Erasmus University Rotterdam

MSc. Econometrics and Management Science
Business Analytics and Quantitative Marketing
Master's Thesis

Convolutional Neural Networks for Multiclass Predictions of Alzheimer's Disease Progression

Thomas Michelotti
410568

April 23, 2020

Erasmus School of Economics

BIGR Biomedical Imaging Group Rotterdam

Supervisor: Prof. dr. P.J.F. Groenen	Supervisor: Dr. E.E. Bron
Second assessor: Dr. M. van de Velden	Supervisor: V. Venkatraghavan

The content of this thesis is the sole responsibility of the author and does not reflect the view of either Erasmus School of Economics or Erasmus University.

Abstract

Alzheimer's disease (AD) is a neurodegenarative disorder that is increasingly affecting the worldwide population. There are currently no disease-modifying treatments available, however, promising drugs are being tested in multi-cohort clinical trials. Accurate early-stage forecasts of the development of AD progression are helpful in improving cohort selection.

In this study, I investigate the performance of a multi-input convolutional neural network (CNN) for predictions of AD progression. I specifically focus on the generalisability of these forecasts to previously unseen individuals. The multiclass predictions distinguish between cognitively normal (CN) patients, patients with a mild cognitive impairment (MCI), and AD patients. The model is based on grey matter density maps of brain MRI scans and two non-image features, i.e. current diagnosis and follow-up time. One of the main challenges of predicting future diagnosis is to outperform a simple model that always predicts current diagnosis. I therefore introduce a custom loss function that differentiates between non-converters, converters to MCI, and converters to AD. Furthermore, the Grad-CAM tool is used to visualise model activations in the brain and the external performance of the methods is validated on a separate dataset.

This study shows that the progression of Alzheimer's disease can be predicted quite accurately based on a single MRI scan and corresponding current diagnosis. When converters are of interest, a custom loss function is beneficial for tackling the imbalance in the data. A model that uses such a custom loss function generalises quite well to predictions further in the future and to an external dataset. The Grad-CAM results show that model activations in the brain are not always consistent, however, the cerebellum appears to be an important brain region for MRI-based predictions of Alzheimer's disease progression.

Contents

1 Introduction **1**

1.1 Alzheimer's Disease . . . 1

1.2 Visualising Atrophy with Magnetic Resonance Imaging . . . 2

1.3 Predicting Alzheimer's Disease Progression . . . 3

1.4 Contributions . . . 5

2 Data **6**

2.1 ADNI . . . 6

2.2 Parelsnoer . . . 7

3 Methodology **7**

3.1 Introduction to Deep Learning . . . 8

3.1.1 Neural Networks . . . 8

3.1.2 Convolutional Neural Networks . . . 12

3.2 Preprocessing Steps . . . 14

3.3 Prediction Framework . . . 17

3.3.1 Main Methods . . . 17

3.3.2 Network Architectures . . . 19

3.3.3 Training Setup . . . 20

3.4 Performance Evaluation . . . 21

3.5 Visualisation with Grad-CAM . . . 22

3.6 Application to External Data . . . 23

3.7 Implementation . . . 23

4 Results **24**

4.1 Prediction Performance . . . 24

4.2 Visualisation with Grad-CAM . . . 26

4.3 Application to External Data . . . 27

5 Discussion **29**

5.1 Main Results . . . 29

5.2 Limitations and Future Work . . . 30

5.3 Conclusion . . . 33

A Supplementary ADNI Results **38**

B Supplementary Parelsnoer Results **39**

1 Introduction

1.1 Alzheimer's Disease

It has been over 100 years since the first patient was diagnosed with dementia by Alois Alzheimer (Alzheimer, 1907). He described "a peculiar disease" after observing abnormal shrinkage and deposits in and around nerve cells during a brain autopsy. Currently, over 50 million people are living with dementia. This number is expected to rise to 82 million in 2030 and to 152 million in 2050, equalling a global rate of one new case of dementia every 3 seconds (Patterson, 2018). Alzheimer's disease (AD) is the most prevalent type of dementia, covering about 50-75% of all cases (Prince et al., 2014). The global costs of dementia are estimated to be around US$1 trillion in 2018, which is expected to double by the year 2030 (Patterson, 2018). These costs can be divided into three main categories. Direct medical care accounts for roughly 20% of total costs, whereas direct social care (professional, residential, and nursing home care) and informal (unpaid) care each account for about 40% (Prince et al., 2015).

AD research ranges from methods that explore causes and genetics to research into effects and possible treatments of the disease. Some studies even focus on the social impact of AD, such as a 70,000 people survey regarding attitudes towards dementia (Alzheimer's Disease International, 2019). The aim of this subsection is to present a general introduction of AD. For a more extensive overview of AD research, I refer to the *Journal of Alzheimer's Disease*'s selection of 300 research reports that they consider the most impactful in the field of AD research since 2010. A comprehensive analysis consisting of three articles was conducted in 2017, reviewing these 300 reports. The first part focuses on pathology (causes and effects of the disease) (Hane et al., 2017a), the second on genetics and epidemiology (incidence, distribution, and possible control of the disease) (Robinson et al., 2017), and the third on diagnosis and treatment (Hane et al., 2017b).

Gaugler et al. (2019) describes the differentiation between several stages of AD. Preclinical Alzheimer's disease is the stage in which measurable changes can be identified in the brain, however no symptoms are developed. The next stage is called mild cognitive impairment (MCI). MCI is defined as a stage in which there are not only measurable changes in the brain, but patients also start to experience symptoms such as a cognitive decline that is greater than expected for someone their age. This neurological disorder is often divided into progressive MCI (pMCI), referring to patients who convert to AD after a certain time period, and stable MCI (sMCI), referring to patients who do not convert. The symptoms that patients experience in this stage do not impair their ability to function normally in daily life. The final stage is known as dementia due to Alzheimer's disease, or simply Alzheimer's disease. It is characterised by symptoms relating to an impairment of patients' memory, thinking, and behaviour. Unlike MCI symptoms, AD symptoms lead to significant interference with everyday activities.

There is currently no cure for AD or other types of dementia, despite substantial financial and intellectual investments. Recently, however, the US multinational biotechnology company Biogen has identified the anti-amyloid antibody *aducanumab* as the first drug to slow down Alzheimer's disease (Schneider, 2019). Their research also illustrates the importance of cohort selection in clinical trials for AD medication. Following an initial assessment of the trial in March 2019, the development of the drug was discontinued because of disappointing results. After reanalysing the results in October 2019, it appeared that the drug did in fact show a reduction in cognitive decline on a subgroup of the patients that continued in the studies for a longer time period. Accurate early-stage forecasts of the development of AD therefore have the potential to be helpful in improving cohort selection for clinical trials.

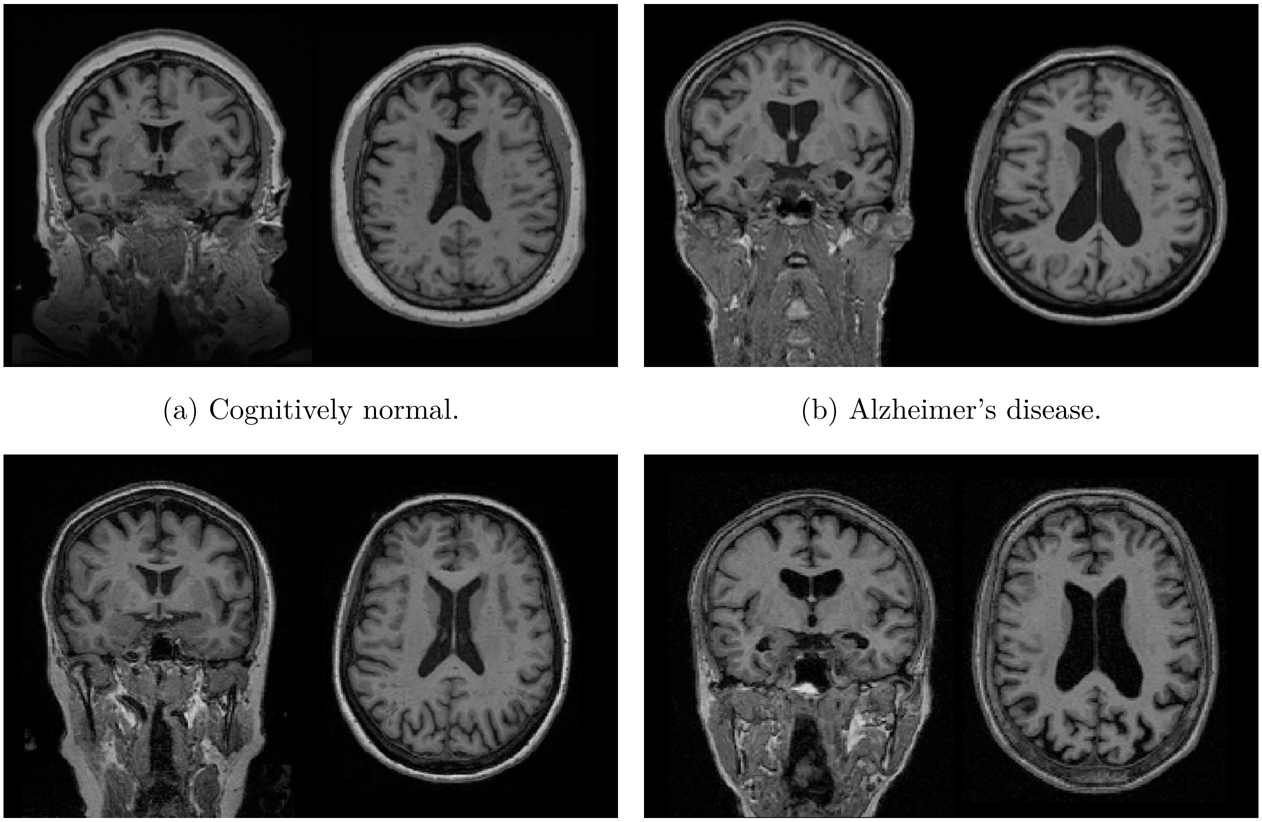
Another reason why predictions of AD progression are of great importance, is that an early diagnosis allows patients to get access to certain therapies that may improve their cognition. This does not only allow them to make important decisions about their future while they still have the capacity to do so, it can also help them to maintain their independence for longer. The early-stage prediction of AD progression therefore improves quality of life and delays institutionalisation, reducing direct social care costs (Prince et al., 2011).

1.2 Visualising Atrophy with Magnetic Resonance Imaging

Predictions of AD progression are often based on biomarkers. Biomarkers are indicators of a certain biological state that can be measured, sometimes even before clinical symptoms arise. For AD, biomarkers can be divided into measures of the amyloid-β (Aβ) protein and measures of neuronal injury and degeneration (Jack et al., 2012). These can be measured by either non-image methods or image modalities. Non-image measures can be taken from e.g. blood samples, samples of cerebrospinal fluid (CSF) obtained with a lumbar puncture, or cognitive tests. The most common image modalities are positron emission tomography (PET) and magnetic resonance imaging (MRI). MRI scans of the brain are 3D images from which the different brain tissues can be observed: grey matter (GM), white matter (WM) and CSF. GM consists of nerve cells, WM represents the fibers that connect these nerve cells and CSF is the liquid that surrounds the brain and spinal cord, of which the main function is to serve as a cushion for the brain and to absorb shocks for the central nervous system. During the progression of Alzheimer's disease, GM is mainly affected (Yang et al., 2010). This process is known as atrophy, commonly described as the shrinking of the brain. Atrophy can be visualised with MRI scans and quantified by measuring the volumes of the different brain tissues from these scans.

In order to illustrate how atrophy can be observed from MRI scans, Figure 1 shows coronal (front view) and axial (top view) cross sections of MRI scans of four different patients. The difference in volumes of brain tissues between CN and AD patients is often quite substantial. For example, the CN scan in Figure 1a clearly shows a relatively small black area (representing CSF) and a relatively large non-black area (representing WM and GM) compared to the AD scan in Figure 1b. This illustrates the vanishing of nerve cells that occurs when someone progresses to Alzheimer's disease.

The relative volumes of GM, WM, and CSF for MCI patients are expected to be somewhere in between those for CN and AD patients. This makes it harder to distinguish MCI from CN and MCI from AD than it is to distinguish CN from AD. Visually, the MCI scan in Figure 1c is very similar to the CN scan. The other MCI scan in Figure 1d seems to show similar relative volumes of brain tissues compared to the AD scan. This illustrates that the boundaries between CN and MCI as well as MCI and AD are generally hardly visible on MRI scans to the naked eye.



(a) Cognitively normal.

(b) Alzheimer's disease.

(c) Mild cognitive impairment.

(d) Mild cognitive impairment.

Figure 1: Four cross sections (coronal and axial) of structural MRI scans for a cognitively normal patient (a), a patient with Alzheimer's disease (b), and two patients with mild cognitive impairment (c), (d). All four patients were 70 year old females at the time of the scan.

An increase in publicly available major data sources, combined with an increase in computational power over the past decades, has made it possible for deep learning techniques (Section 3.1) to become particularly useful for predictions of disease progression based on MRI scans. The next subsection presents an overview of relevant literature in the context of forecasts of AD progression.

1.3 Predicting Alzheimer's Disease Progression

A first question that should be addressed, is the necessity of using deep learning for predictions of AD progression. Other types of models often have the advantage of easier interpretability. An extensive overview of different methods for modelling AD progression based on neuroimaging data is given in Ansart et al. (2019). Examples of successful methods that are not based on deep learning include hidden Markov models (HMM) (Sukkar et al., 2012), proportional hazard models (PHM)

(Lu, 2019), and support vector machines (SVM) (Jung et al., 2015). The downside of these types of models is that they are often based on non-image features (e.g. age, sex, and cognitive test scores) or on biomarker data from predefined regions of interest (ROI) in the brain. While these features contain predictive power for modelling AD progression, approaches based on full MRI scans generally outperform ROI-based methods (Cuingnet et al., 2011). Many methods can make use of data from complete MRI scans, however the advantage of deep learning is that such a model can take the spatial structure of the images into account. Additional non-image features can easily be added to deep learning architectures in order to create multi-input models.

Another relevant distinction concerns the difference between models for classification of current diagnosis and models for prediction of future diagnosis. Research into AD classification often focuses on different tasks. One of those is to distinguish CN patients from AD patients. This is a relatively easy task, as these two groups are usually quite different. State of the art CN-AD classification models achieve accuracies of over 95% (Basaia et al. (2019); Jain et al. (2019)). A more challenging task is to classify MCI patients as either patients that progress to AD (pMCI) or patients that remain stable (sMCI). MRI scans of these two groups are a lot more similar, which makes the classification task quite challenging. State of the art sMCI-pMCI classification models achieve accuracies of around 70-75% (Basaia et al. (2019); Cui et al. (2019)).

Next to these models for classification of current diagnosis, other types of models (including the ones in this study) focus on the task of predicting future diagnosis. Classification models can be trained on cross-sectional data, whereas predictive models require longitudinal data. In general, predicting future diagnosis comes with more uncertainty than classifying current diagnosis, which makes it a more challenging task. For multiclass predictions of future diagnosis (CN, MCI, AD), sMCI and pMCI patients are labelled equally as MCI. The difference between sMCI and pMCI patients is made in a different way than for the sMCI-pMCI classification task. The model has the task of predicting the future diagnosis of MCI patients as either MCI (which is similar to classifying sMCI) or AD (which is similar to classifying pMCI).

The performance of predictive models depends on, among other things, the time interval for which predictions are made and whether they are binary predictions (e.g. CN vs. AD) or multiclass predictions (CN vs. MCI vs. AD). To the best of my knowledge, there is no research of which the results can directly be compared to the results in this study. The most similar prediction framework comes from the TADPOLE challenge (Marinescu et al., 2018). This challenge involves the prediction of future diagnosis (CN, MCI, AD), which is evaluated on follow-up data that was not yet available at the time of the challenge. Marinescu et al. (2020) discusses the one year follow-up results of this challenge. The best-performing methods for the multiclass prediction of clinical diagnosis in the TADPOLE challenge yield a multiclass area under the receiver operating characteristic curve (MAUC) of around 93% and a balanced classification accuracy (BCA) of around 85%. I use the same evaluation framework for the methods in this study (Section 3.4).

Two remarks should be made about the results of the TADPOLE challenge. First, participants are allowed to use all previous information from individuals that are in the test set. The results therefore tell something about how well predictions of AD progression can be made for individuals that have an extensive history of hospital visits. It is, however, also relevant to investigate how well predictions can be made for previously unseen patients. The TADPOLE framework mimics this scenario with a separate challenge that concerns predictions based on only the most recent observation of every individual in the test set. The best-performing methods in this separate challenge yield an MAUC of around 90% and BCA of around 83%.

Second, it is important to note that the accuracy metrics in the TADPOLE challenge only reflect overall performance on the test set, without distinguishing between converters and non-converters. The main challenge in forecasting future diagnosis is to correctly predict patients that convert from their current diagnosis to a more advanced stage. If only 10% of a study population converts, a model that always predicts current diagnosis for future diagnosis obtains an accuracy of 90%, whereas none of the converters are predicted correctly. These converters are very much of interest, for example when the goal is to improve cohort selection for clinical trials in which potential medication is tested. Marinescu et al. (2020) acknowledges this limitation and mentions that there are only 18 converters in their test set. The TADPOLE results will be updated in the future, when more follow-up data is acquired and the number of converters will have increased.

1.4 Contributions

In this study, I develop and validate MRI-based models to predict AD progression. In clinical practice, longitudinal imaging data is rare; most patients only have one MRI scan available. I therefore specifically focus on methodology for cross-sectional data and only validate on individuals that are not used for training the models. The main question I try to answer with this research is: *How accurately can the progression of Alzheimer's disease be predicted based on a single MRI scan and corresponding diagnosis?* I formulate the following subquestions to help answering the main question and to provide guidance throughout this study:

Question 1 *How should the model address the imbalance between converters and non-converters?*

Question 2 *How does the performance of the model differ for short-term and long-term predictions?*

Question 3 *Which features are most important for predicting Alzheimer's disease progression?*

Question 4 *How well does the performance of the model generalise to external data?*

For Question 1, I consider three different models (Section 3.3.1). Question 2 is addressed by constructing these three models for three different time intervals. For Question 3, the Grad-CAM tool is used to visualise the features that the main method identifies as the most important for the task of predicting future diagnosis (Section 3.5). Finally, Question 4 is addressed by evaluating the performance of the three models on an external dataset (Section 3.6).

To the best of my knowledge, this is the first research in the domain of multiclass predictions of AD progression that simultaneously focuses on generalisability to data of previously unseen individuals, and on performance evaluation for converters and non-converters separately.

2 Data

2.1 ADNI

Data used in the preparation of this article is obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). The ADNI was launched in 2003 as a public-private partnership, led by Principal Investigator Michael W. Weiner, MD. The primary goal of ADNI has been to test whether serial magnetic resonance imaging (MRI), positron emission tomography (PET), other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimer's disease (AD).

The original ADNI dataset (ADNI 1, 2004-2009) contains a participant pool of 200 Alzheimer's disease patients, 400 patients with mild cognitive impairment and 200 cognitively normal controls. More data is collected in the follow-up programmes ADNI GO (2009-2011) and ADNI 2 (2011-2016). Currently, the ongoing cohort ADNI 3 further expands the ADNI database. This project was initiated in 2016 for a five-year time period. Inclusion criteria for the ADNI study are described in Petersen et al. (2010).

Since 2004, over 1000 scientific publications have made use of ADNI data. Furthermore, the data is often used for community challenges related to AD classification or prediction of AD progression. Some of these challenges from recent years are the *CADDementia challenge* (Bron et al., 2015), the *International challenge for automated prediction of MCI from MRI data* (Sarica et al., 2016), the *Alzheimer's Disease Big Data DREAM Challenge* (Allen et al., 2016), and the *TADPOLE challenge* (Marinescu et al., 2018). I use the evaluation framework of the TADPOLE challenge to evaluate the performance of the main methods in this study (Section 3.4).

In total, the ADNI dataset contains 7,819 MRI scans of 1,735 unique individuals. When making predictions, each of these MRI scans can be used to predict every available future diagnosis of the same individual. All matches of MRI scan and future diagnosis can be divided based on the amount of time in between them. The intervals for which predictions are made in this research are 0-1 year ahead, 1-2 years ahead, and 2-3 years ahead. Table 1 presents summary statistics of the ADNI data after applying the data preparation steps that are discussed in Section 3.2.

	0-1 year			1-2 years			2-3 years		
Unique subjects	1638			1349			933		
Unique MRI scans	5884			4564			2993		
Unique matches	7894			4876			3020		
	CN	MCI	AD	CN	MCI	AD	CN	MCI	AD
	28%	47%	26%	30%	42%	27%	31%	43%	26%
Conversions	611/7894 (8%)			789/4876 (16%)			673/3020 (22%)		
	MCI	AD		MCI	AD		MCI	AD	
	16%	84%		16%	84%		19%	81%	

Table 1: Summary statistics of the ADNI data for different future time intervals.

Table 1 shows that MCI is the most frequently occurring class in the ADNI data for the three different time intervals. There is no class that is largely underrepresented, so in that sense the data is quite well-balanced. There is, however, a substantial imbalance between converters and non-converters. For the 0-1 year interval, only 8% of the unique matches represents a conversion. Naturally, this percentage becomes larger when predicting diagnoses further in the future. Within the conversion group, another imbalance can be observed. Most of the conversions represent progression to AD, whereas less than a fifth of the conversions represent progression to MCI. This proportion is similar for all three time intervals.

2.2 Parelsnoer

Next to evaluating the different models on the ADNI data itself, I also evaluate the performance of the models on an external dataset. The Parelsnoer Institute (PSI) is a cooperation between the Dutch university medical centres that provides open-source datasets for academic research. One of the Parelsnoer datasets contains brain data including MRI scans. This dataset consists of baseline MRI scans of 556 individuals, 338 of which have at least one future diagnosis available that can be predicted. I divide the data in the same time intervals as the ADNI data. When a unique MRI scan can be matched to multiple future diagnoses within the same interval, one of these is randomly selected to be in the external test set (Section 3.6). Table 2 shows summary statistics of the Parelsnoer data, after splitting the observations in the three different time intervals.

	0-1 year			1-2 years			2-3 years		
Unique matches	263			250			129		
	CN	MCI	AD	CN	MCI	AD	CN	MCI	AD
	29%	33%	38%	29%	32%	39%	21%	25%	54%
Conversions	22/263 (8%)			37/250 (15%)			26/129 (20%)		
	MCI		AD	MCI		AD	MCI		AD
	23%		77%	19%		81%	19%		81%

Table 2: Summary statistics of the Parelsnoer data for different future time intervals.

In the Parelsnoer data, AD is the most common diagnosis for all three time intervals, as opposed to MCI in the ADNI dataset. For predictions 0-1 year ahead and 1-2 years ahead, the class distribution seems quite well-balanced. For predictions 2-3 years ahead, however, there seems to be a substantial class imbalance as more than half of the observations are AD. The imbalance between converters and non-converters, as well as the imbalance between conversions to AD and conversions to MCI, appears to be quite similar to that of the ADNI data for all three time intervals.

3 Methodology

I first introduce the concept of neural networks as a deep learning technique and the usefulness of convolutional neural networks for predictions based on (3D) images. I then address the preprocessing steps of the data, the prediction frameworks of the different methods, and the metrics for evaluating prediction performance. Next, I explain how Grad-CAM is used to visualise important features in the images and how the models are validated on the external Parelsnoer dataset. Finally, I briefly discuss the implementation of the methods including a link to my Github, which contains an explanation of the code and how that can be used to replicate the results in this study.

3.1 Introduction to Deep Learning

This subsection contains a general introduction on neural networks and, more specifically, convolutional neural networks. The aim of this subsection is to provide both the technical framework behind these kind of models, as well as some practical examples for illustration purposes. Furthermore, the main terminology in the context of (convolutional) neural networks is introduced.

3.1.1 Neural Networks

The core of neural networks is the same as for many models: take a number of features as input variables, find certain weights that are optimal according to some criterion, and output a set of predictions. For illustration purposes, imagine being given the task of classifying patients as CN, MCI, or AD based on their age and their test score for some cognitive assessment. The input for such a classification network consists of two nodes, one for each of the input features. The number of output nodes in the final layer is three, equal to the amount of labels. In between input and output, any amount of so-called hidden layers can be constructed, each of which can contain any amount of nodes. For this example, assume two hidden layers with three and five nodes respectively. Figure 2 shows a graph of such a neural network.

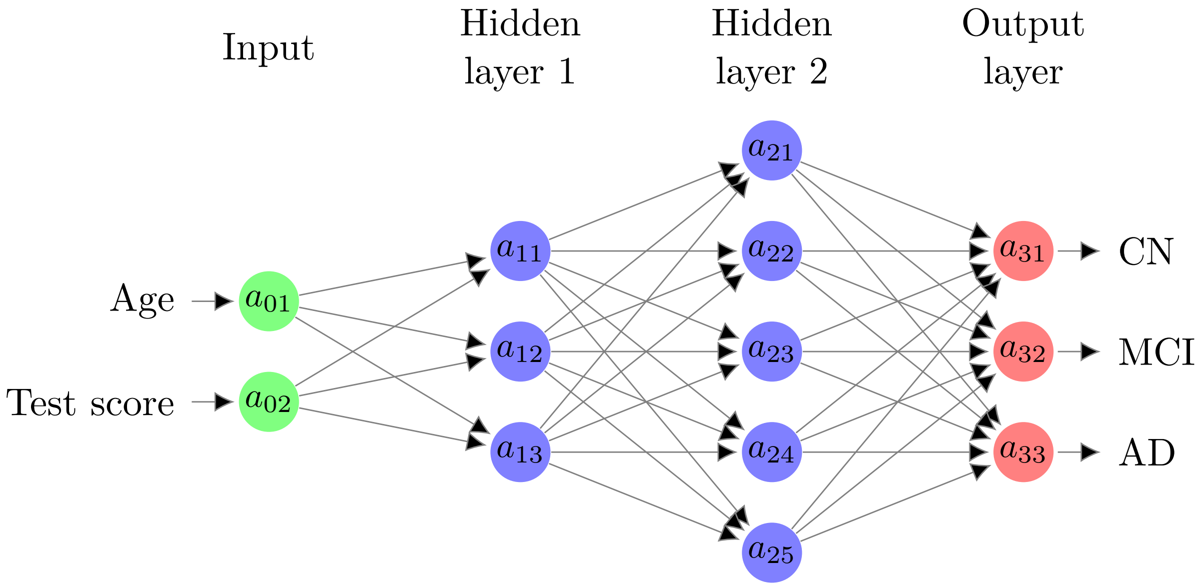


Figure 2: A three-layer neural network with two input features, two hidden layers with three and five nodes respectively, and a final layer with three output nodes.

Regular neural networks are characterised by fully connected layers. This means that all nodes in each layer are connected to all nodes in the previous layer. The idea is that some nodes get *activated* more than others during the training process, meaning that the model learns non-linear relations between certain combinations of input features and their corresponding label. The first hidden layer in Figure 2 computes its activations (a_{11}, a_{12}, a_{13}) as follows. Each new activation takes all previous activations as input (for the first layer, the previous activations are just the input features a_{01} and a_{02}). These are multiplied by certain weights, a bias term is added, and the result is passed through an activation function. The newly computed activations serve as input for the second hidden layer. There, the inputs are again multiplied by certain weights, bias terms are added, and an activation function computes the activations of the second hidden layer ($a_{21}, ..., a_{25}$). The final layer has a similar procedure, except that a different kind of activation function is used in order to compute relative probabilities for the target variables (a_{31}, a_{32}, a_{33}).

More generally, assume a network with L layers ($\ell = 1, ..., L$), K_ℓ nodes per layer ($k = 1, ..., K_\ell$), and C output classes ($c = 1, ..., C$). The ℓ^{th} layer of the network has input activation vector $\boldsymbol{a}_{\ell-1}$, output activation vector $\boldsymbol{a}_\ell$, weight matrix $\boldsymbol{W}_\ell$, and vector of biases (or in statistical terms, intercepts) $\boldsymbol{b}_\ell$. In matrix notation, these variables can be written as

$$\boldsymbol{a}_\ell = \begin{bmatrix} a_{\ell 1} \\ a_{\ell 2} \\ \vdots \\ a_{\ell K_\ell} \end{bmatrix}, \quad \boldsymbol{W}_\ell = \begin{bmatrix} w_{\ell 11} & w_{\ell 12} & \cdots & w_{\ell 1 K_{\ell-1}} \\ w_{\ell 21} & w_{\ell 22} & \cdots & w_{\ell 2 K_{\ell-1}} \\ \vdots & \vdots & \ddots & \vdots \\ w_{\ell K_\ell 1} & w_{\ell K_\ell 2} & \cdots & w_{\ell K_\ell K_{\ell-1}} \end{bmatrix}, \quad \boldsymbol{b}_\ell = \begin{bmatrix} b_{\ell 1} \\ b_{\ell 2} \\ \vdots \\ b_{\ell K_\ell} \end{bmatrix}, \tag{1}$$

where K_ℓ is the amount of nodes in the ℓ^{th} network layer and $K_{\ell-1}$ is the amount of nodes in the previous network layer. Before passing through an activation function, the intermediate results ($\boldsymbol{u}_\ell$) of the ℓ^{th} layer of the network are computed as

$$\boldsymbol{u}_\ell = \begin{bmatrix} u_{\ell 1} \\ u_{\ell 2} \\ \vdots \\ u_{\ell K_\ell} \end{bmatrix} = \boldsymbol{W}_\ell \boldsymbol{a}_{\ell-1} + \boldsymbol{b}_\ell. \tag{2}$$

The activations $\boldsymbol{a}_\ell$ are computed from the intermediate results by using an activation function. Examples of activation functions are the sigmoid function and the hyperbolic tangent function. The most popular choice for activation functions since its introduction by Glorot et al. (2011) is the *rectified linear unit* (ReLU). This activation function is defined as

$$\text{ReLU}(x) = \max(0,\, x). \tag{3}$$

The activations are thus computed from the intermediate results as $\boldsymbol{a}_\ell = \text{ReLU}(\boldsymbol{u}_\ell)$, or directly from the activations in the previous layer as $\boldsymbol{a}_\ell = \text{ReLU}(\boldsymbol{W}_\ell \boldsymbol{a}_{\ell-1} + \boldsymbol{b}_\ell)$. This holds for all layers, except for the final output layer. Here, rather than a ReLU activation, a function is used in order to obtain the desired output. In a classification context, the *softmax* function is a popular choice as its output can be interpreted as the relative probabilities of the output classes. In statistical terms, this function is usually referred to as the multinomial logistic function. The softmax function is computed separately for each intermediate result in the output layer as

$$\text{softmax}(u_{Lc}) = \frac{e^{u_{Lc}}}{\sum_{g=1}^{C} e^{u_{Lg}}}, \qquad c = 1, ..., C, \tag{4}$$

where L refers to the final layer in the network and C is the amount of nodes in the final layer, equal to the number of target variables.

Going back to the previous example and using the notation presented in Equations 1, 2, and 3, Figure 3 shows a more detailed graph of the computations within the first hidden layer of the network in Figure 2. This structure applies to all hidden layers. For the final layer, the ReLU activation is replaced by the softmax function (Equation 4).

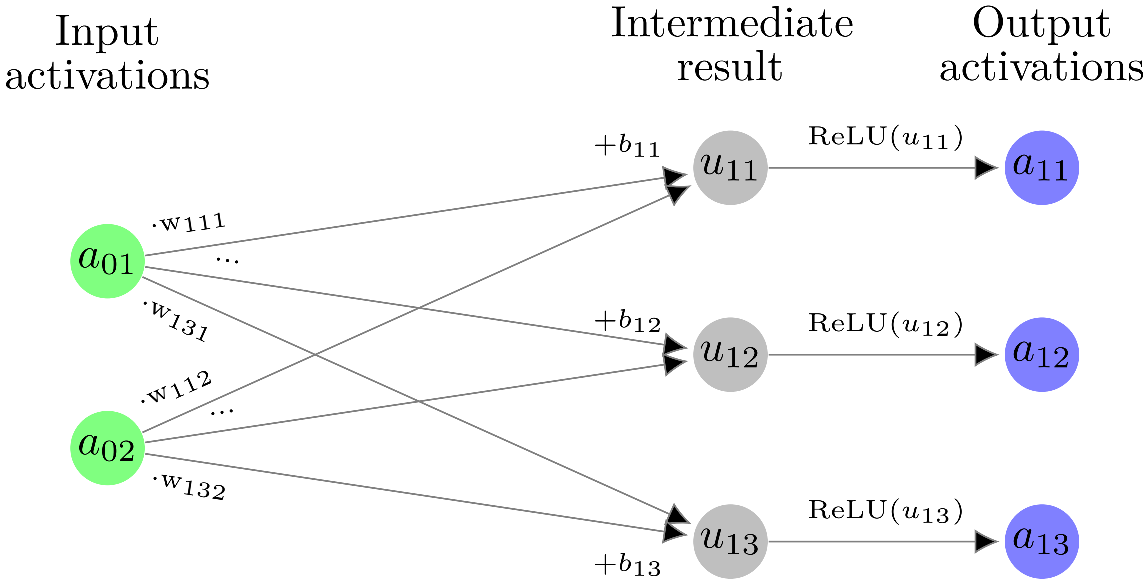


Figure 3: A forward pass through the first hidden layer of the network in Figure 2.

The process of multiplying by weights, adding bias terms, and passing through activation functions for all L layers is called a *forward pass* through the network. Suppose a network is trained on M training samples ($j = 1, ..., M$). Performing a forward pass through the network for all M training samples is called an *epoch*. When training a neural network, the goal is to find an optimal set of weights and biases. The definition of optimal is determined by the loss function of the model. A loss function is a measure of the fit of a model. The loss is lower when predicted labels match the true classes more closely. *Cross-entropy loss* is the most popular loss function in classification settings. In statistical terms, this is referred to as minus the log-likelihood of the multinomial logistic model. This loss function is defined as

$$L_{\text{cross-entropy}} = -\sum_{j=1}^{M}\sum_{c=1}^{C} I_{jc}\log(p_{jc}), \tag{5}$$

where M is the number of training samples, C is the number of classes, I_{jc} is the class indicator that equals one if sample j belongs to class c and zero otherwise, and p_{jc} is the estimated probability that sample j belongs to class c, which is the output of the softmax function in the final layer of the network. Before the first forward pass through the network, the weights are initialised randomly. After the first epoch, when the total loss of all training samples is computed, the network starts adjusting the weights by means of a procedure called *backpropagation*. During a *backward pass* through the network, the chain rule is used in order to compute the gradient with respect to the model parameters. The negative gradient corresponds to the direction of steepest descent in the loss function. The weights are updated by moving into the direction of steepest descent by a certain step size, which is equivalent to the *learning rate* of the model. A high learning rate means that the model is learning faster, however there is a higher risk of overshooting the optimal set of weights. A low learning rate avoids this risk, at the cost of a higher computation time. In practice, one often opts for a learning rate with momentum and/or decay over time. These types of algorithms usually start with a high learning rate and then converge to a lower learning rate when the combination of weights gets closer to the minimum of the loss function. A popular method that makes use of first and second order momentum is the *Adam algorithm* (Kingma and Ba, 2014).

Computing the average gradient over all M training samples before updating the weights is called batch gradient descent (BGD). BGD ensures a stable convergence as the weights are only updated once per epoch, using the average gradient over the entire training set. For a large dataset, however, model updates are slow and a lot of memory is required to fit all training samples. Alternatively, one can opt to choose a batch size b, $b < M$, and update the weights after computing the average gradient for each batch. The weights are updated M/b times per epoch. This process is known as *mini-batch gradient descent*, which is often preferred over BGD.

When the loss no longer decreases after a certain number of epochs, one can assume that the network has learned everything it can from the training samples. Sometimes, however, a model corresponds so closely to the training data that it does not generalise well to newly observed data. This phenomenon is known as *overfitting*. It is common practice to include regularisation techniques in a neural network to avoid overfitting. The most popular regularisation methods include the use of validation or holdout sets, L1 or L2 regularisation, and *dropout*. Dropout, as first introduced by Srivastava et al. (2014), involves the random deactivation of nodes in a network. For example, a dropout rate of 25% means that during every forward pass of one of the training samples, a random 25% of the nodes in the network gets deactivated. As a result, the network is forced to learn more general patterns from the input features, rather than overfitting on sample-specific noise.

Another common technique that is used in neural network architectures is *batch normalisation*. Normalisation of input features is common when there is a large difference between their ranges.

Similarly, the output of the activation function in every layer can be normalised by subtracting the batch mean and dividing by the batch standard deviation. This process is called batch normalisation and helps overcoming the problem of internal covariate shifts (Ioffe and Szegedy, 2015).

Coming back to the example in Figure 2, this network takes two non-image features as its input. Images, such as MRI scans, can be used as input features as well; the number of input nodes simply equals the amount of pixels or voxels of the images. The two main disadvantages of using a neural network with fully connected layers for image-based predictions are (i) the input is transformed into one large vector, meaning that no spatial structures within the images are taken into account, and (ii) the number of parameters gets out of hand. The next section explains how convolutional neural networks overcome both of these disadvantages.

3.1.2 Convolutional Neural Networks

Convolutional neural networks (CNN) are a subcategory of neural networks and are particularly useful for image-based predictive models. A CNN is characterised by convolutional layers, as opposed to fully connected layers. When images form the input of a neural network that consists of fully connected layers, the input is vectorised after which the network tries to learn the relevant pixels or voxels for a certain classification task. This can work quite well, however the spatial structure of the images is not fully employed due to the vectorisation of the input.

Convolutional layers have the valuable property that they take spatial structures within images into account, which I illustrate by means of the following example with 2D images. Suppose a network has the task of classifying handwritten single-digit numbers as 0, 1, ..., 9. The input of the network consists of 2D greyscale images of 128 × 128 pixels. Each pixel has a value between 0 and 255, where 0 corresponds to black and 255 corresponds to white. Figure 4 shows an example of a CNN with input images of size 128 × 128 pixels, three convolutional layers, and an output of size ten, representing the single-digit numbers.

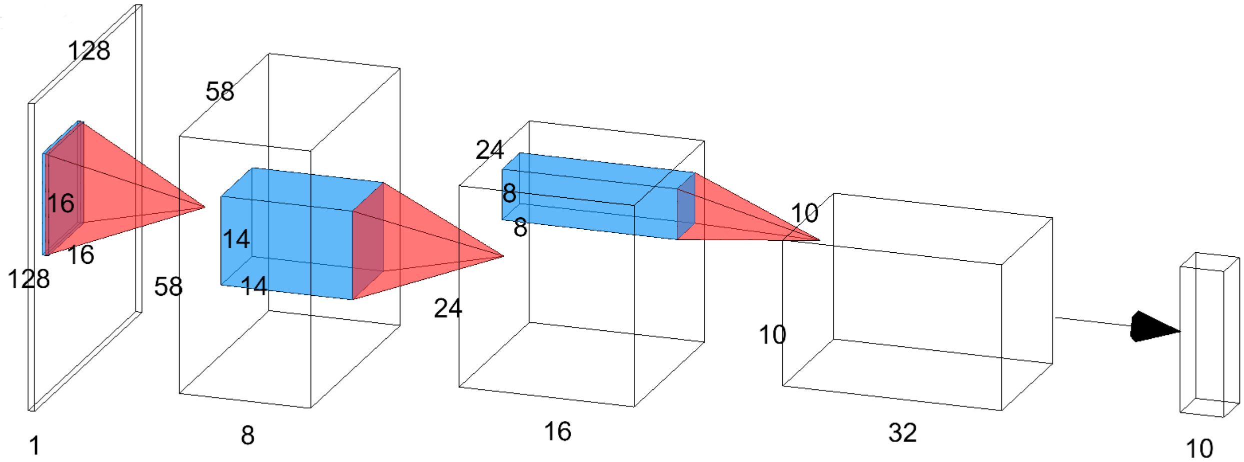


Figure 4: A schematic illustration[1] of a CNN with three layers and ten output classes.

[1] The tool for generating this figure is obtained from LeNail (2019).

Suppose the height, width and depth of the ℓ^{th} convolutional layer are denoted by H_ℓ, W_ℓ and D_ℓ respectively. The input images in Figure 4 have dimensions $H_0 = 128$, $W_0 = 128$ and $D_0 = 1$. The first convolutional layer consists of multiple filters that go over the input image. In this case, the filter has a filter size of 16 × 16. The filter starts at the top left of the image, where it covers the first 16 pixels in both horizontal and vertical direction. The adjacent pixels that are covered at any spatial position of the filter are called a *patch*, which is illustrated by the blue square of size 16 × 16. The filter consists of weights and a bias term, similar to fully connected layers in regular neural networks. At the top left of the image, the filter weights are multiplied by the pixels in that patch and a bias term is added, in order to compute a single output neuron.

The filter slides over the image such that it covers all pixels at least once. After the multiplication with the first patch in the top left corner of the image, the filter moves a certain amount of pixels to the right. This amount is called the *horizontal stride*. At every spatial position, the filter weights are multiplied by the patch and a bias term is added. When the filter reaches the edge on the right side of the picture, it starts sliding from left to right again. This time, however, the filter moves down a certain amount of pixels, which represents the *vertical stride*. This process continues until the bottom right of the input image is reached.

In the example, a single filter in the first convolutional layer produces an output of size 58 × 58 × 1. A convolutional layer usually consists of multiple filters, the output of which is concatenated in order to create the depth of the output of a convolutional layer. The first convolutional layer in this example consists of eight filters, which results in the output size of 58 × 58 × 8. For the second and third convolutional layer, the same process applies. The output of the previous convolutional layer serves as input for the next one. The only difference is that the depth of the input is now larger than one, meaning that an extra dimension is added to the convolutional filters.

In every convolutional layer, the filter size in combination with the horizontal and vertical stride determines the height and width of the output of that layer. It can happen that a certain combination of filter size and stride does not match the size of the input images. In that case, an arbitrary amount of zero values can be added around the edges of the input images. The process of adding zero values in order to achieve matching dimensions is called *zero padding*.

More generally, the dimensions of the output of a convolutional layer can be computed as follows. Suppose that for the ℓ^{th} convolutional layer, the filter size is $F_{\ell v} \times F_{\ell h}$ (vertical × horizontal), the vertical and horizontal strides are $S_{\ell v}$ and $S_{\ell h}$ respectively, the amount of zero padding is P_ℓ at all edges, and the amount of filters is B_ℓ. For the ℓ^{th} convolutional layer of which the input activation map is of size $H_{\ell-1} \times W_{\ell-1} \times D_{\ell-1}$, the output activation map has dimensions

$$\begin{aligned} H_\ell &= (H_{\ell-1} - F_{\ell v} + 2P_{\ell-1})/S_{\ell v} + 1, \\ W_\ell &= (W_{\ell-1} - F_{\ell h} + 2P_{\ell-1})/S_{\ell h} + 1, \\ D_\ell &= B_\ell. \end{aligned} \tag{6}$$

For example, the first convolutional layer in Figure 4 uses a filter size of 16 × 16, a horizontal and vertical stride of 2, zero padding of 1 pixel around the edges of the input images, and 8 filters. Using the formulas in Equation 6, this leads to the output dimensions $H_1 = (128 - 16 + 2)/2 + 1 = 58$, $W_1 = (128 - 16 + 2)/2 + 1 = 58$, and $D_1 = 8$, which corresponds to the output in the figure.

The output of the final layer of the network in this example consists of ten neurons, each one representing one of the single-digit numbers. The arrow pointing to the output vector in Figure 4 represents the output layer, which is often a softmax function. For 3D images such as MRI scans, the same principles apply as for the example with 2D images of handwritten numbers. The only difference is that 3D convolutional layers are used instead of 2D convolutional layers.

Next to regular (3D) convolutional layers, most CNN architectures employ *pooling layers*. The goal of a pooling layer is to reduce the number of parameters in the network. Pooling layers downsample the height and width of the output of the previous convolutional layer. Similar to convolutional layers, the output of a pooling layer is determined by its filter size and the horizontal and vertical stride of the filter. Examples of popular pooling methods are *max pooling* and *average pooling*. Figure 5 illustrates a 2D example of a max pooling layer with a filter size of 2 × 2 and a horizontal and vertical stride of 2.

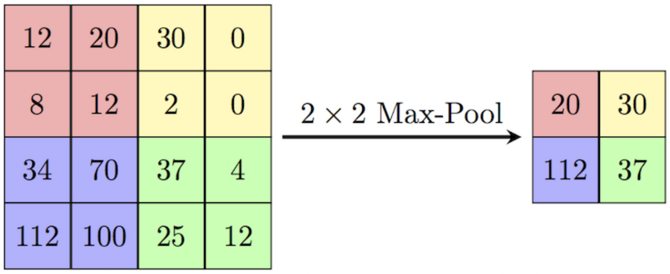


Figure 5: A 2D example of a max pooling layer with a filter size of 2 × 2 and a horizontal and vertical stride of 2.

As an alternative to pooling layers, regular convolutional layers can be used to reduce the amount of parameters as well. A larger stride in combination with a constant filter size leads to a smaller height and width of the output of a convolutional layer. This is referred to as a *convolutional pooling layer*. The difference with regular pooling layers is that the model introduces weights for every convolutional pooling layer, which are updated during backpropagation. Regular pooling layers do not have weights that can be optimised.

In conclusion, the main advantage of convolutional neural networks as opposed to regular neural networks with fully connected layers, is that convolutional filters take the spatial structure of the input into account. This is especially relevant for analysis based on (3D) images. Another advantage is that, compared to fully connected layers, convolutional layers need much fewer parameters in order to produce the same number of output activations.

3.2 Preprocessing Steps

Before implementing a CNN for predictions of AD progression, several preprocessing steps are applied. These can be divided into image preprocessing steps and data preparation steps.

Image Preprocessing

There are substantial differences in the sizes and shapes of brains across individuals. This makes it hard to apply MRI-based predictive models such as convolutional neural networks. It is important to directly compare the same brain areas of different individuals, while these are not perfectly aligned on the original MRI scans. It is therefore common practice to transform all MRI scans to the same template, without losing the relevant voxelwise information.

All images from both the ADNI and the Parelsnoer data are transformed to the same template. Rather than using a standardised template, a template is constructed based on a stratified random sample of 50 patients from the ADNI data. In short, the preprocessing of the images involves the following steps. First, the grey matter volume for every voxel is extracted from the MRI scans. Second, these grey matter density maps are transformed to a common template space using non-rigid registration. Third, the transformed grey matter density maps are multiplied by the Jacobian determinant of the deformation field, in order to restore the information of grey matter density per voxel. For a more detailed description of this preprocessing method, I refer to Bron et al. (2014). Figure 6 schematically illustrates the three preprocessing steps.

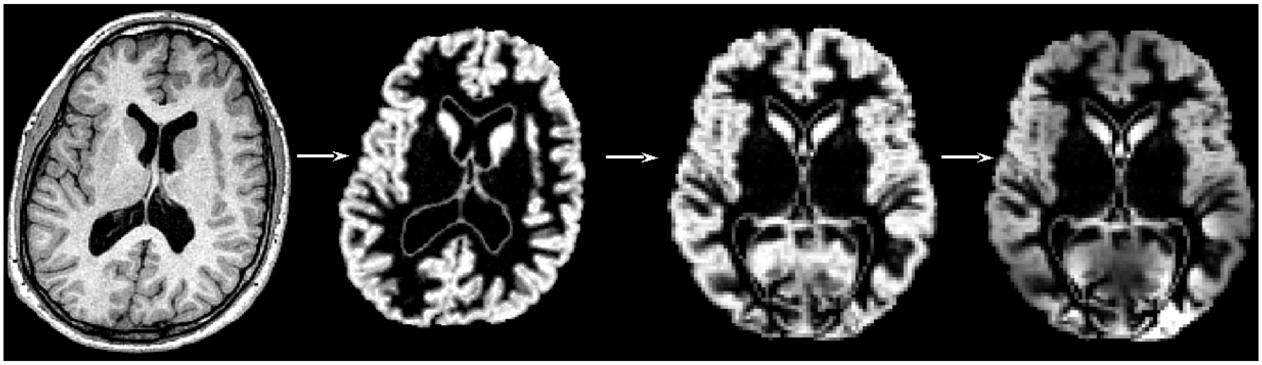


Figure 6: An illustration[2] of the three preprocessing steps of the MRI scans: (i) extraction of GM volume, (ii) transformation to a common template space, and (iii) multiplication by the Jacobian determinant of the deformation field.

After applying these transformations, all images are of size 143 × 179 × 148. In order to reduce the amount of model parameters, the images are downsampled by a factor of four to a size of 36 × 45 × 37. These downsampled images drastically reduce computation time compared to the full-size images.

Data Preparation

Next to the preprocessing of the images, this subsection addresses the data preparation steps. The goal of the models is to predict future diagnosis, rather than to classify current diagnosis. Therefore, every MRI scan is matched to every available future diagnosis of the same individual. These matches

[2] The image is retrieved from `https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FSLVBM`, see Douaud et al. (2007).

are then divided based on the amount of time in between the moment of the MRI scan and the moment of the future diagnosis. The three time intervals for which predictions are made are 0-1 year ahead, 1-2 years ahead, and 2-3 years ahead. These different time intervals are relevant for addressing Question 2 (Section 1.4).

In the ADNI data, every observation is already labelled based on its timing. Baseline observations are denoted by "bl" and follow-up observations are denoted by labels of fixed semi-annual length such as "m06", "m12", and "m18". This notation indicates the amount of months between baseline and follow-up visit, rounded to the nearest six months. During every follow-up visit of an individual, a new MRI scan and/or updated diagnosis can be made. For illustration purposes, suppose an individual is monitored for three years. Figure 7 shows the timeline of the baseline and follow-up visits of this individual, including the data that is collected during every visit.

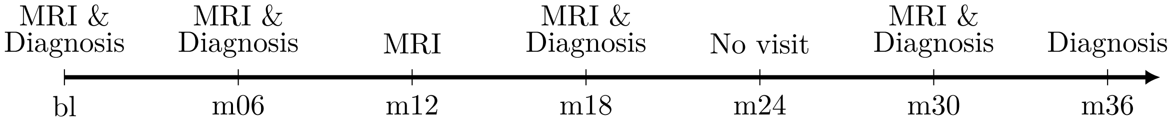


Figure 7: An example of data collection at baseline and at multiple follow-up visits.

Figure 8 displays the matches of MRI scan and future diagnosis that can be made for the three different time intervals using the example in Figure 7. This illustrates that the same MRI scan can be used in multiple intervals, and even multiple times within the same interval if it can be matched to multiple future diagnoses that fall in that interval. Sometimes, an observation consists of only an MRI scan or only a diagnosis. Whenever either the MRI scan, the diagnosis at the time of that MRI scan, or the future diagnosis is not available, that particular match is not included in the data. The reason for excluding matches for which the current diagnosis is not available, is that this feature is used as input for some of the models.

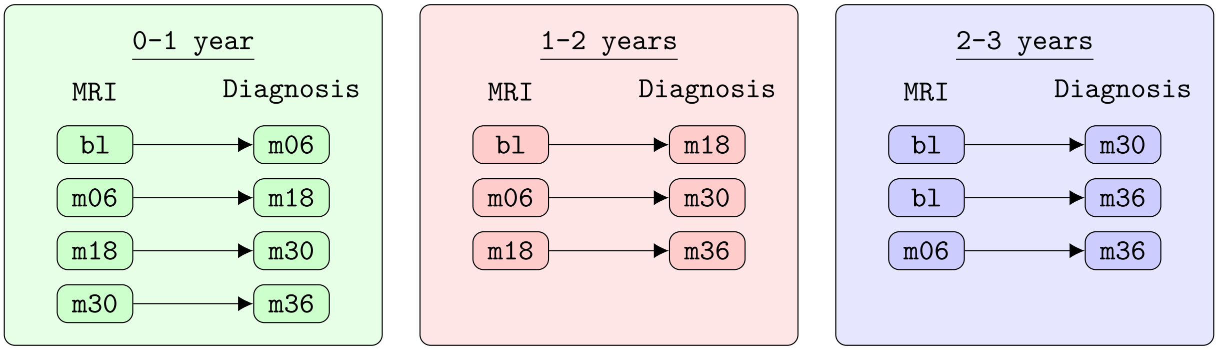


Figure 8: An illustration of how the matches of MRI scan and future diagnosis from the timeline in Figure 7 are divided over the three time intervals.

The registration of the semi-annual follow-up times is not always correct. I therefore calculate the exact follow-up time, based on the date of the baseline observation and the follow-up date. This exact follow-up time is used as input feature for some of the models. When the true follow-up time of an observation is more than three months shorter or longer than indicated by the follow-up time in the data, I move that observation to the time interval that is indicated by the exact follow-up time. Applying these preparation steps to the ADNI data leads to the summary statistics displayed in Table 1 in Section 2.1.

3.3 Prediction Framework

This subsection outlines the prediction framework of the methods in this study. I start by introducing the three methods that are used to answer the main research question and four subquestions. I then explain the network architectures of these three models and the training setup that is used.

3.3.1 Main Methods

Two sources of data are available for making predictions of future diagnosis, namely images and non-image features. The images are the MRI scans of the brain from the ADNI dataset, after applying the preprocessing steps that were discussed in the previous subsection. The two non-image features are the current diagnosis and the amount of time between the moment of the MRI scan and the moment of the future diagnosis. As a first benchmark, I construct a CNN that only takes the images as input. This model uses a regular cross-entropy loss function.

Model 1 *A single-input CNN based only on MRI scans with a cross-entropy loss function.*

This model is expected to yield similar results for converters and non-converters, as it does not take current diagnosis as an input feature. The overall performance, however, is expected to be relatively low because the model does not use all available information. I therefore construct a second benchmark that is based on both MRI scans and the two non-image features.

Model 2 *A multi-input CNN based on MRI scans and two non-image features with a cross-entropy loss function.*

Adding the non-image features to this model should improve overall prediction performance. However, as most observations in the training data are non-converters, the current diagnosis is a very strong predictor. With a regular loss function, the model could focus on this non-image feature too much. Model 2 is thus expected to yield a higher overall performance compared to Model 1. On the subgroup of converters, however, it is expected to yield a lower performance.

For the main method in this study, I introduce a multi-input CNN with a custom loss function. The custom loss function differentiates between converters and non-converters, which is relevant for addressing Question 1 (Section 1.4).

Model 3 *A multi-input CNN based on MRI scans and two non-image features with a custom loss function.*

As this model is able to make use of both image data and non-image features, the objective is to obtain a higher overall prediction performance compared to Model 1. Simultaneously, the custom loss function should ensure that the model does not focus too much on the non-converters. On the subgroup of converters, the model has the objective of outperforming Model 2.

The custom loss function consists of a penalty term that is added to the cross-entropy loss. The formula for the regular cross-entropy loss is multiplied by this penalty term in order to obtain the proposed custom loss function, which can be written as

$$L_{\text{custom}} = -\sum_{j=1}^{M}\sum_{c=1}^{C} I_{jc}\log(p_{jc}) \times \delta_j, \tag{7}$$

where δ_j represents the *converter penalty* of training sample j. The converter penalty is introduced to overcome both the imbalance between converters and non-converters and the imbalance between converters to MCI and converters to AD. This penalty ensures that the model finds converters to MCI and converters to AD equally important compared to non-converters, even though they appear less frequently in the training data. The converter penalty is defined as

$$\delta_j = \begin{cases} 1, & \text{if training sample } j \text{ is a non-converter} \\ \delta_{\text{AD}}, & \text{if training sample } j \text{ is a converter to AD} \\ \delta_{\text{MCI}}, & \text{if training sample } j \text{ is a converter to MCI} \end{cases}, \tag{8}$$

where δ_{AD} is the penalty for converters to AD and δ_{MCI} is the penalty for converters to MCI. These two penalty terms are calculated in the following manner. Suppose 90% of the training data represents non-converters and only 10% represents converters. The average penalty for converters should then be $90/10 = 9$ if they are to be treated equally to non-converters. Suppose now that 80% of the converters represent a conversion to AD and only 20% of the converters represent a conversion to MCI. This means that the penalty for converters to MCI should be $80/20 = 4$ times larger than for converters to AD, while the weighted average of the two penalty terms should be 9. The following two equations should hold:

$$0.80\delta_{\text{AD}} + 0.20\delta_{\text{MCI}} = 9, \tag{9}$$

$$0.25\delta_{\text{MCI}} = \delta_{\text{AD}}. \tag{10}$$

The system of equations that needs to be solved in order to compute the penalty terms for converters to AD and converters to MCI is

$$\begin{bmatrix} 0.80 & 0.20 \\ -1 & 0.25 \end{bmatrix} \begin{bmatrix} \delta_{\text{AD}} \\ \delta_{\text{MCI}} \end{bmatrix} = \begin{bmatrix} 9 \\ 0 \end{bmatrix}, \tag{11}$$

which results in $\delta_{\text{AD}} = 5.625$ and $\delta_{\text{MCI}} = 22.5$. The converter penalty thus takes into account the relative frequencies of converters to AD, converters to MCI, and non-converters in the training set, ensuring that the model focuses on these three groups equally.

3.3.2 Network Architectures

I first explain the network architecture of Model 1. The same architecture is used for Models 2 and 3, except that the two non-image features are added as additional model input.

A CNN is constructed for predictions of AD progression. The same architecture is used for the three different time intervals. The CNN consists of five main blocks. Each block consists of a 3D convolutional layer (filter size 3 × 3 × 3, stride 1), followed by dropout, batch normalisation (BN), and a ReLU activation function. This is succeeded by a 3D convolutional pooling layer (filter size 3 × 3 × 3, stride 2), dropout, BN, and a ReLU activation. The convolutional pooling layers reduce the number of parameters due to the stride of 2, however the parameters in these layers are trainable. The amount of filters in the convolutional and convolutional pooling layers is 8, 16, 24, 32, and 16 in the five blocks respectively. The input of the network consists of MRI scans, processed as described in Section 3.2, each represented as a 3-dimensional matrix of shape (36, 45, 37).

After the five blocks, there is a final 3D convolutional layer with 3 filters that is again followed by dropout, BN, and ReLU activation. The output of this layer is followed by a 3D global average pooling operation. The output layer of the network is a softmax activation function, providing three probabilities that add up to one. The CNN of Model 1 consists of 88,017 parameters in total. A schematic illustration of the CNN architecture of Model 1 is displayed in Figure 9.

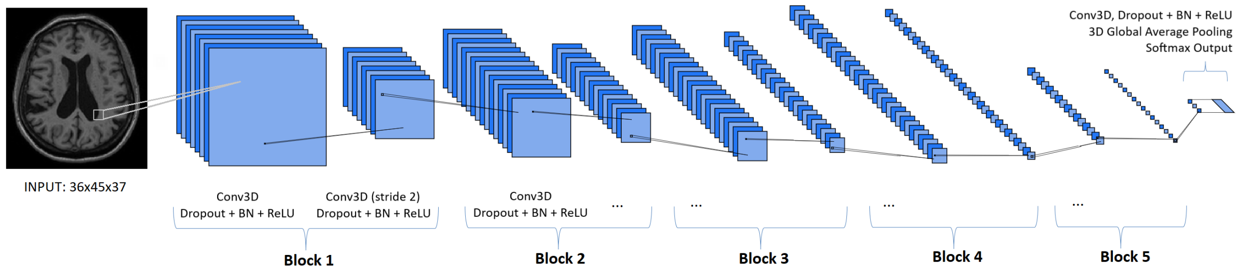


Figure 9: A schematic illustration[3] of the CNN architecture of Model 1.

For Models 2 and 3, two non-image features are added to the architecture of Model 1. These additional input features are concatenated to the output of the fifth block of the network in Figure 9. After concatenation, it is possible to add a single fully connected layer before the softmax output. This is similar to a linear regression with 18 variables: the 16 output nodes of the images and the two non-image features. A downside of this approach is that the CNN cannot learn more complex non-linear relations between the images and the non-image features. On the other hand, adding too many fully connected layers after the concatenation makes it hard for the model to find any relevant relations, while it also drastically increases the amount of model parameters. I therefore choose to add two fully connected layers after concatenation, with 16 and 3 nodes respectively. This leads to a total of 88,315 parameters for Models 2 and 3. Figure 10 displays a schematic illustration of the final part of the CNN architecture of Models 2 and 3. The first four blocks of the architecture are the same as for Model 1.

[3] The tool for generating this figure is obtained from LeNail (2019).

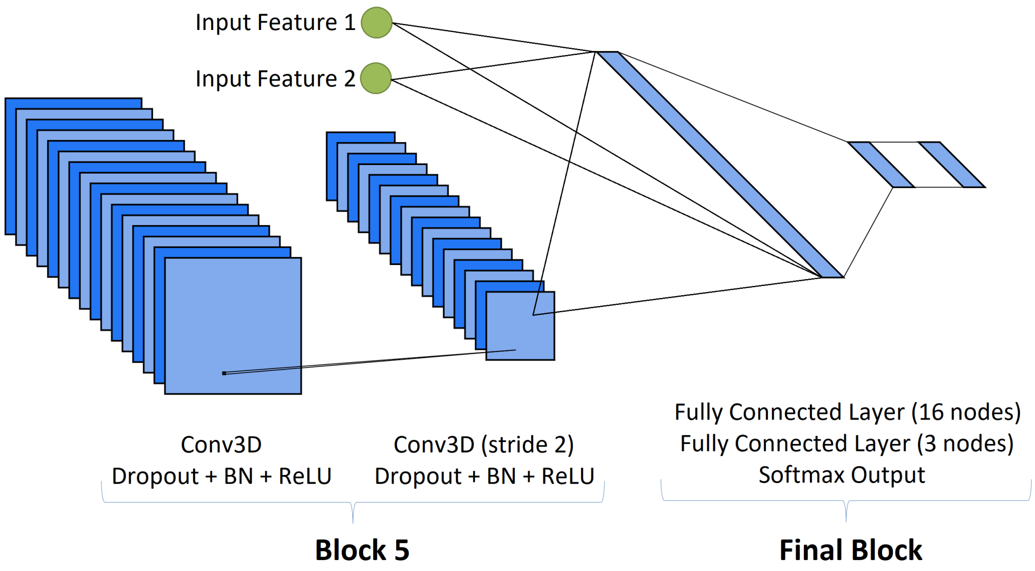


Figure 10: A schematic illustration[4] of the final layers of the CNN architecture of Models 2 and 3.

3.3.3 Training Setup

For each of the three models and each of the three time intervals, the following paradigm is repeated 30 times. These are referred to as *repetitions*. This results in 30 different model configurations for each of the three models and each of the three time intervals.

1. *Shuffle split*: The data is randomly separated into a training set (80%), validation set (10%), and test set (10%) on the subject level. For the validation and test set, one randomly selected observation per subject is included. I use a shuffle split rather than cross-validation, meaning that the split is unrestricted for each of the 30 repetitions. For each time interval, the 30 random splits are the same for the three different models.

2. *Calculation of converter penalty terms*: For Model 3, the penalty terms δ_{AD} and δ_{MCI} are calculated based on the relative frequencies of converters to AD, converters to MCI, and non-converters in the training set of each repetition (Section 3.3.1).

3. *Calculation of batch size*: For all three models, the batch size is calculated based on the relative frequencies of converters and non-converters in the training set of each repetition. I define the batch size as the total amount of training samples divided by the amount of converters, such that on average there is one converter in every batch.

4. *Normalisation*: The voxelwise mean and standard deviation of the images are calculated based on the training set in each repetition. All images in the training, validation, and test set are normalised by subtracting that mean and dividing by that standard deviation. For Models 2 and 3, the same normalisation procedure applies for the two non-image features.

[4] The tool for generating this figure is obtained from LeNail (2019).

5. *Training*: The models are trained on the training set in each repetition. As every individual can have multiple MRI scans available within a time interval, it is possible to train the CNN using all available scans of the individuals in the training set at every epoch. In that case, the model could focus too much on the individuals that have the most MRI scans available. This could lead to a loss of generalisability. I therefore randomly choose one of the available observations per individual at every epoch. The network is compiled with the Adam optimiser (learning rate: 0.001, epsilon: 1e-8, decay: 0.0, β_1: 0.9, β_2: 0.999). Either the cross-entropy loss function (Models 1 and 2) or the custom loss function (Model 3) is applied. A dropout rate of 30% is used for both the convolutional and the convolutional pooling layers.

6. *Model selection*: The loss and the evaluation metrics are calculated after every batch. After every epoch, they are calculated over the entire training and validation set. Training is stopped when the validation loss does not decrease for 20 consecutive epochs. Sometimes, the model can get stuck in a bad local minimum due to the random initialisation of weights. When there is no improvement in the validation loss after the first epoch, or when the final model does not predict all three classes at least once in the training set, the repetition is restarted with a new random weight initialisation, with a maximum of five reinitialisations. The model with the lowest validation loss is selected as the final model of that repetition.

The final model in every repetition is evaluated on the test set of that repetition. The evaluation metrics are calculated for each of these models and averaged over the 30 repetitions.

3.4 Performance Evaluation

As mentioned in Section 2.1, I make use of the evaluation framework of the TADPOLE challenge. This challenge uses two different evaluation metrics for the task of predicting AD progression. Both are explained below.

MAUC

The multiclass area under the receiver operating characteristic (ROC) curve (MAUC) is a generalisation of the area under the ROC curve (AUC) (Hand and Till, 2001). AUC is a measure for binary classification, indicating the ability of a model to separate between two classes. The AUC for classification of class c_i against class c_j is defined as

$$\hat{A}(c_i|c_j) = \frac{S_i - n_i(n_i+1)/2}{n_i n_j}, \tag{12}$$

where S_i is the sum of the ranks of the test samples that belong to class c_i after ranking all samples of c_i and c_j in increasing likelihood of belonging to class c_i, and n_i and n_j are the number of samples belonging to class c_i and c_j respectively. The average AUC of classes c_i and c_j is calculated as

$$\hat{A}(c_i, c_j) = \frac{1}{2}(\hat{A}(c_i|c_j) + \hat{A}(c_j|c_i)). \tag{13}$$

The MAUC is obtained by averaging $\hat{A}(c_i, c_j)$ over all pairs of classes:

$$MAUC = \frac{2}{C(C-1)} \sum_{i=2}^{C} \sum_{j=1}^{i-1} \hat{A}(c_i, c_j), \tag{14}$$

where C is the number of classes. The MAUC is thus a measure that indicates how well a model is able to separate the pairs of classes (CN vs. MCI, CN vs. AD, and MCI vs. AD) in a multiclass classification or prediction framework.

For the subgroup of converters, there are only two classes instead of three. Converters are defined as individuals that progress from their current diagnosis to a more advanced stage. The labels are therefore only MCI and AD, but never CN. It is thus not possible to calculate the MAUC of the converters. Computing AUC for converters is not a good alternative, as the model does calculate a CN probability for the converters which should not be neglected in the evaluation of the prediction accuracy. I therefore include balanced classification accuracy as a second evaluation metric.

BCA

Unlike MAUC, balanced classification accuracy (BCA) (Brodersen et al., 2010) is a non-probabilistic measure. This means that it only looks at the class with the highest likelihood, without taking into account the relative likelihood of this class compared to the other two classes. Uncertainty in the predictions is therefore not reflected in this metric. The BCA of class c is defined as

$$BCA_c = \frac{1}{2}\left[\frac{TP_c}{TP_c + FN_c} + \frac{TN_c}{TN_c + FP_c}\right], \tag{15}$$

such that the BCA of e.g. the AD class depends on the amount of true positives (AD was predicted correctly), the amount of false negatives (CN or MCI was predicted when it should have been AD), the amount of true negatives (CN or MCI was predicted and the true class was also CN or MCI), and the amount of false positives (AD was predicted when it should have been CN or MCI).

Note that for true negatives, BCA does not differentiate between the “negative” classes. An observation that was predicted to be MCI but was in fact CN is considered a true negative, even though the prediction was wrong. The overall BCA is defined as the average BCA over all classes:

$$BCA = \frac{1}{C} \sum_{c=1}^{C} BCA_c, \tag{16}$$

where C is the number of classes. BCA can be calculated over the entire test set, as well as for converters and non-converters separately.

3.5 Visualisation with Grad-CAM

Convolutional neural networks are often seen as a black box, as it is difficult to interpret the consecutive layers of nonlinear operations that construct these types of models. Visualisation techniques aim to overcome this problem. Selvaraju et al. (2017) describes gradient-weighted class activation mapping (Grad-CAM), which can be used to understand the activations of image features. Their algorithm uses gradient information in order to visualise the relative importance of every activation in a CNN. As a result, an attention map can be created for every layer in a CNN.

Grad-CAM is originally intended for visualisation of the final convolutional layer in a network based on 2D images. Yang et al. (2018) describes that for 3D images, the final convolutional layer is often of a much lower resolution. This introduces a trade-off between visualising one of the first layers and one of the final layers of a CNN. Early layers are of high resolution, however, the activation maps might still be changed by succeeding layers. Later layers are more reliable, however the resolution could be too low to be able to distinguish any relevant information.

I use Grad-CAM to visualise network activations of Model 3. The corresponding results are relevant for addressing Question 3 (Section 1.4). For each of the three time intervals, one of the 30 repetitions is used and the activations in the third convolutional layer are visualised. The code for this application is adapted from `https://github.com/eclique/keras-gradcam` in order to be compatible with 3D MRI scans. Separate activation maps are constructed for the three different time intervals and the three different classes (CN, MCI, AD). For each class, only the correctly predicted observations in the test set are taken into account for the visualisation of network activations.

3.6 Application to External Data

Next to training, validating, and testing the performance of the three models on the ADNI data, the generalisability of the models is tested by evaluating them on the external Parelsnoer dataset. The corresponding results are relevant for addressing Question 4 (Section 1.4). The preprocessing of the Parelsnoer data and the evaluation framework are very similar to that of the ADNI data. The Parelsnoer data consists of only baseline MRI scans, which can be matched to every available future diagnosis of the same individual. These matches are then divided into the same three time intervals as the ADNI data: 0-1 year ahead, 1-2 years ahead, and 2-3 years ahead. When a unique MRI scan can be matched to multiple future diagnoses within the same time interval, one of these is randomly selected to be in the external test set. This leads to the summary statistics displayed in Table 2 in Section 2.2.

Models 1, 2, and 3 are repeated 30 times for each of the three different time intervals. For each of the three models and each of the three time intervals, the 30 final model configurations are used to produce forecasts of the Parelsnoer test sets. The evaluation metrics from Section 3.4 are computed and again averaged over the 30 repetitions.

3.7 Implementation

The code for this project is written in Python 3.6.9 and the software is implemented with the open source neural network library Keras, using a Tensorflow 1.12.0 backend. The models are trained and evaluated on a GPU cluster to which three NVIDIA GPU cards are connected: a GeForce GTX 1080, a TITAN V and a Quadro P6000. Finally, all of the code created for this project can be found at https://github.com/thomasmichelotti/CNN-for-predicting-AD-progression, including an explanation on how to replicate the results in this study.

4 Results

I first present the results of the three different methods after training, validating, and testing on the ADNI data. Next, the activation maps obtained with the Grad-CAM tool are displayed. Finally, I show the performance of the models on the external Parelsnoer data.

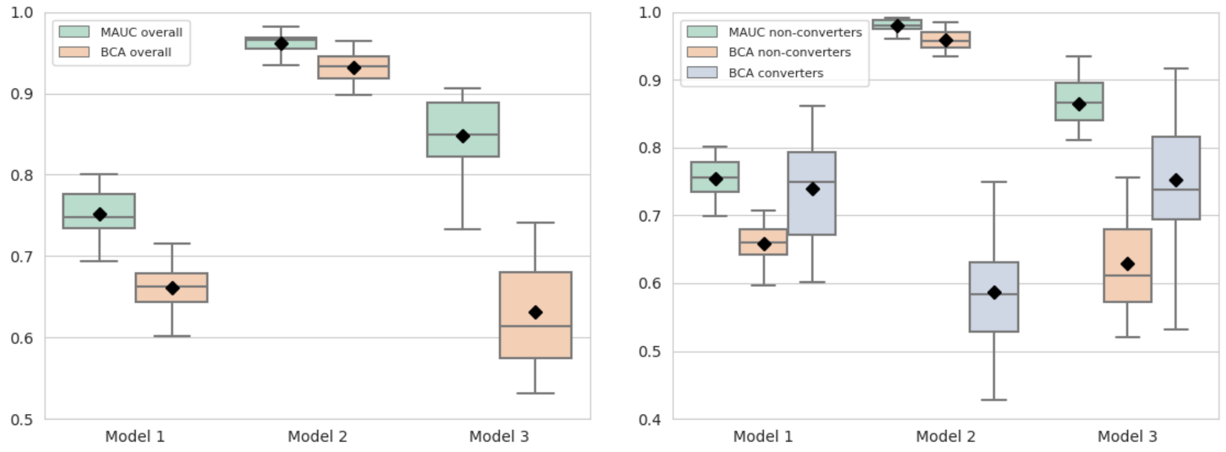
4.1 Prediction Performance

Table 3 displays the average MAUC and BCA scores of the three different models and for the three different time intervals over 30 repetitions. Next to the overall results, the separate performance for non-converters and converters is shown. As discussed in Section 3.4, MAUC cannot be calculated for the subgroup of converters.

	0-1 year		1-2 years		2-3 years	
	MAUC	BCA	MAUC	BCA	MAUC	BCA
Model 1						
Overall	0.752	0.661	0.743	0.661	0.700	0.614
Non-converters	0.754	0.659	0.762	0.662	0.725	0.618
Converters	-	0.740	-	0.677	-	0.593
Model 2						
Overall	0.962	0.932	0.932	0.864	0.862	0.787
Non-converters	0.979	0.958	0.972	0.921	0.936	0.860
Converters	-	0.587	-	0.562	-	0.561
Model 3						
Overall	0.848	0.631	0.815	0.629	0.798	0.673
Non-converters	0.864	0.629	0.859	0.630	0.865	0.704
Converters	-	0.752	-	0.767	-	0.678

Table 3: Average prediction performance of the three different models for the three different time intervals over 30 repetitions.

Before discussing these results in more detail, it should be mentioned that the average scores in Table 3 cannot be meaningfully compared without the distribution over the 30 repetitions. Figure 11a therefore displays boxplots of the overall results for predictions of AD progression up to one year in the future. Figure 11b displays the same results, separated for non-converters and converters. Boxplots indicate the median (middle line in each coloured box) and the interquantile range (IQR; the coloured boxes that span the 25th to the 75th percentile) of the data. The whiskers that extend from above and underneath the box indicate the “maximum” (75th percentile + 1.5 × IQR) and “minimum” (25th percentile - 1.5 × IQR) of the data. The black diamonds in the boxplots indicate the mean of the 30 repetitions.



(a) Overall

(b) Separated

Figure 11: Boxplots of the MAUC and BCA scores of the three different models for predictions up to one year in the future, both (a) overall and (b) separated for non-converters and converters.

For predictions 0-1 year ahead, the results in Table 3 and Figure 11a indicate that Model 2 yields a substantially higher overall performance compared to Models 1 and 3. Model 3 obtains a better overall performance compared to Model 1 based on MAUC, however the overall BCA does not differ much between these two models. The boxplots also show more variability in the results of Model 3 compared to the other two models. This is likely a result of the converter penalty in the custom loss function. The reasons for that are discussed in detail in Section 5.1.

As discussed previously, only 8% of the data in the 0-1 year interval represents a conversion. The high performance and low variability in the results of Model 2 could be caused by the inclusion of current diagnosis as input variable. It is therefore relevant to examine prediction performance for non-converters and converters separately. The results in Table 3 and Figure 11b show that for the subgroup of non-converters, the strong performance of Model 2 is even more apparent than for the overall group. Model 3 again outperforms Model 1 based on MAUC, with similar BCA scores. For the subgroup of converters, Model 2 obtains an average BCA that is substantially lower than its BCA for non-converters. This indicates that a model with a regular loss function indeed focuses too much on the current diagnosis. Models 1 and 3 obtain a similar BCA for the subgroup of converters, which is substantially higher compared to Model 2. The boxplots indicate a relatively large amount of variability in the BCA scores for converters, likely caused by the low number of observations.

For predictions of AD progression 1-2 years and 2-3 years in the future, Table 3 shows that the performance is generally lower for predictions further in the future. Appendix A contains the boxplots of the results over 30 repetitions for predictions 1-2 years and 2-3 years ahead. The relative performance between the three models is similar for all time intervals: Model 2 outperforms the other two models based on the subgroup of non-converters, however, it obtains the lowest scores for the subgroup of converters. Model 3 obtains the highest average scores for the subgroup of converters, although this score is very similar to that of Model 1 for the first time interval. It is interesting to see that Model 3 seems to outperform Model 1 more clearly for predictions further in the future, both for the subgroup of converters and the subgroup of non-converters.

4.2 Visualisation with Grad-CAM

Before discussing the activations in the brain that are obtained with the Grad-CAM tool, it is useful to shortly discuss the regions in the brain that are known to be related to memory. Figure 12 shows the four main regions: amygdala, hippocampus, cerebellum, and prefrontal cortex.

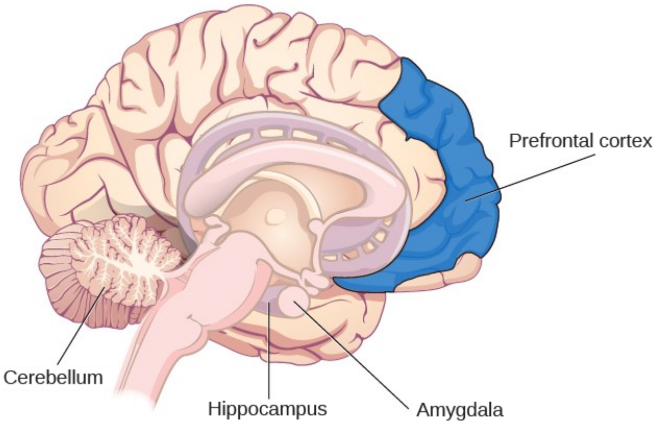


Figure 12: Regions in the brain that are related to memory.

The amygdala is related to the processing of emotional information and is mainly associated with fear and the memory of fear. The hippocampus is specifically involved with declarative (explicit), episodic, and recognition memory. The cerebellum is associated to the processing of procedural (implicit) memory, such as remembering how to play the piano. Finally, the prefrontal cortex is involved with memory related to semantic tasks. Figure 13 displays the Grad-CAM results for Model 3, for the three different classes (CN, MCI, AD) and the three time intervals.

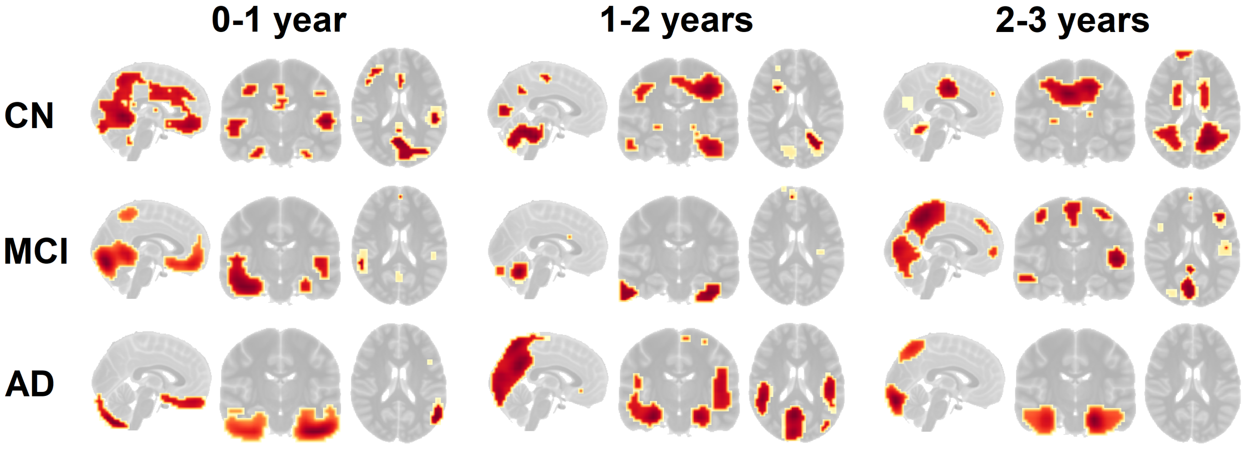


Figure 13: Grad-CAM results of the three different classes for the three time intervals. A threshold is used such that only the strongest activations are shown.

For predictions up to one year in the future, Figure 13 displays model activations in the cerebellum and prefrontal cortex for MCI and AD. For CN, the activations are slightly more scattered over different regions. In the activation maps for predictions 1-2 years ahead, activations are visible in the cerebellum for all three classes. The prefrontal cortex shows only weak activations in this interval. The CN activations are again the most scattered over different regions. For AD, the parietal and occipital lobes show strong activations, which is not the case for the first time interval. For predictions 2-3 years in the future, the activations are quite scattered over different regions for both CN and MCI, although the parietal lobe shows the strongest activations for MCI. The cerebellum and parietal lobe again appear to be highly relevant for the AD class.

An important note must be made about the activation maps in Figure 13. For each of the three time intervals, one of the 30 model repetitions is chosen and the activation maps are averaged over the test subjects in that repetition. The output of the Grad-CAM tool, however, shows slightly different activation maps for some of the model repetitions, similar to the variability that is present in the prediction performance (Figure 11). It is therefore hard to make firm conclusions regarding the exact features that the CNN mostly focuses on when making predictions of AD progression. Section 5.1 discusses this variability in the results in more detail.

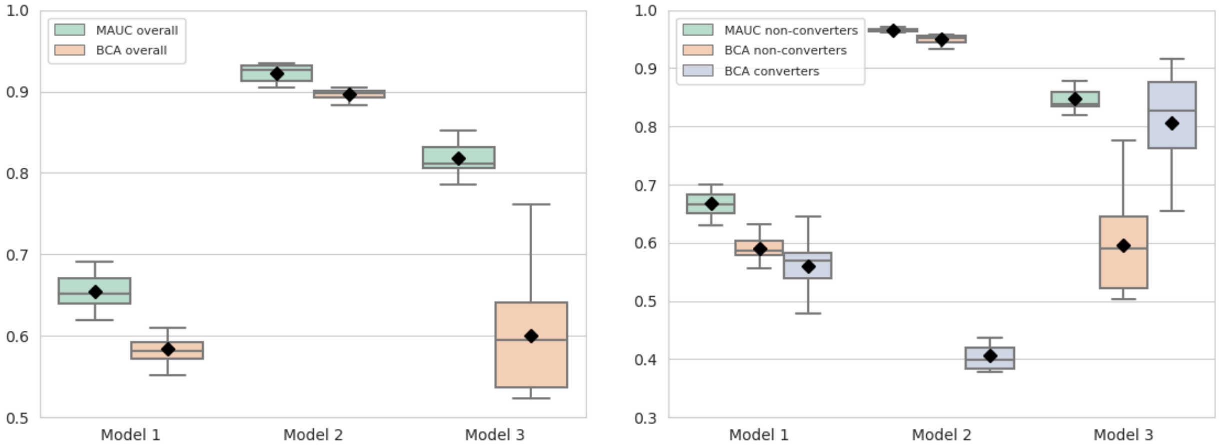
4.3 Application to External Data

Table 4 displays the average MAUC and BCA scores of the three different models and for the three different time intervals on the external Parelsnoer dataset. Next to the overall results, the separate performance for non-converters and converters is shown.

	0-1 year		1-2 years		2-3 years	
	MAUC	BCA	MAUC	BCA	MAUC	BCA
Model 1						
Overall	0.654	0.584	0.653	0.591	0.650	0.588
Non-converters	0.667	0.590	0.658	0.588	0.667	0.600
Converters	-	0.561	-	0.627	-	0.569
Model 2						
Overall	0.923	0.896	0.892	0.839	0.858	0.792
Non-converters	0.965	0.949	0.962	0.924	0.940	0.874
Converters	-	0.407	-	0.437	-	0.488
Model 3						
Overall	0.818	0.601	0.796	0.621	0.822	0.717
Non-converters	0.847	0.596	0.861	0.633	0.899	0.769
Converters	-	0.806	-	0.748	-	0.597

Table 4: Average prediction performance of the three different models for the three different time intervals on the Parelsnoer data over 30 repetitions.

Again, it is relevant to show the distribution of the results over the 30 repetitions. Figure 14a displays boxplots of the overall results of the three models for predictions up to one year in the future when applied on the external Parelsnoer data. Figure 14b displays the same results, separated for non-converters and converters.



(a) Overall

(b) Separated

Figure 14: Boxplots of the overall MAUC and BCA scores of the three different models for predictions up to one year in the future for the Parelsnoer data, both (a) overall and (b) separated for non-converters and converters.

Table 4 and Figure 14a show that the overall performance of the models on the Parelsnoer data is generally lower than on the ADNI data for predictions up to one year in the future. The difference in performance is larger for Model 1 compared to the other two models. This indicates that a model based on only images generalises less well to external data compared to models that make use of additional input features. The overall MAUC and BCA scores of Model 2 are higher than those of the other two models. Model 3 appears to more clearly outperform Model 1 compared to the performance on the ADNI data based on MAUC, while the BCA is again similar.

The results can again be separated for converters and non-converters. Table 4 and Figure 14b show that for non-converters, Model 2 strongly outperforms the other two models. Model 3 performs much better than Model 1 based on MAUC, with similar BCA scores. For Model 2, the difference in performance between converters and non-converters is even larger than it was for the ADNI data. Both Models 1 and 3 outperform Model 2 for the subgroup of converters, and Model 3 also seems to perform substantially better than Model 1. This indicates that Model 3 with a custom loss function generalises quite well to external data. Similar to the results of the ADNI data and the Grad-CAM tool, the boxplots in Figure 14 indicate a relatively large amount of variability in the results of Model 3. The reasons for that are discussed in more detail in Section 5.1.

For predictions 1-2 years and 2-3 years in the future, Table 4 shows that the performance is generally lower for predictions further in the future. Appendix B contains boxplots of the Parelsnoer results for predictions 1-2 years and 2-3 years ahead. It is interesting to note that the performance of Model 1 seems quite stable for the different time intervals, both for converters and non-converters. For Model 2, the overall performance drops more substantially for predictions further in the future, mostly due to the subgroup of non-converters. For the subgroup of converters, the performance of Model 3 drops substantially for predictions further in the future. On average, however, Model 3 outperforms the other two models for the subgroup of converters for all time intervals.

5 Discussion

In this section, I first review the main results that are presented in the previous section in more detail, mostly regarding the variability in the results of Model 3. Next, I discuss limitations of this research as well as suggestions for future work. Finally, I present the main conclusions that can be drawn from the results in this study, referring back to the research questions in Section 1.4.

5.1 Main Results

In this study, I compare an image-based CNN with a regular loss function (Model 1), a multi-input CNN based on images and two non-image features with a regular loss function (Model 2), and a multi-input CNN based on images and two non-image features with a custom loss function (Model 3). Model 2 outperforms the other two models based on the subgroup of non-converters, which is likely caused by the model focusing almost exclusively on the current diagnosis. As a result, Model 2 performs quite poorly on the subgroup of converters. Model 3 yields the highest results for the subgroup of converters, while also performing well for the subgroup of non-converters. Furthermore, the Grad-CAM tool shows several brain regions with substantial model activations. Specifically, the cerebellum appears to be an important brain region for predictions of AD progression. Finally, the Parelsnoer results show that Model 3 generalises reasonably well to external data. The goal of this subsection is to discuss the main results in more detail and to provide some insights regarding the variability in the results that was discussed in the previous section.

Although the average MAUC and BCA scores indicate that Model 3 achieves the goal of tackling the imbalance between non-converters and converters in the data, the boxplots in Figure 11 show more variability in the results over the 30 repetitions for Model 3 than for Models 1 and 2. This variability is also reflected in the Grad-CAM results presented in Section 4.2. Furthermore, the variability is visible in the results of Model 3 on the external Parelsnoer dataset (Figure 14).

After examining the 30 repetitions of Model 3 individually, it seems that there are some "outlier repetitions" that can roughly be divided into two groups. Some of the repetitions seem to perform very well on the subgroup of converters, but less well on the subgroup of non-converters. Some other repetitions yield relatively high scores for the subgroup of non-converters, but lower scores for the subgroup of converters. This is probably a result of the discrete converter penalty in the custom loss function (Equation 8). The loss is computed after every batch that is passed through the network. I defined the batch size as the total amount of training samples divided by the amount of converters, such that on average there is one converter in every batch (Section 3.3.3). There is, however, no restriction that there should be exactly one converter in every batch. When all batches in the training set only consist of non-converters, the average converter penalty for every batch is exactly one. In that case, the loss function is continuous, which makes it easier for the backpropagation algorithm to converge to the minimum of the loss function. Some of the batches, however, contain one or multiple converters. The average loss over these batches is therefore substantially higher, which causes sudden jumps in the loss function. This hinders the backpropagation algorithm and makes the optimisation of the loss function more unstable. As the model stops training after a certain epoch (when the validation loss has not increased for 20 consecutive epochs), the final batches of the final epoch are very important. When the final batches do not contain any converters, the final gradient steps in the loss function are taken into the direction of a minimum that is optimal for non-converters. This means that in the final optimisation steps of the training process, more importance is given to features that are relevant for predicting non-converters, such as the current diagnosis. If this is the case for a certain model repetition, this repetition will likely predict non-converters more accurately than converters.

On the other hand, if the final batches of the final epoch contain more than one converter per batch, the final gradient steps in the loss function are taken into the direction of a minimum that is more optimal for converters. The model repetitions for which this is the case will likely predict converters more accurately than non-converters. Even when there is exactly one converter present in the final batches of the final epoch of a model repetition, there is still the problem that the converter penalty differentiates between converters to MCI and converters to AD. This means that, even when you restrict the batches in the CNN to always contain exactly one converter, the final performance of the model will likely be better for either converters to MCI or converters to AD.

An important sidenote about the results of Model 3 should thus be made. Even though the converter penalty in the custom loss function enables the CNN to learn from the current diagnosis without focusing almost exclusively on that input feature (which is the case for Model 2), it does make the loss function discrete. This creates jumps in the gradient descent algorithm, which makes the optimisation of the CNN more unstable. Some of the 30 repetitions therefore converge to a minimum that is not optimal for both non-converters and converters. This explains the variability in the ADNI results (Section 4.1), the visualised activation maps for different model repetitions (Section 4.2), and the Parelsnoer results (Section 4.3).

5.2 Limitations and Future Work

Several limitations to this study should be addressed. First, I mention two important limitations that hold for research into AD progression forecasts in general. Second, I address some limitations that are specific to this study. Furthermore, I discuss some ideas for directions of future research in the domain of multi-input models for predicting AD progression.

There are two common limitations that hold for most studies that focus on the task of predicting future diagnosis in AD progression. The first concerns the availability of data. Although the ADNI database contains a substantial amount of MRI scans and other measurements, the amount of training samples is quite low compared to the amount of parameters to be optimised in deep learning models. As discussed previously, only a small percentage of the data represents a conversion to a more advanced stage. The subgroup of converters is therefore particularly small, especially for short-term predictions. In this study, I used a custom loss function to overcome the imbalance between converters and non-converters. Alternatively, it is possible to undersample the subgroup of non-converters or use data augmentation to upsample the subgroup of converters in the training data. It would be interesting to see if these techniques, possibly combined with a custom loss function, can improve prediction performance for AD progression.

Second, the ground truth diagnosis for dementia is the postmortem diagnosis based on pathology. Data with ground truth diagnosis is only rarely available. In the ADNI and Parelsnoer data, diagnosis is based on clinical criteria without autopsy confirmation. Clinical diagnosis is typically confirmed by follow-up observations, however it is possible that some of the patients are included in the wrong class. Due to this noise in registration of diagnoses, there are a few observations in the data for which the current diagnosis is a more advanced stage than the future diagnosis. These observations make it harder for a CNN to learn the correct relations between input and output. However, due to limited availability of data with ground truth diagnosis, clinical diagnosis is currently the best reference standard for AD research. It could be interesting to investigate how the performance changes when observations for which the current diagnosis is a more advanced stage than the future diagnosis are removed from the training data.

The following limitations are specific to this study. First, the CNN architectures presented in Section 3.3.2 have not been optimised. Examples of parameters that could be tuned in order to improve prediction performance are the number of network layers, the learning rate for the Adam algorithm, and the dropout rate in the convolutional layers. As this is an exploratory study focused on comparing the performance of several models, an optimised network achitecture is not of primary interest. In the future, when a deep learning model performs well enough to be used in clinical practice, it is important to tune all relevant hyperparameters in the network. Another consequence of the fact that this is an exploratory study, is that all images are downsampled by a factor of four in order to decrease computation time (Section 3.2). A study that extensively compares model performance with different factors for image downsampling could provide useful insights regarding the necessity of using full-size MRI scans for predictions of AD progression.

Furthermore, this study focuses on prediction performance based on a single MRI scan and corresponding diagnosis. The reason for this is that in clinical practice, most patients do not have an extensive history of hospital visits with multiple MRI scans over time. When a patient does have more than one previous MRI scan available, the method in this study cannot take that into account as additional information. A very helpful extension to this study would be to let the model use multiple previous data points as input, whenever that data is available. Within the area of deep learning, this could be achieved with a Recurrent Neural Network (RNN). RNN's are becoming increasingly popular for predictions based on longitudinal (image) data.

Another limitation is that this study does not investigate the necessity of using different models for different time intervals. In this study, the three models are constructed separately for predictions 0-1 year ahead, 1-2 years ahead, and 2-3 years ahead. There are two reasons for this design choice.

First, a CNN based only on MRI scans that does not include follow-up time as input feature is used as a benchmark (Model 1). It would be much harder for this model to accurately predict AD progression when the range of follow-up times is too large, which would make for an unfair comparison. Second, the focus of this study is on generalisability to new patients, meaning that only one scan per individual is used in every training epoch. By splitting the data into three time intervals, more data can be used in the training process. In future research, however, it would be interesting to compare the performance of separate models for different time intervals to a single model that predicts AD progression for e.g. 0-3 years ahead.

As explained in Section 1.1, the diagnoses CN, MCI, and AD are successive stages during the progression of Alzheimer's disease. This means that there is a natural ordering among the classes. A drawback of using a softmax ouput function (Equation 4) in the models is that it does not take into account the ordinal nature of the data. One way to tackle this is through the loss function. In my research, I tried adding an *ordinal penalty* γ_j to the custom loss function, either separately or combined with the converter penalty. I defined the ordinal penalty as

$$\gamma_j = 1 + \frac{|\operatorname{argmax}(\boldsymbol{y}_{j,\text{true}}) - \operatorname{argmax}(\boldsymbol{y}_{j,\text{pred}})|}{C-1}, \tag{17}$$

where $\boldsymbol{y}_{j,\text{true}}$ is the one-hot encoded vector of length three that indicates the true diagnosis of training sample j (e.g. $[0,\ 1,\ 0]'$ for MCI), $\boldsymbol{y}_{j,\text{pred}}$ is the vector of estimated probabilities for training sample j (e.g. $[0.19,\ 0.72,\ 0.09]'$), and C is the number of classes, equal to three. In order to illustrate how this penalty term works in practice, suppose a training sample has a true diagnosis of AD. The model outputs a probability for each of the three classes. If AD gets the highest probability, the ordinal penalty γ_j will be 1. When MCI gets the highest probability, the ordinal penalty γ_j will be 1.5. Finally, when CN gets the highest probability, the ordinal penalty γ_j will be 2. The ordinal penalty thus forces the model to take the ordinal nature of the data into account. The corresponding results show that a custom loss function with only an ordinal penalty and without converter penalty yields similar results to Model 2 (without converter and without ordinal penalty). A custom loss function with both a converter and an ordinal penalty yields similar results to Model 3 (with only a converter penalty). The ordinal penalty therefore did not seem to improve prediction performance, which is why I did not include it in the main results of this study.

There could be two reasons why the ordinal penalty of Equation 17 does not achieve the expected results. First, similar to the converter penalty, the ordinal penalty is discrete. As a result, the loss function becomes discontinuous, which makes optimisation more unstable. This could be solved by making the ordinal penalty continuous, for instance by taking the index of the true class and subtracting the weighted sum of the predicted classes (e.g. $\gamma_j = |(0*1+1*2+0*3)-(0.19*1+0.72*2+0.09*3)|$). Second, it could be hard in general to take the ordinal nature of the output classes into account through the loss function. Another possibility is to change the encoding of the output classes to CN = [0,0], MCI = [0,1], and AD = [1,1], and to change the softmax output function to a sigmoid output function. It would be interesting to further study the necessity of taking the ordinal nature of the output classes into account, and comparing different methods of doing so.

A limitation that is related to the interpretation of the model output concerns the fact that only activations in the brain are visualised in this study. With the Grad-CAM tool, it is only possible to visualise activations of convolutional layers. After the concatenation of images and non-image features in the network architecture of this study, the final block consists only of fully connected layers (Figure 10). As a result, the Grad-CAM tool cannot be used to compare the relative importance of the MRI scans and the two non-image features for the prediction performance. In future research in the area of multi-input deep learning models, a useful extension would consist of a method that can accurately compare the relative importance of the different model inputs.

A final suggestion for future research also concerns the usage of multi-input models. In the ADNI dataset, a lot more information is available that can be used to improve prediction performance for AD progression. A multi-input model that - next to MRI scans, current diagnosis and follow-up time - takes into account e.g. cognitive test scores, information about the size of the brain, and other AD biomarkers, has the potential to predict AD progression even more accurately.

5.3 Conclusion

Before formulating an answer to the main research question, I shortly address the conclusions regarding the four subquestions that were introduced in Section 1.4.

Question 1 *How should the model address the imbalance between converters and non-converters?*

To answer Question 1, I constructed three different models. This study concludes that, if the goal is to obtain the highest overall prediction performance, it is not necessary to address the imbalance between converters and non-converters. If current diagnosis is included as input feature, a model with a regular loss function obtains a very high overall prediction performance. This is a result of the fact that the majority of the subjects are non-converters, which can easily be predicted based on current diagnosis. When converters are of interest, however, it is very important to address the imbalance in the data. Even though the converter penalty that is used for the custom loss function in this study helps tackling the imbalance in the data, an important remark is that this penalty is discrete. As a result, the optimisation process is slightly unstable which sometimes causes a model repetition to focus too much on either converters or non-converters. Nevertheless, the usage of a custom loss function that differentiates between non-converters, converters to MCI, and converters to AD, appears to be beneficial for predicting AD progression.

Question 2 *How does the performance of the model differ for short-term and long-term predictions?*

The three main models in this study were constructed for three different time intervals. In general, the performance of the models drops slightly for predictions further in the future. This study indicates that Model 3 with custom loss function experiences a relatively low drop in overall performance compared to the other two models. Especially for the subgroup of converters, Model 3 outperforms the other two models more clearly for predictions 1-2 years ahead and 2-3 years ahead, compared to the 0-1 year interval.

Question 3 *Which features are most important for predicting Alzheimer's disease progression?*

The Grad-CAM tool was used to visualise activations of Model 3 with custom loss function. The corresponding results indicate that it is hard to make firm conclusions about the most important features. The cerebellum, however, seems to be an important brain region for predictions of AD progression, as it appears in all three time intervals for the AD class. The MCI class also shows activations in the cerebellum, but not as strongly as the AD class. For CN, the activations are more scattered over different regions compared to the other two classes.

Question 4 *How well does the performance of the model generalise to external data?*

The external performance of the three main methods is validated on the Parelsnoer dataset. The corresponding results indicate that there is a drop in performance on the Parelsnoer data compared to the ADNI data, for all three models. For Model 3 with custom loss function, however, the drop is not very large, which shows that the performance generalises quite well to external data.

Finally, I formulate an answer to the main research question: *how accurately can the progression of Alzheimer's disease be predicted based on a single MRI scan and corresponding diagnosis?* This study shows that a CNN is able to predict AD progression quite accurately based on a single MRI scan, current diagnosis, and follow-up time, when a custom loss function is used to overcome the imbalance between non-converters, converters to MCI and converters to AD. The main drawback of the converter penalty is that the loss function is no longer continuous, which makes the optimisation of the model less stable. When this limitation is addressed, and more (non-image) features are added to the multi-input CNN, my expectation is that prediction performance will improve even further. Hopefully, this will bring deep learning models for predictions of Alzheimer's disease progression one step closer to being implemented in clinical practice.

References

Allen, G. I., Amoroso, N., Anghel, C., Balagurusamy, V., Bare, C. J., Beaton, D., Bellotti, R., Bennett, D. A., Boehme, K. L., Boutros, P. C., et al. (2016). Crowdsourced estimation of cognitive decline and resilience in Alzheimer's disease. *Alzheimer's & Dementia*, 12(6):645–653.

Alzheimer, A. (1907). Uber eine eigenartige Erkrankung der Hirnrinde. *Zentralbl. Nervenh. Psych.*, 18:177–179.

Alzheimer's Disease International (2019). World Alzheimer Report 2019 - attitudes to dementia. *London: Alzheimer's Disease International. Retrieved from https://www.alz.co.uk/research/WorldAlzheimerReport2019.pdf.*

Ansart, M., Epelbaum, S., Bassignana, G., Bône, A., Bottani, S., Cattai, T., Couronne, R., Faouzi, J., Koval, I., Louis, M., et al. (2019). Predicting the progression of mild cognitive impairment using machine learning: A systematic and quantitative review.

Basaia, S., Agosta, F., Wagner, L., Canu, E., Magnani, G., Santangelo, R., Filippi, M., Alzheimer's Disease Neuroimaging Initiative, et al. (2019). Automated classification of Alzheimer's disease and mild cognitive impairment using a single MRI and deep neural networks. *NeuroImage: Clinical*, 21:101645.

Brodersen, K. H., Ong, C. S., Stephan, K. E., and Buhmann, J. M. (2010). The balanced accuracy and its posterior distribution. In *2010 20th International Conference on Pattern Recognition*, pages 3121–3124. IEEE.

Bron, E. E., Smits, M., Van Der Flier, W. M., Vrenken, H., Barkhof, F., Scheltens, P., Papma, J. M., Steketee, R. M., Orellana, C. M., Meijboom, R., et al. (2015). Standardized evaluation of algorithms for computer-aided diagnosis of dementia based on structural MRI: the CADDementia challenge. *NeuroImage*, 111:562–579.

Bron, E. E., Steketee, R. M., Houston, G. C., Oliver, R. A., Achterberg, H. C., Loog, M., van Swieten, J. C., Hammers, A., Niessen, W. J., Smits, M., et al. (2014). Diagnostic classification of arterial spin labeling and structural MRI in presenile early stage dementia. *Human brain mapping*, 35(9):4916–4931.

Cui, R., Liu, M., Alzheimer's Disease Neuroimaging Initiative, et al. (2019). RNN-based longitudinal analysis for diagnosis of Alzheimer's disease. *Computerized Medical Imaging and Graphics*, 73:1–10.

Cuingnet, R., Gerardin, E., Tessieras, J., Auzias, G., Lehéricy, S., Habert, M.-O., Chupin, M., Benali, H., Colliot, O., Initiative, A. D. N., et al. (2011). Automatic classification of patients with Alzheimer's disease from structural MRI: a comparison of ten methods using the ADNI database. *neuroimage*, 56(2):766–781.

Douaud, G., Smith, S., Jenkinson, M., Behrens, T., Johansen-Berg, H., Vickers, J., James, S., Voets, N., Watkins, K., Matthews, P. M., et al. (2007). Anatomically related grey and white matter abnormalities in adolescent-onset schizophrenia. *Brain*, 130(9):2375–2386.

Gaugler, J., James, B., Johnson, T., Marin, A., and Weuve, J. (2019). 2019 Alzheimer's Disease Facts and Figures. *Alzheimers & Dementia*, 15(3):321–387.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323.

Hand, D. J. and Till, R. J. (2001). A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine learning*, 45(2):171–186.

Hane, F. T., Lee, B. Y., and Leonenko, Z. (2017a). Recent progress in Alzheimer's disease research, part 1: Pathology. *Journal of Alzheimer's Disease*, 57(1):1–28.

Hane, F. T., Robinson, M., Lee, B. Y., Bai, O., Leonenko, Z., and Albert, M. S. (2017b). Recent progress in Alzheimer's disease research, part 3: diagnosis and treatment. *Journal of Alzheimer's Disease*, 57(3):645–665.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Jack, C. R., Vemuri, P., Wiste, H. J., Weigand, S. D., Lesnick, T. G., Lowe, V., Kantarci, K., Bernstein, M. A., Senjem, M. L., Gunter, J. L., et al. (2012). Shapes of the trajectories of 5 major biomarkers of Alzheimer disease. *Archives of neurology*, 69(7):856–867.

Jain, R., Jain, N., Aggarwal, A., and Hemanth, D. J. (2019). Convolutional neural network based Alzheimer's disease classification from magnetic resonance brain images. *Cognitive Systems Research*, 57:147–159.

Jung, W. B., Lee, Y. M., Kim, Y. H., and Mun, C.-W. (2015). Automated classification to predict the progression of Alzheimer's disease using whole-brain volumetry and dti. *Psychiatry investigation*, 12(1):92.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

LeNail, A. (2019). NN-SVG: Publication-ready neural network architecture schematics. *Journal of Open Source Software*, 4(33):747.

Lu, P. (2019). *Statistical Learning from Multimodal Genetic and Neuroimaging data for prediction of Alzheimer's Disease*. PhD thesis, Sorbonne Université.

Marinescu, R. V., Oxtoby, N. P., Young, A. L., Bron, E. E., Toga, A. W., Weiner, M. W., Barkhof, F., Fox, N. C., Eshaghi, A., Toni, T., et al. (2020). The Alzheimer's disease prediction of longitudinal evolution (TADPOLE) challenge: Results after 1 year follow-up. *arXiv preprint arXiv:2002.03419*.

Marinescu, R. V., Oxtoby, N. P., Young, A. L., Bron, E. E., Toga, A. W., Weiner, M. W., Barkhof, F., Fox, N. C., Klein, S., Alexander, D. C., et al. (2018). TADPOLE challenge: Prediction of longitudinal evolution in Alzheimer's disease. *arXiv preprint arXiv:1805.03909*.

Patterson, C. (2018). World Alzheimer Report 2018 - the state of the art of dementia research: new frontiers. *London: Alzheimer's Disease International. Retrieved from https://www.alz.co.uk/research/WorldAlzheimerReport2018.pdf*.

Petersen, R. C., Aisen, P., Beckett, L. A., Donohue, M., Gamst, A., Harvey, D. J., Jack, C., Jagust, W., Shaw, L., Toga, A., et al. (2010). Alzheimer's disease neuroimaging initiative (ADNI): clinical characterization. *Neurology*, 74(3):201–209.

Prince, M., Albanese, E., Guerchet, M., and Prina, M. (2014). World Alzheimer Report 2014 - dementia and risk reduction: an analysis of protective and modifiable factors. *London: Alzheimer's Disease International. Retrieved from https://www.alz.co.uk/research/WorldAlzheimerReport2014.pdf.*

Prince, M., Bryce, R., and Ferri, C. (2011). World Alzheimer Report 2011 - the benefits of early diagnosis and intervention. *London: Alzheimer's Disease International. Retrieved from https://www.alz.co.uk/research/WorldAlzheimerReport2011.pdf.*

Prince, M., Wimo, A., Guerchet, M., Ali, G.-C., Wu, Y.-T., and Prina, M. (2015). World Alzheimer Report 2015 - the global impact of dementia: an analysis of prevalence, incidence, cost and trends. *London: Alzheimer's Disease International. Retrieved from https://www.alz.co.uk/research/WorldAlzheimerReport2015.pdf.*

Robinson, M., Lee, B. Y., and Hane, F. T. (2017). Recent progress in Alzheimer's disease research, part 2: genetics and epidemiology. *Journal of Alzheimer's Disease*, 57(2):317–330.

Sarica, A., Cerasa, A., Quattrone, A., and Calhoun, V. (2016). A machine learning neuroimaging challenge for automated diagnosis of mild cognitive impairment. *J. Neurosci. Methods. Available online at: https://www.kaggle.com/c/mci-prediction.*

Schneider, L. (2019). A resurrection of aducanumab for Alzheimer's disease. *The Lancet Neurology.*

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626.

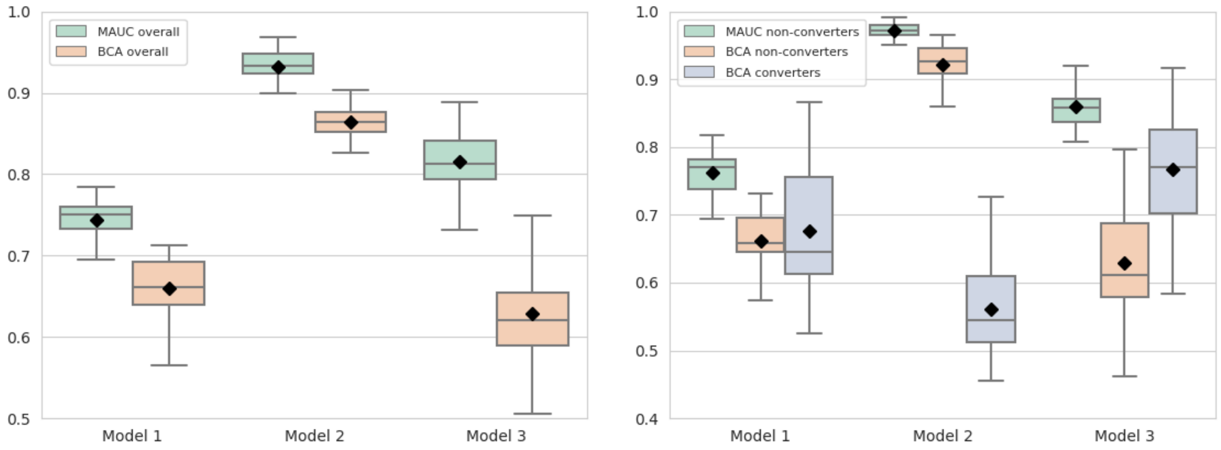
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Sukkar, R., Katz, E., Zhang, Y., Raunig, D., and Wyman, B. T. (2012). Disease progression modeling using hidden markov models. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2845–2848. IEEE.

Yang, C., Rangarajan, A., and Ranka, S. (2018). Visual explanations from deep 3D convolutional neural networks for Alzheimer's disease classification. In *AMIA Annual Symposium Proceedings*, volume 2018, page 1571. American Medical Informatics Association.

Yang, D. W., Hong, Y. J., Cho, A.-H., Yoon, B., Shim, Y. S., Cho, J. H., Kim, J.-Y., and Kim, B. S. (2010). Differences of gray matter and white matter volume in Alzheimer's disease and subcortical ischemic vascular dementia. *Alzheimer's & Dementia: The Journal of the Alzheimer's Association*, 6(4):S10–S11.

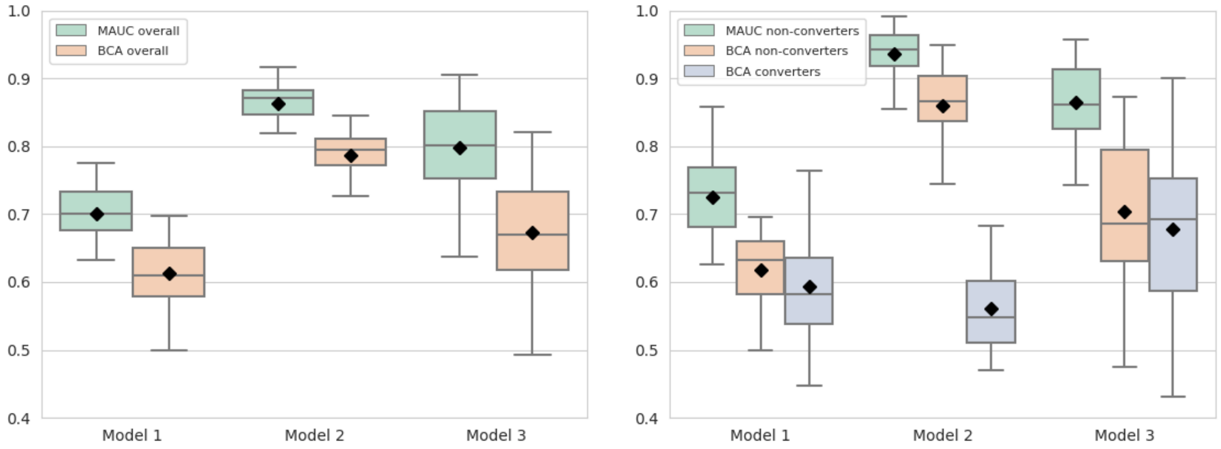
A Supplementary ADNI Results



(a) Overall

(b) Separated

Figure A1: Boxplots of the MAUC and BCA scores of the three different models for predictions between one and two years in the future, both (a) overall and (b) separated for non-converters and converters.

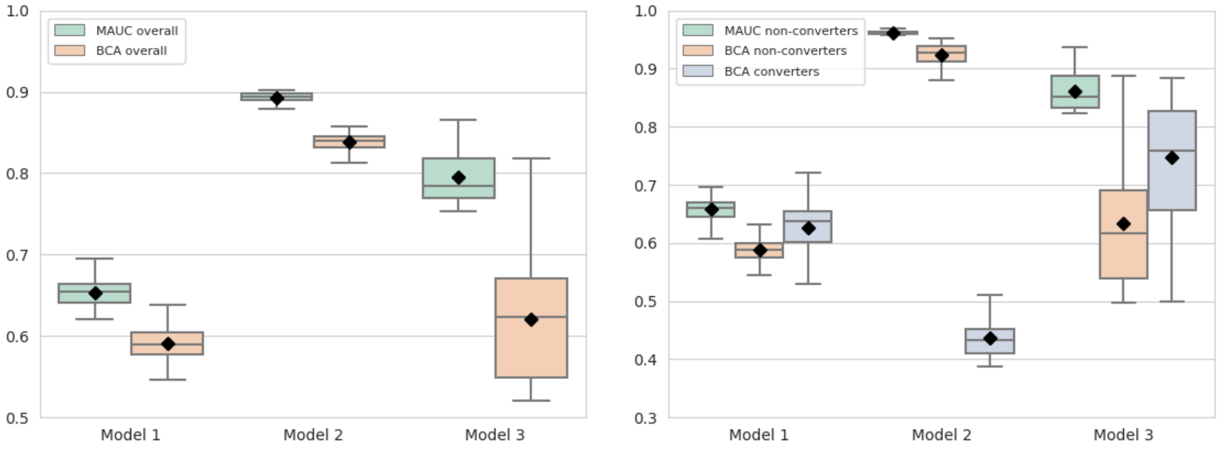


(a) Overall

(b) Separated

Figure A2: Boxplots of the MAUC and BCA scores of the three different models for predictions between two and three years in the future, both (a) overall and (b) separated for non-converters and converters.

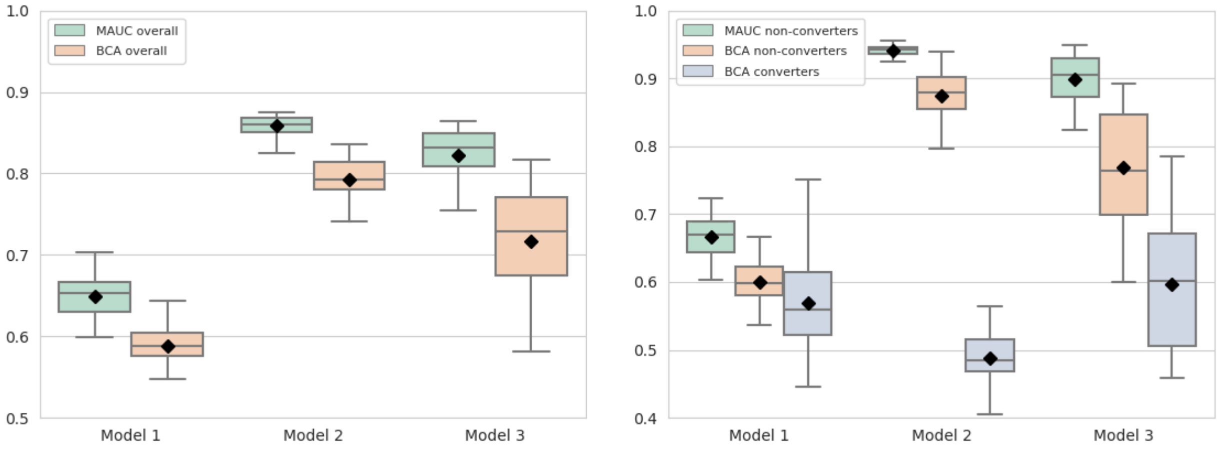
B Supplementary Parelsnoer Results



(a) Overall

(b) Separated

Figure B1: Boxplots of the MAUC and BCA scores of the three different models for predictions between one and two years in the future for the Parelsnoer data, both (a) overall and (b) separated for non-converters and converters.



(a) Overall

(b) Separated

Figure B2: Boxplots of the MAUC and BCA scores of the three different models for predictions between two and three years in the future for the Parelsnoer data, both (a) overall and (b) separated for non-converters and converters.