of a Neuropsychological Battery for Mild Cognitive Impairment with Small Vessel Disease: The VMCI-Tuscany Study.. *Journal of Alzheimer's Disease,* Volume 43, p. 1313–1323.

Salvadori, E. et al., 2016. VMCI-Tuscany Study Group, 2016. Operationalizing mild cognitive impairment criteria in small vessel disease: the VMCI-Tuscany Study.. *Alzheimers Dement ,* Volume 12, p. 407–418.

Sánchez Fernández, I. et al., 2020. Deep learning in rare disease. Detection of tubers in tuberous sclerosis complex.. *PloS one,* Volume 15(4), p. e0232376.

Saratxaga, C. et al., 2021. MRI Deep Learning-Based Solution for Alzheimer's Disease Prediction.. *Journal of personalized medicine,* Volume 11(9), p. 902.

Saravanan, N., Sathish, G. & Balajee, J. M., 2018. Data wrangling and data leakage in machine learning for healthcare.. *International Journal of Emerging Technologies and Innovative Research,* Volume 5(8), pp. 553-557.

Sarkar, I., 2010. Biomedical informatics and translational medicine.. *Journal of translational medicine,* Volume 8(1), pp. 1-12.

Sarraf, S., Tofighi, G. & Initiative, A. D. N., 2016. DeepAD: Alzheimer's disease classification via deep convolutional neural networks using MRI and fMRI.. *BioRxiv,* p. 070441.

Selkoe, D. J. & Lansbury, P. J., 1999. Alzheimer's disease is the most common neurodegenerative disorder.. *Basic Neurochemistry: molecular, cellular and medical aspects,* Volume 6, pp. 101-102.

Selvaraju, R. et al., 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization.. *In Proceedings of the IEEE international conference on computer vision,* pp. 618-626.

Sheu, Y., 2020. Illuminating the Black Box: Interpreting Deep Neural Network Models for Psychiatric Research.. *Frontiers in Psychiatry,* Volume 11.

Shi, L. et al., 2018. Mapping the contribution and strategic distribution patterns of neuroimaging features of small vessel disease in poststroke cognitive impairment.. *J. Neurol. Neurosurg. Psychiatry,* Volume 89, p. 918–926.

Shin, H. et al., 2018. Medical image synthesis for data augmentation and anonymization using generative adversarial networks.. *In International workshop on simulation and synthesis in medical imaging,* Issue Springer, Cham, pp. 1-11.

Shorten, C. & Khoshgoftaar, T. M., 2019. A survey on image data augmentation for deep learning.. *Journal of Big Data,* Volume 6(1), pp. 1-48.

Simonyan, K., Vedaldi, A. & Zisserman, A., 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps.. *In In Workshop at International Conference on Learning Representations.*

Simonyan, K. & Zisserman, A., 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition.].. *arXiv:1409.1556 [cs.*

Singh, A., Sengupta, S. & Lakshminarayanan, V., 2020. Explainable deep learning models in medical image analysis.. *Journal of Imaging,* Volume 52, p. 6(6).

Sivaranjini, S. & Sujatha, C. M., 2019. Deep learning based diagnosis of Parkinson's disease using convolutional neural network.. *Multimedia Tools and Applications.*

Small, G. W. et al., 1997. Diagnosis and treatment of Alzheimer disease and related disorders: consensus statement of the American Association for Geriatric Psychiatry,. *the Alzheimer's Association, and the American Geriatrics Society.* , Volume 278 (16), pp. 1363-1371.

Smith, E., 2017. Clinical presentations and epidemiology of vascular dementia.. *Clin Sci,* Volume 131, p. 1059–1068.

Smith, S., 2002. Fast robust automated brain extraction. *Hum Brain Mapp,* Volume 17, p. 143–155.

Smith, S. et al., 2004. Advances in functional and structural MR image analysis and implementation as FSL. *Neuroimage,* Volume 23(Suppl 1, pp. S208-219.

Srivastava, N. et al., 2014. Dropout: a simple way to prevent neural networks from overfitting.. *The journal of machine learning research,* Volume 15(1), pp. 1929-1958..

Stevens, L. et al., 2020. Recommendations for reporting machine learning analyses in clinical research.. *Circulation: Cardiovascular Quality and Outcomes,* Volume 13(10), p. e006556.

Suk, H., Lee, S. W., Shen, D. & Initiative, A. D. N., 2014. Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis.. *NeuroImage,* Volume 101, pp. 569-582.

Suk, H.-I. & Shen, D., 2013. Deep learning-based feature representation for AD/MCI classification.. *in MICCAI International Conference on Medical Image Computing and Computer-Assisted Intervention,* Volume 16, pp. 583-590.

Suk, H., Shen, D. & Initiative, A. D. N., 2015. Deep learning in diagnosis of brain disorders.. *In Recent Progress in Brain and Cognitive Engineering,* pp. 203-213.

Szegedy, C. et al., 2015. Going deeper with convolutions.. *In Proceedings of the IEEE conference on computer vision and pattern recognition,* pp. 1-9.

Tan, C. et al., 2018. A survey on deep transfer learning. *In International conference on artificial neural networks Springer, Cham,* p. 270–279.

Tang, Z. et al., 2019. Interpretable classification of Alzheimer's disease pathologies with a convolutional neural network pipeline.. *Nature communications,* Volume 10 (1), pp. 1-14.

T. D., 2016. Tensorflow: A system for large-scale machine learning.. *In 12th {USENIX} symposium on operating systems design and implementation ({OSDI},* Volume 16, pp. 265-283.

Tessa, C. et al., 2019. [117] Tessa, C. et al. Central modulation of parasympathetic outflow is impaired in de novo Parkinson's disease patients.. *PLOS ONE,* Volume 14(1), p. e0210324.

Thibeau-Sutre, E. et al., 2021. ClinicaDL: an open-source deep learning software for reproducible neuroimaging processing..

Thomas, A., Ré, C. & Poldrack, R. A., 2021. Challenges for cognitive decoding using deep learning methods.. *arXiv preprint arXiv,* p. 2108.06896.

Tufail, A., Ma, Y. K. & Zhang, Q. N., 2020. Binary classification of Alzheimer's disease using sMRI imaging modality and deep learning. *Journal of digital imaging,* Volume 33(5), pp. 1073-1090.

Ünal, M. O., 2019. *Mehmet Ozan Ünal.* [Online]
Available at: https://mozanunal.com/2019/11/img2sh/

Valliani, A. & Soni, A., 2017. Deep residual nets for improved Alzheimer's diagnosis.. *In Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics,* pp. 615-615.

Varma, S. & Simon, R., 2006. Bias in error estimation when using cross-validation for model selection.. *BMC bioinformatics,* Volume 7(1), pp. 1-8.

Vasquez, B. P. & Zakzanis, K. K., 2015. The neuropsychological profile of vascular cognitive impairment not demented: a meta-analysis.. *J Neuropsychol,* Volume 9, p. 109–136.

Vemulapalli, R., Van Nguyen, H. & Zhou, S. K., 2017. Deep Networks and Mutual Information Maximization for Cross-Modal Medical Image Synthesis.. *In Deep Learning for Medical Image Analysis,* p. 381–403.

Vieira, S., Pinaya, W. H. & Mechelli, A., 2017. Using deep learning to investigate the neuroimaging correlates of psychiatric and neurological disorders: Methods and applications.. *Neurosci Biobehav Rev,* Volume 74, pp. 58-75.

Villemagne, V. et al., 2013. Amyloid β deposition, neurodegeneration, and cognitive decline in sporadic Alzheimer's disease: a prospective cohort study.. *The Lancet Neurology,* Volume 12(4), pp. 357-367.

Wang, H. & Zheng, H., 2013. Model Validation. Machine Learning.. *In Encyclopedia of Systems Biology,* pp. 1406-1407.

Wang, K. et al., 2017. Generative adversarial networks: introduction and outlook.. *IEEE/CAA Journal of Automatica Sinica,,* Volume 4(4), pp. 588-598.

Wang, S. et al., 2018. Classification of Alzheimer's disease based on eight-layer convolutional neural network with leaky rectified linear unit and max pooling.. *Journal of medical systems,* Volume 42(5), pp. 1-11.

Wang, S. et al., 2017. Automatic recognition of mild cognitive impairment from mri images using expedited convolutional neural networks.. *In International Conference on Artificial Neural Networks,* pp. 373-380.

Wang, S., Wang, H., Shen, Y. & Wang, X., 2018. Automatic recognition of mild cognitive impairment and alzheimers disease using ensemble based 3d densely connected convolutional networks.. *In 2018 17th IEEE International conference on machine le.*

Wardlaw, J. et al., 2013. Neuroimaging standards for research into small vessel disease and its contribution to ageing and neurodegeneration.. *Lancet Neurol,* Volume 12, p. 822–838.

Weingarten, M., Lockwood, A. H., Hwo, S. Y. & Kirschner, M. W., 1975. A protein factor essential for microtubule assembly.. *Proceedings of the National Academy of Sciences,* Volume 72(5), pp. 1858-1862.

Wen, J. et al., 2020. Alzheimer's Disease Neuroimaging Initiative, 2020. Convolutional neural networks for classification of Alzheimer's disease: Overview and reproducible evaluation.. *Medical image analysis,* Volume 63, p. 101694..

Winblad, B. et al., 2004. . Mild cognitive impairment--beyond controversies, towards a consensus: report of the International Working Group on Mild Cognitive Impairment.. *J. Intern. Med,* Volume 256, p. 240–246.

Winkler, J. et al., 2019. Association between surgical skin markings in dermoscopic images and diagnostic performance of a deep learning convolutional neural network for melanoma recognition.. *JAMA dermatology,* Volume 155(10), pp. 1135-1141.

Wu, C. et al., 2018. Discrimination and conversion prediction of mild cognitive impairment using convolutional neural networks.. *Quantitative imaging in medicine and surgery,* Volume 8(10), p. 992.

Yadav, S. & Shukla, S., 2016. Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification.. *In 2016 IEEE 6th International conference on advanced computing (IACC),* pp. 78-83.

Yagis, E. et al., 2021. Effect of data leakage in brain MRI classification using 2D convolutional neural networks. *Scientific reports,* Volume 11(1), pp. 1-13.

Yagis, E., De Herrera, A. G. & Citi, L., 2019. Generalization Performance of Deep Learning Models in Neurodegenerative Disease Classification.. *in 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM),* p. 1692–1698.

Yamashita, R., Nishio, M., Do, R. K. & Togashi, K., 2018. Convolutional neural networks: an overview and application in radiology.. *Insights into imaging,* Volume 9(4), pp. 611-629.

Yang, C., Rangarajan, A. & Ranka, S., 2018. Visual explanations from deep 3D convolutional neural networks for Alzheimer's disease classification. *In AMIA annual symposium proceedings. American Medical Informatics Association.,* Volume 2018, p. 1571.

Yang, J. a. Y. G., 2018. Modified convolutional neural network based on dropout and the stochastic gradient descent optimizer. *Algorithms*, pp. 11(3), 28.

Yegnanarayana, B., 2009. Artificial Neural Networks.. *New Delhi, India: PHI Learning Pvt. Ltd.*

Yin, C., Li, S., Zhao, W. & Feng, J., 2013. Brain imaging of mild cognitive impairment and Alzheimer's disease.. *Neural regeneration research,* Volume 8(5), p. 435.

Young, P. N. et al., 2020. Imaging biomarkers in neurodegeneration: current and future practices. *Alzheimer's research & therapy,* Volume 12(1), pp. 1-17.

Zaharchuk, G. et al., 2018. Zaharchuk, G., Gong, E., Wintermark, M., Rubin, D. and Langlotz, C.P., 2018. Deep learning in neuroradiology. *American Journal of Neuroradiology,* Volume 39(10), pp. 1776-1784.

Zeiler, M. & Fergus, R., 2014. Visualizing and understanding convolutional networks.. *In European conference on computer vision,* Issue Springer, Cham, pp. 818-833.

Zhang, G. et al., 2020. A method for the estimation of finely-grained temporal spatial human population density distributions based on cell phone call detail records. *Remote Sensing,* pp. 12(16), p.2572 .

Zhang, H. et al., 2007. High-dimensional spatial normalization of diffusion tensor images improves the detection of white matter differences: an example study using amyotrophic lateral sclerosis.. *IEEE Trans Med Imaging,* Volume 26, p. 1585–1597.

Zhang, Y. et al., 2021. Grad-CAM helps interpret the deep learning models trained to classify multiple sclerosis types using clinical brain magnetic resonance imaging.. *Journal of Neuroscience Methods ,* Volume 353, p. 109098.

Zhou, A. & Jia, J., 2009. Different cognitive profiles between mild cognitive impairment due to cerebral small vessel disease and mild cognitive impairment of Alzheimer's disease origin.. *Journal of the International Neuropsychological Society,* Volume 15(6), pp. 898-905.

Zhou, Y., Yu, F. & Duong, T., 2014. Multiparametric MRI Characterization and Prediction in Autism Spectrum Disorder Using Graph Theory and Machine Learning.. *PLoS One,* Volume 9.

Zhu, B. et al., 2018. Image reconstruction by domain-transform manifold learning.. *Nature*, Volume 555, p. 487–492.

Zhu, G. et al., 2019. Applications of deep learning to neuro-imaging techniques.. *Frontiers in neurology*, Volume 10, p. 869.

9. Appendix 1

Summary of the previous studies performing classification of neurological disorders using MRI and with clear data leakage.

Reference	Description
Gunawardena et al., 2017	*"The MRI scan produces a 3-dimensional (3D) model of the body. Performing image processing techniques in a 3D MRI film is hard. Therefore it is necessary to convert those 3D MRI films into a series of 2D images before doing any preprocessing […] Series of 2D images were pre-processed before feature extraction and classification […] Preprocessed images were further processed in order to achieve the best result. All the images which were to be input to the CNN model were resized into 160 x 160 dimension because different sizes may reduce the accuracy of the classification […] Afterward, the data set was shuffled. Then the data set has been divided (split) into training set and testing set with a ratio of 80/20 (80% for training and 20% for testing)."*
Hon & Khan, 2017	*"Typically, from a 3D MRI scan, we have a large number of images that we can choose from. In most recent methods, the images to be used for training are extracted at random. Instead, in our porposed method, we extract the most informative slices to train the network. For this, we calculate the image entropy of each slice." [...]* *"We used our entropy-based sorting mechanism to pick the most informative 32 images from the axial plane of each 3D scan. That resulted in a total of 6400 training images, 3200 of which were AD and the other 3200 were HC." [...]* *"5-fold cross-validation was used to obtain the results, with an 80% - 20% split between training and testing." [...]* *"in our method, there are total 6,400 images; a 5-fold cross-validation*

	(80% - 20%) split therefore results in a training size of 5,120).” [...]
Jain et al., 2019	*“Brain MR images are in NIfTI format. NIfTI images are volumetric (3D) images, therefore images that we have after pre-processing are all of size 256x256x256. These images comprise of 2D images called slices. Hence, we have 256 slices corresponding to each NIfTI image […] image entropy based sorting mechanism is used to take most informative slices in which image entropy for each slice was calculated and top 32 slices based on entropy value were selected of each subject […] Above steps of data processing results in a balanced data-set of 4800 (150 subjects x 32 slices corresponding to each subject) slices which contains 1600 CE, 1600 MCI, and1600 CN slices” [...]* *“Our balanced dataset of 4800 images is shuffled and split into training and test set with split ratio 80:20.”*
Khagi et al., 2019	*"We have used 28 Normal controls (NC) and 28 Alzheimer’s disease (AD) patients for classification, selecting 30 important slices from each patient. Once all the slices are collected, each model was trained, validated and tested in ratio of 6:2:2 on random selection basis."*
Sarraf et al., 2017	*“The preprocessed rs-fMRI time series data were first loaded into memory using neuroimaging package Nibabel (http://nipy.org/nibabel/) and were then decomposed into 2D (x, y) matrices along z and time (t) axes. Next, the 2D matrices were converted to lossless PNG format using the Python OpenCV (opencv.org). The last 10 slices of each time course were removed since they included no functional information. Also, any slices with sum of pixel intensities equal to zero were ignored. During the data conversion process, a total of 793,800 images were produced, including 270,900 Alzheimer’s and 522,900 normal control PNG samples […] The random datasets were labeled for binary classification, and 75% of the images were assigned to the training dataset, while the remaining 25% were used*

	for testing purposes." [...] *"The preprocessed MRI data were then loaded into memory using a similar approach to the fMRI pipeline and were converted from NII to lossless PNG format using Nibabel and OpenCV, which created two groups (AD and NC) × four preprocessed datasets (MRI 0, 2, 3, 4). Additionally, the slices with zero mean pixels were removed from the data […] This step produced a total number of 62,335 images, with 52,507 belonging to the AD group and the remaining 9,828 belonging to the NC group per dataset […] Next, the model was trained and tested by 75% and 25% of the data"*
Wang et al., 2017	Note that in addition to slice-level split, significant data leakage could come from the way augmentation is implemented in this paper. For example, a slice could end up in the training set and a slightly brighter copy of it in the test set. *"In this work, we employ the following data augmentation techniques: brightness augmentation, horizontal and vertical shifts, shadow augmentation and flipping." [...]* *"The selected dataset includes serial brain MRI scans from 400 individuals with MCI (age: 74.8±7.4years, 257 Male/143 Female), and 229 healthy elderly controls (age: 76.0±5.0years, 119 Male/110 Female)[…] After data augmentation, we obtain 8000 images including 4000 images of MCI and 4000 images of healthy control. We extract 5000 images for training, 1500 images for validation, 1500 images for testing."*
Puranik et al., 2018	*"After the conversion of images to the JPEG format, the last 5 frame images from each time course were scraped as it didn't specifically denote any significant characteristic of the brain. Moreover, the images that were removed were complete black, and would only contribute as noise to the CNN. This generated 474,320 images in all of which 154,000 were*

	Alzheimer disease prone images, 209,440 were normal and 110,880 comprised of EMCI images. These images were pooled together and then randomly shuffled for bifurcation into training and testing dataset in the ratio of 85% and 15% respectively"
Basheera et al., 2019	*"The CNN is used for classification. In our article, we used 224 x 224-sized gray segmented images as input to the CNN." [...]* *"Our total data set has 18,017 GM segmented images. We shuffled and split the data set in the ratio 80:20 as training and test data sets." [...]*
Nawaz et al., 2020	*"Every 3D MRI image contains 256 256 166 slices per volume which cannot be fed to a 2D CNN model. Therefore, we have rescaled each 3D MRI volume and have converted it into 2D slices each of size 300 300 with a single channel for each plane (axial, coronal, sagittal). Each patient contains around 690± 2D slices which can be further fed to train the 2D-CNN model. The pre-processed slices of 3D images are shown in Fig. 1 during different stages." [...]* *"In this paper, we have used 3D structural MRI scans of 160 patients (52 NC, 62 MCI, and 45 AD) to train our 2D-CNN model. The unbalanced (a total of 67413) 2D images are used as a dataset which includes 20972 images for AD class, 26192 images for MCI, and 18513 for NC class. Networks are trained from scratch on data for 70 epochs with a batch size of 100. Experiments are performed using 60% data for training, 20 % for testing, and 20% for the validation set."* Please note that in Table 1 the number of images has been reported.

10. Appendix 2

Summary of the previous studies performing classification of neurological disorders using MRI and suspected to have potential data leakage.

Reference	Description
Farooq et al., 2017	*"MRI scans are provided in the form of 3D Nifti volumes. At first, skull stripping and gray matter (GM) segmentation is carried out on axial scans through spatial normalization, bias correction and modulation using SPM-8* tool. GM volumes are then converted to JPEG slices using Python Nibabel package. Slices from start and end which contain non information are discarded from the dataset".* Paragraph III.A. *"A subject is scanned at different point of times in different visits, i.e., baseline, after on two and three years. Each such scan is considered as a separate subject in this work. The dataset consists of 33 AD, 22 LMCI, 39 MCI patients and 45 healthy controls which makes a total 355 MRI volumes. Augmentation is done by simply flipping the image along horizontal axis. The balances set includes a total of 9506 images for each class, and a total of 38024 images for all classes". [...]* *"All experiments are performed by splitting data into 25% as test and 75% as train data. 10% data from train set is used as validation set".*
Ramzan et al., 2019	*"After applying the preprocessing methods on fMRI data, preprocessed 64×64x48x140 4D fMRI scans are obtained in which each scan contains 64×64x48 3D volumes per time course (140 s). These 4D scans are then converted to 2D images along with image height and time axis. This results in 6720 images of size 64x64 per fMRI scan. The first and last three slices are removed as they contain no functional information. Therefore, from each scan information*

	from 44 slices is used. Hence, 6160 2D images are obtained from each fMRI scan and are saved in portable network graphics (PNG) format. The data acquired from ADNI is processed and converted to 2D images by using the aforementioned pre-processing methods. In this way, we have created a dataset that was used for training deep learning networks." […] "*In the dataset, there are 138 4D scans and 850,080 2D images. For the evaluation, we split the dataset into a training dataset, validation dataset and testing dataset with 70%, 20%, and 10% split ratio, respectively as described in Table 6. The dataset was randomly shuffled before splitting.*" Please note that in Table 6, the number of images rather than the number of subjects has been reported for the training, validation, and testing dataset.
Raza et al., 2019	*"We used the AlexNet model that takes a 2-d image as an input whereas our brain MRI data is 3-d. Data permutation is used in which multiple slices (Central 20 slices) are extracted from MRI brain data to increase training samples." [...]* *"split ratio for training and test data is set to 0.8 in the experiment. In each plane of OASIS dataset, the number of images for training and testing the classifier are 6656 and 1664 respectively. Similarly, for each plane in ADNI dataset, the number of images for training and testing the classifier is 34912 and 8728 respectively."*
Pathak et al., 2020	*"In our work, we have converted MRI samples into JPEG slices in MATLAB tool. Pixel size of each sample is reduced to 8-bit from 14-bit size by rescaling to 255." [...]* *"Dataset consists of 110 AD, 105 MCI and 51 NC subjects, where each subject contains 44–50 sample of images. Out of which 110 AD subjects are collected from Horizon imaging center [17]. There are*

	total of 9540 images used for training the network and 4193 images for testing. Data augmentation on images is done with rescale operation." [...] *"We have conducted four experiments of our dataset. For two experiments, as shown in Table 4, 70% of the data was used for training and 30% for validation."* Please note that in Table 4 the number of images rather than the number of subjects has been reported for training and validation. *"Remaining two experiments are conducted with our dataset by removing some blank and unwanted images. In this, 75% of the reduced data was used for training and 25% for validation for remaining two experiments are shown in Table 5."* Please note that also in Table 5 the number of images rather than the number of subjects has been reported for training and validation.
Libero et al., 2015	We suspect that feature selection was performed on the whole dataset, before the application of the ML validation scheme. *"Nineteen high-functioning adults with ASD (15 males/4 females; mean age: 27.1 years) and 18 typically developing (TD) peers (14 males/4 females; mean age: 24.6 years) participated in this multimodal neuroimaging study (see Table 1 for demographic information)." […]* *"Groups were compared on the resulting cortical thickness values using ANCOVAs conducted using SPSS 22.0 software. Age was used as a covariate for all between-group analyses, as well as average hemispheric cortical thickness." […]* *"1H-MRS ratios were compared using ANCOVA, covarying for age, and GM content." […]* *"To compare the ASD and TD groups on FA, RD, MD, and AD, t-tests were conducted point-wise along each fiber tract for 100 points.*

	A permutation based multiple comparison correction was applied to determine statistical significance (Nichols & Holmes, 2002), p < .05." *"Leave-one-subject-out cross validation was performed for both regression and classification." […]* *"The data points included were the significant resulting values of the statistical analyses of separate neuroimaging modalities."*
Zhou et al., 2014	We suspect that feature selection was performed on the whole dataset, before the application of the ML validation scheme. *"To reduce possible classifier overfitting and improve generalization, feature selection was performed in two steps. First, principal component analysis was used to decompose the covariance matrix of the imaging features using the singular value decomposition program in Matlab (release 2010b; MathWorks, Natick, Mass) [33] after variance normalization. Then the number of sorted components based on singular values that contained 99% or 95% of the information from the covariance matrix of all features was determined. Finally, an advanced feature selection algorithm, based on mutual-information and integration of both mRMR criteria [34], was used to select imaging features based on the number of features (components) determined via principal component analysis."*
Sivaranjini, et al., 2019	*"The image dataset with 80% of the input data is used for training and the remaining 20% is used for testing. The number of images from each subject given to the deep learning model is averaged to be 40 ± 5 slices based on the selection criterion as shown in Table 2. These images are given to the subsequent convolution layers."* Please note that also in Table 2 the number of images rather than the number

	of subjects has been reported for training and testing.
Lui et al., 2014	We suspect that feature selection was performed on the whole dataset, before the application of the ML validation scheme. *"All original features are normalized by removing the mean of each feature and dividing by its SD. We used the feature selection procedure, mRMR, 24 to incrementally choose the most representative subset of imaging features, to increase relevance, and decrease redundancy." […]* *"We used 5 types of mainstream classifiers on the features chosen by mRMR: support vector machine (SVM), naive Bayesian, Bayesian network, radial basis network, and multilayer perceptron […] We also applied the above methodology to evaluate the achievable performance of different classifiers using the single best feature alone and for mRMR selected features."*
Hasan et al., 2019	*"Hasan and Meziane [2] refined these texture measures by ignoring the irrelevant features using analysis of variance method (ANOVA) and reduced to eleven texture measures for each co-occurrence matrix, namely, the contrast, the dissimilarity, the correlation, the sum of square variance, the sum variance, he sum average, the difference entropy, the inverse difference normalized (IDN), the information measure of correlation I (IMC1), the inverse difference moment normalized (IDMN) and the weighted distance in addition to the cross correlation. The total number of texture measures was reduced from 190 to 100 feature measures after using ANOVA." […]* *"In this study, a total of 6000 MRI axial slices from 600 patients (300 normal, and 300 abnormal) were collected […] The number of slices for each MRI scan is about 75 slices. […] The collected MRI dataset*

is adopted to validate the proposed method. Support vector machine (SVM) with 10-fold cross validation method are applied for accuracy rate estimation of the proposed method. The dataset is divided randomly into 10 folds that are roughly of equal size. Each MRI slice in the given dataset was normalized with 'zero- center' before submission to CNN."

11. Appendix 3

Summary of the previous studies performing classification of neurological disorders using MRI and that provide insufficient information to assess data leakage.

Reference	Description
Al-Khuzaie et al., 2021	Section 3 *"Table 1 demonstrates the number of MRI slices."* Section 4 "*The training data set was 75% and the validation data set was 25%."* Please note that also in Figure 3 the input is *"MRI slices dataset".*
Wu et al., 2018	*"Then, from among about 160 slices of raw MR scans of each subject, we discarded the first and last 15 slices without anatomical information, resulting in about 130 slices for each subject. Next, we selected 48 different slices randomly from the remaining slices with the interval of 4, and thus generated 16 RGB color images for each subject. Third, the selected slices were converted into portable network graphics (PNG) format. Finally, all of the RGB color images were resized to 256×256 pixels and converted to the Lightning Memory-Mapped Database (LMDB) for high throughput of the CaffeNet deep learning platform. To ensure the robustness of the model, five random datasets were created to repeat the training and testing of the CNN classifiers (5-fold cross-validation). The flow chart for this is shown as in Figure 4." [...]* *"Differential diagnosis of MCI" "According to aforementioned data augmentation, all baseline MR data were expanded to up to 7,200 slices (4,800 for training, 2,400 for testing) for 150 NC subjects, 7,200 slices (4,800 for training, 2,400 for testing) for 150 patients with sMCI, and 7,536 slices (5,024 for training,*

2,512 for testing) for 157 patients with cMCI. During the training model, embedded five-fold cross validation was employed to train a robust model."

12. Appendix 4

OASIS-200 is sub-sampled ten times by selecting 34 subjects (17 healthy controls (label=0) and 17 Alzheimer disease patients (label=1))

Sub-sample 1				Sub-sample 2				Sub-sample 3				Sub-sample 4				Sub-sample 5			
id	label	sex	age	id	label	sex	age	id	label	sex	age	id	label	sex	age	id	label	sex	age
27	0	F	82	8	0	F	89	97	0	F	60	9	0	M	89	95	0	M	61
40	0	F	78	54	0	F	73	58	0	F	73	74	0	M	69	77	0	M	68
63	0	M	71	97	0	F	60	68	0	M	71	15	0	M	87	15	0	M	87
24	0	F	83	65	0	F	71	74	0	M	69	53	0	M	74	7	0	F	90
50	0	F	74	13	0	F	88	2	0	F	91	46	0	M	75	6	0	F	90
53	0	M	74	37	0	F	80	46	0	M	75	28	0	F	81	68	0	M	71
78	0	M	68	12	0	F	88	88	0	M	64	91	0	M	62	46	0	M	75
21	0	M	84	43	0	F	76	26	0	F	82	25	0	F	83	34	0	F	80
80	0	F	67	58	0	F	73	56	0	F	73	75	0	F	69	44	0	F	75
89	0	F	64	27	0	F	82	18	0	M	86	76	0	F	69	26	0	F	82
71	0	M	70	30	0	F	81	19	0	F	85	97	0	F	60	81	0	F	67
29	0	F	81	64	0	F	71	49	0	F	74	61	0	F	72	73	0	F	69
23	0	F	84	2	0	F	91	75	0	F	69	13	0	F	88	98	0	F	59
85	0	F	65	89	0	F	64	87	0	F	64	66	0	F	71	71	0	M	70
59	0	F	73	53	0	M	74	72	0	F	70	51	0	F	74	40	0	F	78
49	0	F	74	80	0	F	67	61	0	F	72	50	0	F	74	22	0	M	84
91	0	M	62	85	0	F	65	0	0	F	94	27	0	F	82	52	0	M	74
135	1	F	80	188	1	M	68	173	1	F	72	182	1	M	70	146	1	M	78
148	1	F	77	138	1	M	79	176	1	F	71	115	1	M	84	134	1	F	80
128	1	M	81	194	1	M	66	188	1	M	68	138	1	M	79	179	1	F	71
191	1	F	67	163	1	M	73	127	1	F	81	132	1	F	80	170	1	F	72
192	1	F	66	161	1	M	74	116	1	F	83	154	1	F	76	122	1	M	82
198	1	F	63	162	1	F	73	197	1	M	64	117	1	F	83	149	1	F	77
144	1	F	78	169	1	F	73	198	1	F	63	124	1	F	81	165	1	F	73
163	1	M	73	113	1	F	84	118	1	F	83	111	1	M	86	183	1	M	70
104	1	M	90	189	1	F	67	168	1	M	73	198	1	F	63	172	1	F	72
140	1	F	78	187	1	M	69	134	1	F	80	161	1	M	74	141	1	F	78
186	1	F	69	198	1	F	63	147	1	F	78	160	1	F	74	175	1	F	72
197	1	M	64	192	1	F	66	106	1	M	88	135	1	F	80	154	1	F	76

187	1	M	69	179	1	F	71	157	1	F	75	106	1	M	88	112	1	F	84
130	1	M	80	119	1	F	83	107	1	F	87	101	1	F	92	150	1	M	77
172	1	F	72	101	1	F	92	146	1	M	78	125	1	M	81	195	1	F	65
157	1	F	75	121	1	F	83	196	1	M	64	129	1	F	80	168	1	M	73
152	1	M	77	183	1	M	70	181	1	M	70	152	1	M	77	124	1	F	81

(Continued)

Sub-sample 6				Sub-sample 7				Sub-sample 8				Sub-sample 9				Sub-sample 10			
id	label	sex	age	id	label	sex	age	id	label	sex	age	id	label	sex	age	id	label	sex	age
75	0	F	69	4	0	F	90	24	0	F	83	39	0	F	78	88	0	M	64
71	0	M	70	59	0	F	73	89	0	F	64	38	0	F	78	8	0	F	89
39	0	F	78	44	0	F	75	33	0	M	80	37	0	F	80	85	0	F	65
14	0	F	88	55	0	F	73	31	0	M	81	72	0	F	70	14	0	F	88
84	0	F	65	33	0	M	80	55	0	F	73	33	0	M	80	26	0	F	82
1	0	F	93	41	0	F	77	67	0	F	71	97	0	F	60	90	0	F	63
90	0	F	63	37	0	F	80	66	0	F	71	86	0	M	65	4	0	F	90
24	0	F	83	71	0	M	70	83	0	F	65	49	0	F	74	28	0	F	81
67	0	F	71	9	0	M	89	98	0	F	59	80	0	F	67	57	0	F	73
61	0	F	72	86	0	M	65	88	0	M	64	60	0	F	72	59	0	F	73
87	0	F	64	2	0	F	91	40	0	F	78	55	0	F	73	9	0	M	89
89	0	F	64	34	0	F	80	51	0	F	74	73	0	F	69	80	0	F	67
72	0	F	70	66	0	F	71	7	0	F	90	25	0	F	83	82	0	F	66
43	0	F	76	81	0	F	67	86	0	M	65	23	0	F	84	19	0	F	85
34	0	F	80	1	0	F	93	68	0	M	71	32	0	F	80	62	0	M	72
70	0	F	70	25	0	F	83	95	0	M	61	19	0	F	85	61	0	F	72
77	0	M	68	61	0	F	72	54	0	F	73	54	0	F	73	66	0	F	71
176	1	F	71	168	1	M	73	156	1	M	75	125	1	M	81	176	1	F	71
129	1	F	80	137	1	M	79	117	1	F	83	153	1	M	76	162	1	F	73
111	1	M	86	196	1	M	64	164	1	F	73	129	1	F	80	157	1	F	75
196	1	M	64	178	1	M	71	148	1	F	77	180	1	M	71	149	1	F	77
143	1	M	78	130	1	M	80	159	1	M	75	140	1	F	78	104	1	M	90
100	1	F	96	134	1	F	80	155	1	F	75	177	1	M	71	165	1	F	73
139	1	F	79	157	1	F	75	187	1	M	69	138	1	M	79	123	1	M	82
122	1	M	82	100	1	F	96	124	1	F	81	167	1	F	73	191	1	F	67
186	1	F	69	195	1	F	65	129	1	F	80	189	1	F	67	138	1	M	79
135	1	F	80	116	1	F	83	193	1	F	66	169	1	F	73	130	1	M	80
178	1	M	71	192	1	F	66	163	1	M	73	195	1	F	65	199	1	F	62
165	1	F	73	177	1	M	71	132	1	F	80	100	1	F	96	178	1	M	71
195	1	F	65	101	1	F	92	162	1	F	73	196	1	M	64	148	1	F	77
114	1	M	84	151	1	F	77	194	1	M	66	132	1	F	80	164	1	F	73
173	1	F	72	159	1	M	75	169	1	F	73	127	1	F	81	106	1	M	88

174	1	F	72	126	1	F	81	190	1	M	67	197	1	M	64	126	1	F	81
161	1	M	74	164	1	F	73	127	1	F	81	101	1	F	92	109	1	F	86

13. Appendix 5

Thirty-four subjects (17 AD and 17 HC) have been randomly sampled ten times to produce sub-sampled OASIS-34 datasets. The demographic features of each sub-sampled dataset are listed. Differences between AD and HC groups were assessed through a t-test and a χ^2-test for age and gender, respectively. The p-values are also reported.

OASIS subsample		AD patients	Healthy controls	p-value
Sample-1	Age (range, years)	62 – 84	63 - 90	
	Age (mean ± SD, years)	73.7 ± 7.0	74.0 ± 6.9	0.72
	Gender (women/men)	11/6	10/7	0.45
Sample-2	Age (range, years)	60 – 91	63 – 92	
	Age (mean ± SD, years)	76.0 ± 9.1	73.7 ± 7.6	0.02
	Gender (women/men)	16/1	10/7	0.22
Sample-3	Age (range, years)	60 – 94	63 - 88	
	Age (mean ± SD, years)	74.8 ± 9.3	75.1 ± 7.7	0.47
	Gender (women/men)	12/5	10/7	0.45
Sample-4	Age (range, years)	60 – 89	63 – 92	
	Age (mean ± SD, years)	75.2 ± 8.3	79.2 ± 6.6	0.49
	Gender (women/men)	11/6	9/8	0.07
Sample-5	Age (range, years)	59 – 90	65 – 84	
	Age (mean ± SD, years)	75.2 ± 9.0	75.3 ± 4.8	0.29
	Gender (women/men)	9/8	12/5	0.49

(continued…)

Sample-6	Age (range, years)	63 – 93	64 – 96	
	Age (mean ± SD, years)	73.1 ± 8.4	76.2 ± 7.8	0.05
	Gender (women/men)	15/2	10/7	0.15
Sample-7	Age (range, years)	65 – 93	64 – 96	
	Age (mean ± SD, years)	78.1 ± 8.3	76.5 ± 8.4	0.27
	Gender (women/men)	13/4	10/7	0.29
Sample-8	Age (range, years)	59 – 90	66 – 83	
	Age (mean ± SD, years)	71.9 ± 8.2	74.5 ± 5.2	1.00
	Gender (women/men)	11/6	11/6	0.15
Sample-9	Age (range, years)	60 – 85	64 – 96	
	Age (mean ± SD, years)	74.7 ± 6.8	75.9 ± 8.7	0.05
	Gender (women/men)	15/2	10/7	0.34
Sample-10	Age (range, years)	63 – 90	62 – 90	
	Age (mean ± SD, years)	75.8 ± 9.4	76.7 ± 7.1	0.24
	Gender (women/men)	14/3	11/6	0.38

AD = Alzheimer's disease; HC = Healthy controls; OASIS = Open Access Series of Imaging Studies; SD = standard deviation.

14. Appendix 6

Subject IDs and associated demographics for OASIS_200 dataset. The first 100 subjects are from the healthy control group (label = 0) and the last 100 subjects belong to Alzheimer disease patient group (label = 1). Age is in years. F, female; M, male; OASIS, Open Access Series of Imaging Studies.

OASIS_IDs	NIFTI_IDs	labels	sex	age		OASIS_IDs	NIFTI_IDs	labels	sex	age
221	0	0	F	94		278	100	1	F	96
270	1	0	F	93		400	101	1	F	92
284	2	0	F	91		447	102	1	F	92
65	3	0	M	90		226	103	1	M	90
83	4	0	F	90		247	104	1	M	90
299	5	0	F	90		273	105	1	F	89
301	6	0	F	90		31	106	1	M	88
445	7	0	F	90		137	107	1	F	87
19	8	0	F	89		179	108	1	F	87
32	9	0	M	89		28	109	1	F	86
197	10	0	F	89		351	110	1	M	86
271	11	0	F	89		440	111	1	M	86
169	12	0	F	88		35	112	1	F	84
176	13	0	F	88		161	113	1	F	84
342	14	0	F	88		223	114	1	M	84
260	15	0	M	87		304	115	1	M	84
363	16	0	M	87		53	116	1	F	83
157	17	0	F	86		122	117	1	F	83
317	18	0	M	86		123	118	1	F	83
201	19	0	F	85		286	119	1	F	83
254	20	0	F	85		290	120	1	M	83
110	21	0	M	84		380	121	1	F	83
186	22	0	M	84		16	122	1	M	82
428	23	0	F	84		23	123	1	M	82
75	24	0	F	83		84	124	1	F	81
113	25	0	F	83		158	125	1	M	81
146	26	0	F	82		164	126	1	F	81
426	27	0	F	82		352	127	1	F	81

13	28	0	F	81		441	128	1	M	81
106	29	0	F	81		21	129	1	F	80
228	30	0	F	81		42	130	1	M	80
337	31	0	M	81		134	131	1	M	80
33	32	0	F	80		166	132	1	F	80
138	33	0	M	80		267	133	1	M	80
180	34	0	F	80		329	134	1	F	80
244	35	0	F	80		335	135	1	F	80
330	36	0	F	80		373	136	1	F	80
446	37	0	F	80		60	137	1	M	79
206	38	0	F	78		263	138	1	M	79
259	39	0	F	78		339	139	1	F	79
280	40	0	F	78		52	140	1	F	78
64	41	0	F	77		185	141	1	F	78
338	42	0	M	77		217	142	1	F	78
195	43	0	F	76		268	143	1	M	78
220	44	0	F	75		287	144	1	F	78
234	45	0	M	75		308	145	1	F	78
423	46	0	M	75		399	146	1	M	78
1	47	0	F	74		425	147	1	F	78
10	48	0	M	74		233	148	1	F	77
165	49	0	F	74		238	149	1	F	77
212	50	0	F	74		315	150	1	M	77
241	51	0	F	74		388	151	1	F	77
354	52	0	M	74		405	152	1	M	77
365	53	0	M	74		15	153	1	M	76
62	54	0	F	73		402	154	1	F	76
279	55	0	F	73		82	155	1	F	75
326	56	0	F	73		205	156	1	M	75
355	57	0	F	73		272	157	1	F	75
369	58	0	F	73		424	158	1	M	75
404	59	0	F	73		452	159	1	M	75
139	60	0	F	72		240	160	1	F	74
237	61	0	F	72		418	161	1	M	74
332	62	0	M	72		3	162	1	F	73
170	63	0	M	71		124	163	1	M	73

203	64	0	F	71		210	164	1	F	73
216	65	0	F	71		291	165	1	F	73
255	66	0	F	71		312	166	1	F	73
341	67	0	F	71		374	167	1	F	73
398	68	0	M	71		451	168	1	M	73
449	69	0	F	71		454	169	1	F	73
85	70	0	F	70		56	170	1	F	72
256	71	0	M	70		115	171	1	M	72
371	72	0	F	70		269	172	1	F	72
112	73	0	F	69		298	173	1	F	72
199	74	0	M	69		316	174	1	F	72
293	75	0	F	69		432	175	1	F	72
422	76	0	F	69		67	176	1	F	71
130	77	0	M	68		155	177	1	M	71
343	78	0	M	68		288	178	1	M	71
356	79	0	F	68		411	179	1	F	71
68	80	0	F	67		430	180	1	M	71
303	81	0	F	67		39	181	1	M	70
438	82	0	F	66		120	182	1	M	70
30	83	0	F	65		142	183	1	M	70
133	84	0	F	65		453	184	1	F	70
322	85	0	F	65		22	185	1	F	69
358	86	0	M	65		73	186	1	F	69
78	87	0	F	64		390	187	1	M	69
135	88	0	M	64		300	188	1	M	68
292	89	0	F	64		98	189	1	F	67
70	90	0	F	63		307	190	1	M	67
114	91	0	M	62		382	191	1	F	67
457	92	0	F	62		66	192	1	F	66
109	93	0	F	61		94	193	1	F	66
455	94	0	F	61		143	194	1	M	66
456	95	0	M	61		184	195	1	F	65
72	96	0	F	60		46	196	1	M	64
200	97	0	F	60		243	197	1	M	64
289	98	0	F	59		362	198	1	F	63
372	99	0	M	59		41	199	1	F	62

15. Appendix 7

Subject IDs and associated demographics for ADNI dataset. The first 100 subjects are from the Alzheimer disease group (label = 1) and the last 100 subjects belong to the healthy control group (label = 0). Age is in years. ADNI, Alzheimer's Disease Neuroimaging Initiative; F, female; M, male.

ADNI_IDs	NIFTI_IDs	label	age	sex
5275	0	1	78	F
5006	1	1	68	F
4252	2	1	87	F
4338	3	1	81	M
4990	4	1	75	F
4756	5	1	84	M
5029	6	1	80	M
4954	7	1	61	M
4774	8	1	86	M
4195	9	1	62	M
4124	10	1	72	M
4672	11	1	67	M
5163	12	1	67	M
4615	13	1	87	M
5149	14	1	84	M
5087	15	1	65	F
5027	16	1	76	M

ADNI_IDs	NIFTI_IDs	label	age	sex
4075	100	0	73	M
4266	101	0	70	F
4348	102	0	66	F
56	103	0	78	F
4388	104	0	67	M
89	105	0	71	M
4739	106	0	65	M
4071	107	0	85	M
4150	108	0	74	M
416	109	0	82	F
4262	110	0	73	F
4083	111	0	85	M
4080	112	0	79	F
4545	113	0	67	F
23	114	0	78	M
4643	115	0	65	F
4382	116	0	76	F

4537	17	1	77	F
4039	18	1	56	M
4625	19	1	64	M
4879	20	1	80	F
5162	21	1	69	M
4732	22	1	77	M
4993	23	1	72	F
5013	24	1	68	F
4968	25	1	79	M
5206	26	1	85	M
4845	27	1	68	F
5016	28	1	64	F
4280	29	1	80	M
5090	30	1	59	M
5184	31	1	73	F
4024	32	1	56	F
4001	33	1	89	F
4905	34	1	73	F
4894	35	1	61	F
5070	36	1	71	M
5138	37	1	61	M
5205	38	1	59	F
4153	39	1	79	M

59	117	0	79	F
257	118	0	86	F
4093	119	0	70	F
4616	120	0	85	M
4345	121	0	70	M
677	122	0	81	M
4389	123	0	81	M
4393	124	0	74	M
4399	125	0	78	F
4313	126	0	77	F
4577	127	0	85	M
4032	128	0	70	F
4021	129	0	67	M
4082	130	0	76	M
4060	131	0	85	M
4339	132	0	84	M
4349	133	0	71	F
4277	134	0	72	F
4340	135	0	67	F
4208	136	0	78	M
4278	137	0	75	M
4391	138	0	75	M
4856	139	0	65	F

4728	40	1	82	M
5146	41	1	73	F
4982	42	1	58	F
4258	43	1	76	M
5208	44	1	69	M
4192	45	1	82	M
4740	46	1	88	M
4589	47	1	75	F
5019	48	1	63	F
5240	49	1	63	F
4949	50	1	78	F
5210	51	1	86	M
4853	52	1	71	F
5106	53	1	74	M
4223	54	1	76	M
5015	55	1	78	F
5071	56	1	76	M
4641	57	1	74	F
4172	58	1	76	M
4770	59	1	76	M
4783	60	1	83	M
4971	61	1	77	M
5123	62	1	73	F

4357	140	0	74	F
4158	141	0	84	M
4304	142	0	75	M
4104	143	0	72	M
4580	144	0	70	F
4448	145	0	64	F
4270	146	0	75	F
4795	147	0	61	M
842	148	0	79	M
4264	149	0	74	F
311	150	0	83	F
4086	151	0	82	M
4010	152	0	71	F
4367	153	0	65	F
4222	154	0	82	F
4386	155	0	85	F
5023	156	0	64	F
4218	157	0	81	M
4878	158	0	73	F
4120	159	0	82	F
4076	160	0	73	F
685	161	0	95	F
21	162	0	79	F

4863	63	1	70	M
4730	64	1	81	F
4719	65	1	79	F
4657	66	1	72	F
4549	67	1	79	M
4692	68	1	83	M
4997	69	1	61	F
4906	70	1	76	F
5054	71	1	74	F
4820	72	1	86	F
5252	73	1	57	M
4827	74	1	71	M
5005	75	1	78	M
4501	76	1	79	M
4912	77	1	69	F
4867	78	1	75	M
4546	79	1	71	M
4526	80	1	80	M
5241	81	1	88	M
5017	82	1	84	M
4110	83	1	79	F
4733	84	1	75	M
4792	85	1	80	M

4257	163	0	79	M
4291	164	0	76	F
4612	165	0	69	F
4559	166	0	67	F
4308	167	0	74	M
4762	168	0	74	M
454	169	0	89	F
4196	170	0	79	M
4084	171	0	68	F
555	172	0	87	M
4552	173	0	63	M
4505	174	0	80	F
4410	175	0	69	F
4200	176	0	70	F
4576	177	0	71	F
4320	178	0	71	F
4164	179	0	73	M
4173	180	0	70	F
4424	181	0	66	F
4043	182	0	82	M
4026	183	0	74	M
4453	184	0	66	M
4028	185	0	64	F

4696	86	1	73	F		4642	186	0	58	F
4209	87	1	78	F		69	187	0	81	M
5074	88	1	75	F		4092	188	0	82	F
5231	89	1	74	F		4511	189	0	70	M
4477	90	1	82	F		4491	190	0	84	M
4660	91	1	77	F		473	191	0	83	M
4859	92	1	72	M		210	192	0	83	F
5037	93	1	67	M		4041	193	0	78	F
5112	94	1	75	F		4014	194	0	81	M
4755	95	1	72	M		751	195	0	77	M
4772	96	1	79	F		4225	196	0	70	M
5018	97	1	73	M		498	197	0	80	M
5059	98	1	72	M		4037	198	0	76	M
4994	99	1	85	M		4337	199	0	72	M

16. Appendix 8

Subject IDs and associated demographics for PPMI dataset. The first 100 subjects are from the Parkinson's disease patient group (label = 1) and the last 100 subjects

PPMI_IDs	NIFTI_IDs	label	sex	age		PPMI_IDs	NIFTI_IDs	label	sex	age
3625	0	1	F	67		3515	100	0	F	74
3060	1	1	M	75		3468	101	0	M	57
3577	2	1	M	68		3809	102	0	F	53
3830	3	1	F	52		3277	103	0	M	66
3709	4	1	M	69		4010	104	0	M	42
3591	5	1	M	63		3216	105	0	F	52
3154	6	1	F	73		3350	106	0	M	79
3814	7	1	M	67		3390	107	0	M	66
3056	8	1	M	56		3544	108	0	M	70
3327	9	1	F	54		3527	109	0	M	62
3176	10	1	M	62		3851	110	0	F	54
3229	11	1	M	73		3959	111	0	M	73
3770	12	1	F	55		3464	112	0	M	51
4099	13	1	F	60		3767	113	0	F	53
4102	14	1	M	69		3257	114	0	F	53
4038	15	1	F	71		3480	115	0	F	72
4071	16	1	M	58		3952	116	0	F	69
3575	17	1	M	61		3967	117	0	M	57
3771	18	1	F	75		3270	118	0	M	55

3003	19	1	F	57		3424	119	0	F	64
3364	20	1	F	39		4116	120	0	M	65
3608	21	1	M	46		3000	121	0	F	69
3116	22	1	M	65		3016	122	0	M	57
3522	23	1	M	54		3779	123	0	M	56
3288	24	1	F	47		3813	124	0	M	65
3632	25	1	M	55		3806	125	0	F	59
3309	26	1	F	54		3029	126	0	M	66
3150	27	1	F	57		3526	127	0	M	61
3970	28	1	M	67		3619	128	0	F	32
3638	29	1	M	66		3151	129	0	M	58
3232	30	1	F	68		3114	130	0	F	64
3454	31	1	F	57		3301	131	0	M	52
3616	32	1	M	78		4004	132	0	F	65
3455	33	1	M	67		3479	133	0	M	58
3023	34	1	F	71		3570	134	0	M	72
3083	35	1	F	66		3853	135	0	M	47
3325	36	1	F	67		3636	136	0	M	64
3218	37	1	M	64		3161	137	0	M	45
3429	38	1	M	65		3310	138	0	M	65
3653	39	1	F	80		3201	139	0	F	65
3514	40	1	M	71		3013	140	0	F	79
3119	41	1	M	64		4104	141	0	M	66

3752	42	1	M	52		3074	142	0	M	31
4022	43	1	M	48		3053	143	0	M	69
4122	44	1	M	64		3611	144	0	F	42
3436	45	1	M	51		3478	145	0	M	77
3207	46	1	F	58		3169	146	0	M	57
3439	47	1	M	57		3215	147	0	F	70
3067	48	1	M	74		4079	148	0	M	63
3066	49	1	F	64		3157	149	0	F	64
3290	50	1	M	63		4090	150	0	M	57
3230	51	1	M	70		3428	151	0	F	58
3787	52	1	M	49		3206	152	0	F	31
4115	53	1	M	67		3368	153	0	F	53
3311	54	1	M	75		3355	154	0	M	32
3634	55	1	M	43		3405	155	0	F	64
3077	56	1	M	63		3160	156	0	M	80
3417	57	1	M	57		3361	157	0	F	56
3822	58	1	M	56		3613	158	0	F	56
4092	59	1	F	77		3320	159	0	M	56
3021	60	1	F	64		3411	160	0	M	41
4034	61	1	F	55		3519	161	0	M	74
3958	62	1	M	76		3008	162	0	F	82
4113	63	1	F	34		3969	163	0	F	81
3630	64	1	F	61		3358	164	0	M	49

3588	65	1	F	49		3362	165	0	F	42
3621	66	1	F	54		3219	166	0	M	70
3473	67	1	F	55		3759	167	0	F	54
3584	68	1	M	43		4085	168	0	M	67
3102	69	1	M	64		4032	169	0	M	68
3819	70	1	F	53		3551	170	0	M	64
3442	71	1	M	63		4118	171	0	F	68
3472	72	1	M	61		3615	172	0	M	66
4035	73	1	M	60		3965	173	0	M	83
3815	74	1	M	62		3064	174	0	F	60
3432	75	1	M	64		3057	175	0	F	60
3838	76	1	F	61		3807	176	0	F	73
4077	77	1	M	48		3075	177	0	M	76
3282	78	1	F	62		4139	178	0	M	81
3190	79	1	M	82		3466	179	0	M	48
3307	80	1	M	66		3410	180	0	M	74
3710	81	1	M	56		3523	181	0	M	64
3462	82	1	F	44		3768	182	0	M	60
3802	83	1	M	70		3651	183	0	M	77
3433	84	1	F	82		3004	184	0	M	59
3128	85	1	F	60		3115	185	0	M	61
3132	86	1	M	50		3855	186	0	F	49
3080	87	1	M	80		3156	187	0	M	70

3186	88	1	F	62		3453	188	0	F	60
4078	89	1	M	70		3525	189	0	M	56
3589	90	1	F	75		3852	190	0	M	77
3666	91	1	M	52		3071	191	0	M	72
3001	92	1	M	65		3521	192	0	M	65
3631	93	1	F	68		3955	193	0	M	54
3205	94	1	M	73		3656	194	0	M	79
3006	95	1	F	58		3554	195	0	M	75
3434	96	1	M	54		4105	196	0	M	67
3220	97	1	F	74		3859	197	0	M	60
3461	98	1	M	63		3817	198	0	M	74
3961	99	1	M	37		3457	199	0	F	63