UNIVERSITY OF COPENHAGEN
FACULTY OF SCIENCE

Demographic bias in public neuroimaging databases, and its effect on AI systems for computer-aided diagnosis

Master thesis

Camilla Kergel Pedersen

Ku-id: wkl125

Supervised by: Aasa Feragen and Melanie Ganz-Benjaminsen

Submitted on: 31th of May 2021

Abstract

Deep learning is thriving, and neural networks are currently being developed for all kinds of computational problems. Recently it has been shown that a convolutional neural network performs very well on diagnosing Alzheimer's Disease (AD) patients based on their MRI scan [1]. However, the focus is often placed on computational time and performance when developing such a network. Little attention is often paid to the subjects' demographics and whether the training data is representative and balanced according to the demographics [2].

This project investigates whether a bias in ethnicity in the publicly available dataset ADNI will propagate as a bias in a model for a convolutional network.

A state-of-the-art convolutional neural network used to diagnose AD [1] is explored. The network is described and implemented in Python. Furthermore, a simpler network with comparable performance and a similar structure is presented. They were trained on different splits of preprocessed magnetic resonance (MRI) images from the American Alzheimer's Disease Neuroimaging Initiative (ADNI) [3] dataset. They yielded around 80-85 % accuracy and sensitivities above 65 % depending on the split and model. Using the models to predict AD on a test set from the Australian Imaging, Biomarker Lifestyle Flagship Study of Ageing (AIBL) [4] yielded a performance comparable to the AD prediction on the ADNI images. However, when classifying images from the Italian Alzheimer's Disease Neuroimaging Initiative dataset (I-ADNI) [5], all the models showed a worrying drop in performance. The best model could only achieve 78.0 % accuracy and sensitivity (almost no CN patients were present in the Italian dataset). In the worst case, the model only had the ability to classify the Italian patients randomly. Subsequently, it was found that the models in almost all cases have higher performance on MRI images taken at field strength 1.5T images varying up to over 10 % and that there were no significant differences in performance for the genders.

This project was challenged by the low amount of data from AIBL and I-ADNI and that the sites were very imbalanced concerning healthy controls (CN) and AD patients. Furthermore, it can be hard to ensure that no hidden bias in the different sites unexpectedly affects the results.

Conclusively the results suggest that the ethnic bias in the training dataset *did* propagate as a bias in the convolutional neural networks leading to a concerning drop in performance. More data should be included to investigate this further; however, this can have crucial consequences for how the ADNI data should be applied to deep learning in the future.

Contents

List of Figures

List of Tables

Glossary

AD Alzheimer's Disease.

ADNI Alzheimer's Disease Neuroimaging Initiative.

AI Artificial Intelligence.

AIBL The Australian ADNI dataset.

CN Cognitive Normal.

DARTEL A Fast Diffeomorphic Registration Algorithm. Used for normalizing MRI scans..

DTU Danish Tecnical University.

HPC High-Performance Computing - the name of the cluster used in this project.

I-ADNI The Italian ADNI dataset.

LSF Load Sharing Facility - the cluster used in this project.

MCI Mild Cognitive Impairment.

MNI Montreal Neurological Institute.

MRI Magnetic Resonance Imaging.

Six-layer model In this project, two neural networks are described, and this is the simple model that is used if not stated otherwise.

SPM Statistical Parametric Mapping.

t-SNE t-Distributed Stochastic Neighbor Embedding.

Thinlinc Server to store the code and data for this project..

1 Introduction

Systems using artificial intelligence (AI) and deep learning are becoming more and more popular for analyzing medical images. They are used so widely that their results may affect the overall ability to diagnose and treat different diseases. Therefore, many scientists are working hard to develop and improve these AI systems to get better performance. Much work is put into improving the AI algorithms, and little attention is often paid to the data used to train the AI . This creates a very high risk that the data-set contains an imbalance, which can cause unforeseen problems. This imbalance can occur for many reasons - some diseases might be more often found in one gender or be more frequent in specific parts of the world. The imbalance could also be due to some countries having better resources for healthcare. Since this bias can be based on ethnicity, geography, gender, or another demographic factor, this might result in specific groups of people getting a lower standard of healthcare due to the lack of training data. The motivation for this project is to investigate this hypothesis and raise awareness of the consequences of working with a biased dataset.

The data-set investigated is the publicly available ADNI [3] dataset, which is one of the most frequently used datasets in neuroimaging research and development. This dataset contains magnetic resonance (MRI) images of brains. It divides them into three target categories - either as having Alzheimer's Disease, having a mild cognitive impairment, or no cognitive impairment. A much investigated AI task, which will be the starting point for this project, is to diagnose the patients. This project focuses on the categorization of AD vs. CN patients using their structural MRI scans. Using a neural network trained on the American ADNI dataset, the performance will be evaluated using the American [3], Australian [4] and Italian [5] datasets for testing. Then it will be explored whether their performances are comparable.

In conclusion, it is stated that this project aims to reconstruct the neural network described in the paper by Basaia et al. [1] and train it on the American ADNI data-set. Then the trained network should be tested on ADNI data-sets from different parts of the world to see if there is a difference in performance.

The hypothesis is that a bias in the ADNI dataset, will propagate as a bias in the computer-aided diagnosis algorithms trained on the data. The bias will occur due to different circumstances worldwide, such as differences in access to health care, genetic differences, and contrasting lifestyles.

2 Background

2.1 Alzheimer's disease

Alzheimer's disease is an illness that results in ever-diminishing brain function and is the most common cause of dementia. The symptoms are failing memory, disorientation, mood and personality changes, and low ability to do everyday tasks. The disease progresses very slowly and comes insidiously. In the beginning, it can be hard to tell if there is even cause for concern, and many other diseases exist with similar symptoms [6].
Changes in the brain structure are some of the main features of Alzheimer's disease. However, little is actually know about what causes these changes. It is believed that a build-up of the protein Amyloid-beta, which is quite toxic to synapses, and another protein tau that tangles bundles of nerve fibers, is a major cause of these changes. This process begins up to 15-20 years before symptoms appear. Connections between the neurons are lost, and other complex brain changes occur, such as damage in the hippocampus, memory, and brain tissue shrinkage [6].
Worldwide 55 million people suffer from this disease, and only 1 out of 4 people with the disease get a diagnosis. Alzheimer's and dementia is the most common cause of disabilities for older people, [7] and every year 700.000 people die from this disease. Furthermore, 277 billion dollars was spent on this disease in 2018 in the U.S. Alzheimer's disease is the 6th leading cause of death in the U.S. and is the only disease of the 10 leading causes that cannot be cured or slowed down in any way [8].

Diagnosis of Alzheimer's

An MRI scan is one of many features used for diagnosing AD, and it is acquired together with a comprehensive clinical exam and blood and urine tests. The clinical examination focuses on previous medical history, the patient's ability to do everyday tasks, and recent changes in behavior. Furthermore, neuropsychological tests are conducted to investigate the patient's memory, attention, and ability to solve small problems.
Therefore it is entirely possible that a brain scan which highly suggests that the patients have AD will still be correctly classified as healthy by doctors if there are no clinical symptoms [9].

Diagnosing Alzheimers from an MRI

Even though it is impossible to diagnose Alzheimer's based solely on an MRI scan, it can be a very power full indicator, which is why it is possible for a neural network to detect the disease

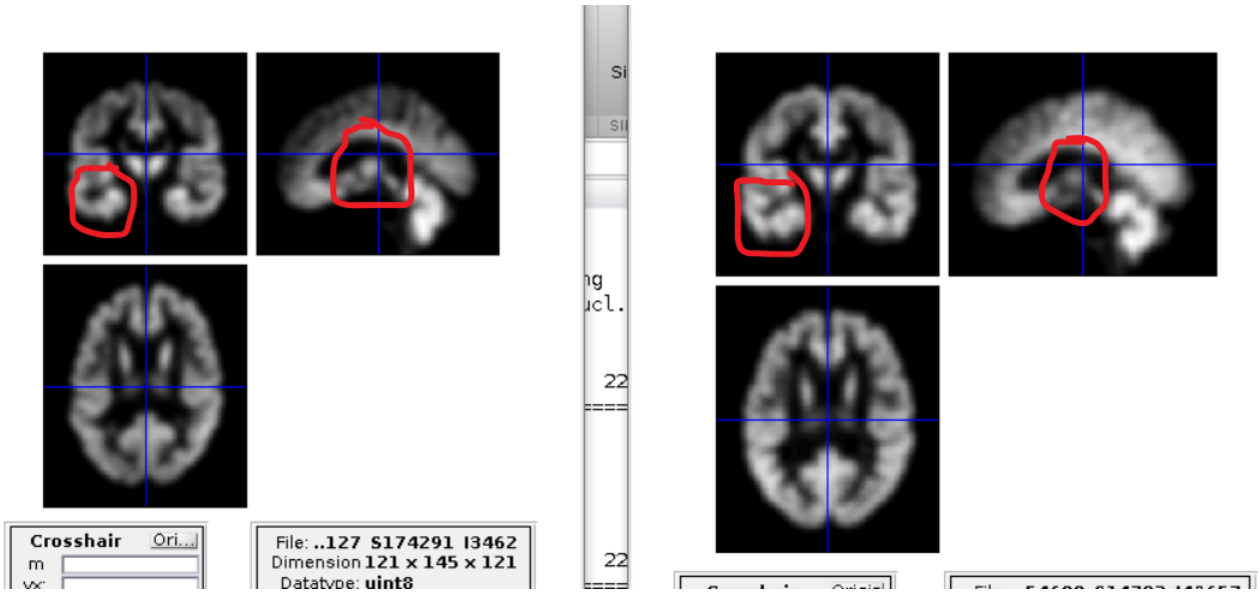


Figure 1: Two preprocessed MRI images viewed in SPM12 showing a Male AD patient aged 73 on the left and a healthy female patient aged 76 on the right. Differences in the amount of grey matter (brighter pixels) is clearly observed in the entorhinal cortex and hippocampus placed in the middle of the lower region of the brain. Note the red circles.

with very high accuracy. To investigate what kind of changes in the brain the network might pick up as features, two subjects are compared - one is labeled as healthy, and one is labeled as having AD . In general, both AD and normal aging have similar effects on the brain. The size of the ventricles increases, and the amount of grey matter decreases. However, with AD the grey matter is reduced in special regions such as the entorhinal cortex, medial temporal structure, and the hippocampus [10]. For visualization of this, see figure 1.
It is the changes in the brain structure that are in focus for this project. Using a convolutional neural network, it will be possible to diagnose the patients using 3D-MRI scans of their brains, and both now and in the future, this will be a tool for making a correct diagnosis. Therefore it is of uttermost importance that the network does not inherit a bias from the data set.

2.2 Differences in lifestyles for each country

Backing up the hypothesis that people from some countries might have a higher risk of getting Alzheimer's than others and thereby would contribute with more data, some of the risk factors of AD are investigated: smoking, diabetes, depression, obesity, and education [11] [12]. This is done by exploring the statistics of these risk factors for each country investigated in this project, which can be explored in table 1.

Country	BMI	Annual cigarette consumption	Diabetes (% of population)	Depression (% of population)	Education Index	Life expectancy
North America	28.5	1016.6	10.8 %	5.9 %	0.900	78.9
Australia	27.2	917.0	5.0 %	5.9 %	0.924	83.0
Italy	26	1493.3	5.6 %	5.1 %	0.793	83.3

Table 1: Statistics on risk factors for Alzheimer's disease from 2019 [13]

.

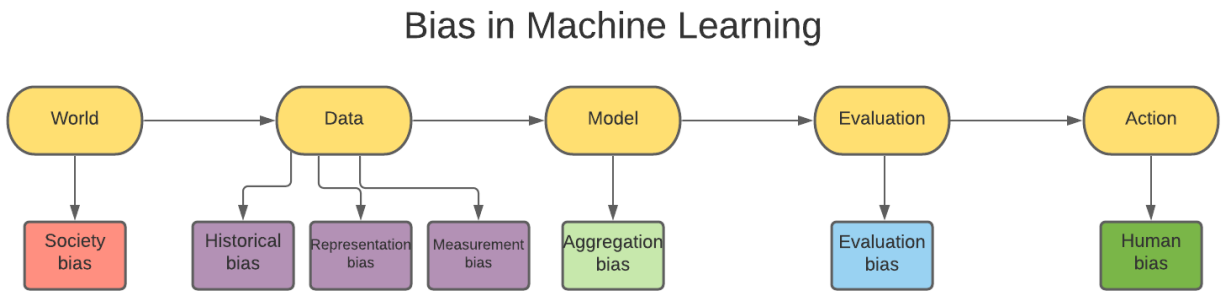


Figure 2: Overwiew of pipelines to make an AI , and where a potential bias might occur.

To summarize, obesity and diabetes rates are highest in the United States, which also have the lowest life expectancy. Australia has the lowest smoking rate and the highest education index. However, smoking rates are highest in Italy, which also has the lowest education index and lowest rate of depression.
Conclusively this table backs up the assumption that there is a large difference in lifestyles between the counties, affecting their risks of getting Alzheimer's disease.

2.3 Bias and Fairness in AI systems

Before this project is begun, it is important to define exactly what bias and fairness in AI systems refer to. This section is inspired by the definitions mentioned in [14].
First of all, an AI is concluded as fair if the performance is independent of given sensitive information about the subjects such as gender, ethnicity, or disability. The concept of bias goes hand in hand with this - the bias represents where the potential *unfairness* might originate from. There are roughly 5 levels of the pipeline to make an AI , which are overviewed in figure 2. A bias could be induced in all of them.
Firstly there is the bias in the world: the bias in the society will often show in the collected data. This bias is not something one can remedy, but it is crucial to be aware of. Next, there is the potential bias in the data acquired to train AI systems. A representation bias occurs from the way data is sampled - some feature classes might be unintentionally overrepresented.

The effect of the representation bias is the entire basis of this project.
Apart from the representation bias, there can be a *historical bias* based on historically disadvantaged groups, i.e., women and black people. Lastly, *measurement bias* can occur when there is a bias in how the subjects for the AI 's are chosen. Easily available data sets are not often representative of the real world, and data quality can vary across groups.
Bias can also occur in the model. *Aggregation bias* happens when subjects from distinct groups have some factors in common that one might be unaware of, and this factor is relevant to the given problem. This could, for example, be a predictor for an illness where an important feature is your height. The average height of a person is very different across the world. Using a single model for this would inherit this bias causing the model to overestimate some groups and underestimate others.
Evaluation bias occurs when the evaluation method for the model is not representative of the way the model should be used or the training data.
Lastly, there is the human bias occurring when evaluating predictions and taking action. Humans might disregard some correct predictions based on poor experience with that particular group of data.
All these biases might be represented intentionally (the representation bias) or unintentionally (all others) in this project. This will be elaborated on in the discussion.

2.4 Basics for magnetic resonance imaging

Nuclear magnetic resonance has been used in chemistry and physics for decades to study molecular structures. It was discovered in 1952 by Felix Bloch and Edward Purcell. Lauterbur and Mansfield developed magnetic resonance imaging (MRI) in 1973. MRI is one of the most important imaging techniques in medicine, relying on the properties of the spins of nuclei in the presence of an external magnetic field. It has excellent soft-tissue contrast and is often used for examining many parts of the body such as joints, organs, or the brain [15].

Magnetic field strength

During this project, it will be essential if the images in question for a given explanation are MR images acquired at magnetic field strength 3T or 1.5T. This refers to the magnetic field strength of an MRI image measured in a unit called Tesla, which is what the 'T' stands for. The higher the magnetic field strength, the higher is the signal-to-noise ratio, indicating that higher field strength means that the MRI is less noisy and of better quality.

MNI space

MNI stands for Montreal Neurological Institute, referring to the place where the MNI standard brain was determined. It was created by using many MRI scans from normal control patients and is a method for normalizing an MRI scan.
Before MNI, the Talairach atlas was used for normalization [16]. Talairach is a 3-dimensional coordinate system used to map different placements of structures in the brain independently of the individual differences between the shape and size using Brodmann areas.
Brodmann et al. [17] split the brainstem into different areas in 1900 based on the cells' size, shape, and density. It was later shown that these areas actually differ a lot from their surroundings in function. Thus they are still used. There are 52 areas in total.
Some issues have occurred with Talairach. The Brodmann areas are placed in a rather approximate way. The authors simply looked at some pictures of the Brodmann map and estimated where to place it in the Talairach atlas.
MNI wanted to create a brain more representative of the population, and they did so by using actual real MRI images. By a two-step procedure, they matched the template from Talairach - first, they took 241 scans and manually defined the landmarks that are also present in the Talairach coordinate system. They scaled each brain to match the coordinate system's positions and then took 305 more MRI images. Then they used an algorithm to match the brains to the 241 that were already matched to the Talairach atlas and had an average of 305 brain scans [16]. Thus they had created MNI 305, which is the space used for this project to normalize the images in the preprocessing.

2.5 Neural networks

When constructing a neural network, the main idea is to use the functions of a biological human brain and recreate the concept mathematically. The human brain consists of a lot of neurons that are connected by synapses. The neurons each represent different features, and the synapses represent the strength of the connections between the different neurons.
This knowledge is used when it is tried to model the human brain with a computer system. The neurons and the synapses are modeled, and it is attempted to teach the model how to adjust these connections while learning. The input to a neuron is most often described by a vector representing the input values. Since the neuron is described as a function, it uses the

input x a vector of weights w and a bias b to calculate an output value by the formula:

$$w \cdot x + b$$

Related to the previously described similarities to the human brain, it can be said that w describes the synapses and their strength, x defines the output of the other neurons and b adjusts the sensitivity for when the neurons should fire.
Usually, a neural network consists of layers of neurons that are connected. The input is then sent through these layers during training, and the neurons and connections are adjusted accordingly through backpropagation. One frequent layer is a fully connected layer, consisting of a set of neurons all connected to the neurons in the next layer. Because of the many connections, they can be computationally expensive, especially for large datasets. When parsing input through a fully connected layer, the output is equal to: $output = activation(dot(x, w) + b$ like described earlier. There are many different activation functions, but the one used for this project is the ReLU activation function:

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

For the output layer the sigmoid/logistic function is used:

$$f(x) = \frac{1}{1 + e^{-x}}$$

since this squeezes the output value down to between 0 and 1, representing the probability that the subject belongs to a class. If this output is above 0.5, the subject belongs to the class, otherwise not. A project was created earlier in my education explaining all of the basics of a neural network in a lot more detail [18].
There are lots of different kinds of neural networks depending on the type of the problem. For this project, where the type of data is images, a convolutional neural network is implemented.

Convolution

Doing convolution over an image is the central part of why a CNN works. The goal is that the network can learn how nearby pixels are correlated rather than treating each connection between pixels in the same way as it would for a fully connected network. For an ordinary 2D image, this means to run a filter ($N \times N$) matrix over the pixels of the image to make a "mathematical summary" of the most important pixels. Each value in the filter determines the weight of the corresponding pixel in the output image.

For example, the horizontal Sobel filter, used to detect horizontal edges is studied:

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

A horizontal edge in an image will be visible in the pixels by having a large difference between two consecutive horizontal rows of pixels, indicating a significant change in the colors. A straightforward piece of a grey-scale example image is studied:

$$\begin{bmatrix} 0.2 & 0.2 & 0.2 \\ 0.3 & 0.3 & 0.3 \\ 0.4 & 0.4 & 0.4 \end{bmatrix}$$

Running the Sobel filter over this will give $-1 \cdot 0.2 + (-2) \cdot 0.2 + (-1) \cdot 0.2 + 3(0 \cdot 0.3) + 1 \cdot 0.4 + 2 \cdot 0.4 + 1 \cdot 0.4 = 1$ indicating a very high probability of an edge in these pixels. For a different example, it could be a high probability of an eye, a dog, or a face. For this example, the output will just be a value, but the output will usually be a smaller image in practice.
Usually, the image will be much larger than the filter, so the filter is run over the image horizontally and vertically and get another image out. See figure 3. Doing this repeatedly through several convolutional layers followed by fully connected layers in a CNN will eventually result in a single or very few values indicating what category an image belongs to.

For this project, 3D convolution is applied, which is a very similar concept. The only difference is that instead of using 2D matrices as images and filters, they are 3D and moving along all 3 axes in the images.
In a CNN, the values in the filter are not defined like in the previous examples. It is learned during the training process, and therefore the expectation is that the filters become specially designed for the given problem. Often, the first convolutional layer in a CNN will detect the main features - this could be the contours of the brain. The next layer will then maybe detect the different structures in the brain, and the next layer will detect abnormalities inside these structures and so on - exactly like human brains learn to identify objects.

Channels

Another aspect of doing convolutions is the term channels. Like the word insinuates, this means that each pixel can contain different kinds of information. For example, there will be

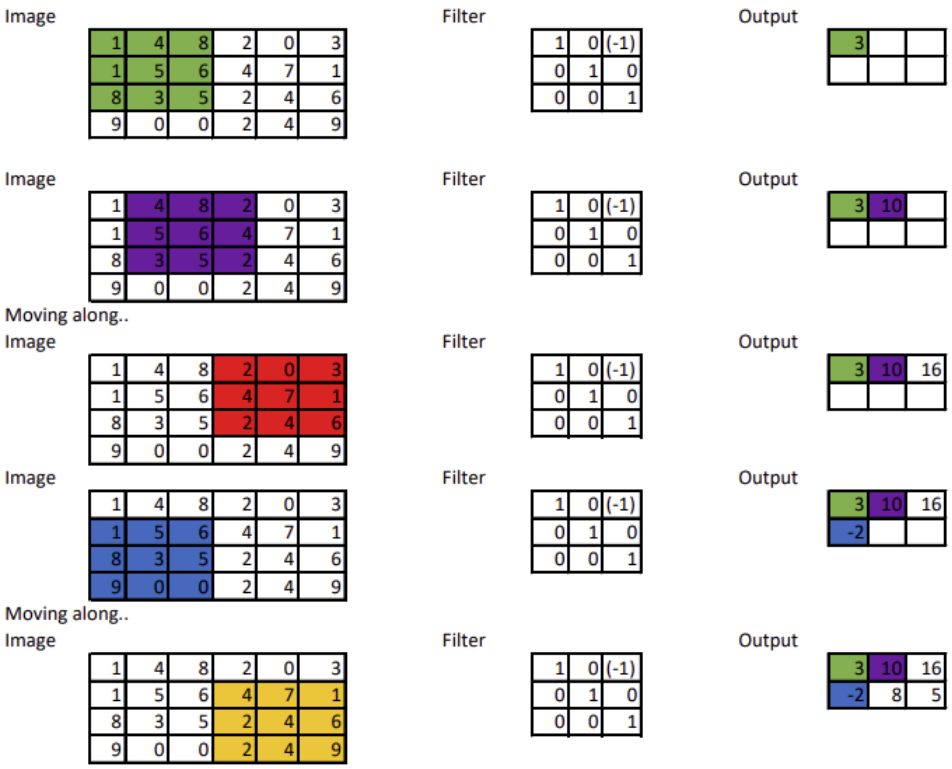


Figure 3: Figure showing an example of a convolution step-wise (Assuming there is no zero padding and stride=1). Running the filter over the image row-wise and column-wise, where each color in the output corresponds to the pixels marked by that color in the image.

three channels for each pixel in an RGB image - the amount of blue, green, and red color, respectively. When processing an image through a CNN, the number of output channels for each layer can be chosen, and the intention is that different channels might store different parts of the information from the convolution.

Loss function

Basically, the goal during training for the network is to minimize the loss function, which describes mathematically how good the model's performance is at a given time (epoch). Calculating the optimal weights is impossible as there are too many unknown parameters. Instead, an optimization problem is created searching for the weights that minimize the loss. The problem in this project is a binary classification problem. So the loss function used is the binary cross-entropy. This is the formula for multiclass cross-entropy:

$$CE = -\sum_{i}^{C} t_i \log(s_i)$$

where t_i and s_i are the ground truth and the prediction stating the probability that the subject belongs to class i. The idea behind taking the logarithm is that a large difference between t_i and s_i will yield a large penalty and vice versa.
Because our problem is binary - either a patient is healthy or has AD - $C = 2$ and it can be rewritten as:

$$CE = -\sum_{i}^{2} t_i \log(s_i) = -t_1 \log(s_1) - t_2 \log(s_2) = -t_1 \log(s_1) - (1 - t_1) \log(1 - (s_1))$$

Since it is known that the probability of a person is healthy must be $1 - p$, where p is the probability that the patient is sick.

Optimizer

The optimizer should adapt the weights such that the loss is minimized. There are several methods to do this. The simplest and most well-known one is gradient descent. That simply works by taking the first-order derivative of the loss function and taking a step in the direction of that derivative. The step size depends on the learning rate.
The problem with gradient descend is that the learning rate is not adapted, and it can get

stuck in a local minimum. Also, it can easily begin to oscillate if the learning rate is not set appropriately. Furthermore, it is prolonged on large datasets. This is because of the need to take the derivative of the loss function with respect to each data point, and each data point might also have several features. This will take an enormous amount of computations.
Therefore one would mostly use a development of gradient descend called stochastic gradient descend. The idea behind SGD is to induce some randomness into the model, so all the data points are not used for the calculation each time a step is taken. SGD randomly chooses a point for each iteration and only uses that to determine the direction of a step. However, the downfall here is that SGD is very sensitive to outliers since one odd data sample will have a tremendous impact on the direction of the gradient.
Instead of just sampling one data point, one can consider sampling a few data points called minibatch gradient descent, combining SGD and GD, trying to achieve the best from both worlds.
The oscillation of SGD can make it hard to reach convergence. So a method called momentum is introduced to reduce oscillations and help the acceleration of SGD in the relevant direction. The idea is to add a fraction γ to the update vector of the past timestep to the current update vector, so more than just the last update step can be taken into account:

$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta)$$

A huge problem with all the previously mentioned methods is that the learning rate is fixed during the training. In general, having a small fixed learning rate will make computation very slow as one will have to take many steps, and having a learning rate that is too high could cause one to jump over the minima. Also, the non-convex nature of a neural network can lead the perfect learning rate in one direction to be too small or too large in another direction. The ideal situation is to have an adaptive learning rate that is large when one is far from the minima and gradually smaller as one approaches the minima. This is what ADAM [19] tries to achieve by using extensions of SGD.
The first one is AdaGrad which introduces the adaptive learning rate by incorporating knowledge of previously observed data. Larger learning rates are used for parameters with features that are infrequent and lower learning rates for frequent ones such that each parameter has its own

learning rate.

$$\mathrm{w_t} = \mathrm{w_{t-1}} - \eta'_{\mathrm{t}} \frac{\partial \mathrm{L}}{\partial \mathrm{w_{t-1}}}$$
$$\text{where } \eta'_{\mathrm{t}} = \frac{\eta}{\sqrt{\alpha_t + \varepsilon}}$$
$$\text{and } \alpha_t = \sum_{i=1}^{t} \left(\frac{\partial \mathrm{L}}{\partial \mathrm{w_{t-1}}} \right)^2$$

A weakness here is that the learning rate for each parameter is always decreasing since every term in α_t will always be positive. This can lead to a point where the model simply stops learning even before the minimum is reached. A further improvement with AdaDelta is that instead of just adding past squared gradients, the window of accumulated past gradients is restricted to a fixed size. The sum of gradients is defined as a decaying average of all past squared gradients instead of just previously stored gradients. So the running average at one timestep depends only on the previous average and the current gradient.
The last thing needed is to calculate the momentum changes by making the momentum, from SGD with momentum, adaptive, which ADAM does as it stands for Adaptive Moment Estimation. So now both the learning rate and the value of the momentum are based on the given parameter and previous observations meaning that the ideal conditions for faster convergence and a good tuning of hyper-parameters is present which is ideal when implementing a neural network, where there are so many options for a design.

2.6 Evaluation metrics

For evaluating the performance of the network, three evaluation metrics were used: Accuracy, sensitivity, and specificity. They are defined as follows:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Se = \frac{TP}{TP + FN}$$

$$Sp = \frac{TN}{TN + FP}$$

where T and F are true and false, and P and N are positives and negatives. In practice, the accuracy evaluates the overall performance. The sensitivity measures the ability to identify if a subject belongs to a class, and specificity is the ability to measure if the subject does not belong to the class. Putting this in perspective, it means that a model classifying everyone

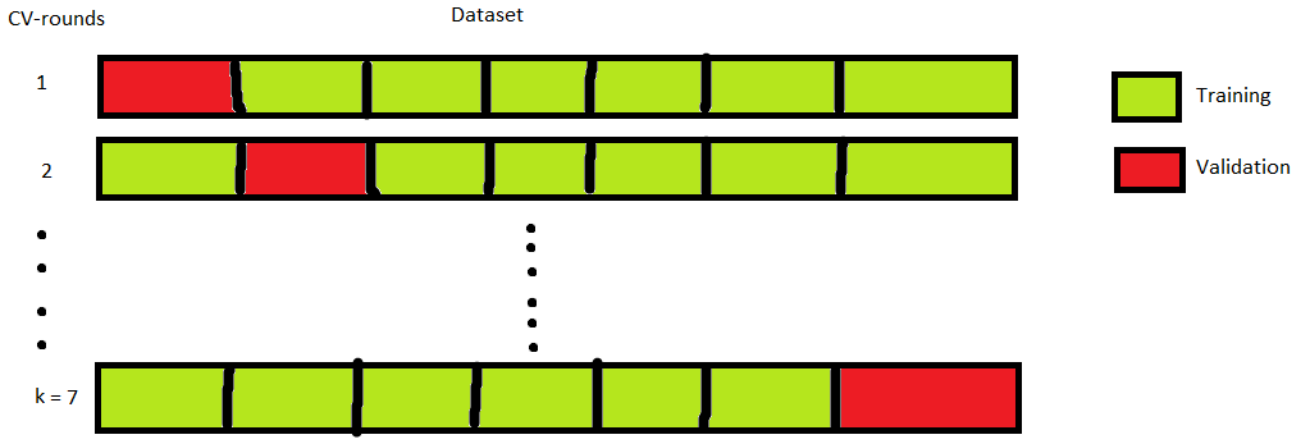


Figure 4: Figure showing the principle of cross-validation. Each horizontal line represents the data set for one round, and the colors mark how it is split into training and validation set.

as having AD would have a very high sensitivity and a very low specificity. Specificity and sensitivity are essential when the test set is imbalanced, because this can bias the accuracy if the model is biased towards one class.

2.7 Cross-validation

To evaluate the final model for this project, cross-validation was used [20]. Cross-validation is a way to validate the robustness of a machine learning model and how it will react to new data. The idea is to do a K-fold split of the training and validation data (The test data is kept out of it). Each split consists of normally 15-20% of the data for validation and the rest for training. Then the model is trained k times and evaluated on the validation part of the split, see figure 4, and average performance is found. This is done to ensure the stability and robustness of the model since this is a way to prove that the performance of the model is not dependent on the choice of training and validation data.

2.8 Data augmentation

Data augmentation is a way to create more data for training. It works by editing the images already present in the data-set in a small way that changes the distributions of the pixels while still maintaining the motive such that it resembles the data the model could be exposed to in the future. This can, for example, be done by small rotations or cropping. For this project it was important that the images still resembled a brain even after the editing.

2.9 T-SNE

t-Distributed Stochastic Neighbor Embedding (T-SNE) is an unsupervised non-linear algorithm used for visualizing high dimensional data in a low dimensional space. It is used to get an intuition of how the data is arranged. It was developed by Laurens van der Maatens and Geoffrey Hinton in 2008. It is different from another well-known visualization method, PCA [21]. While PCA likes to maximize variance and preserve large pairwise distances, T-SNE only keeps the small pairwise distances. In practice, this means T-SNE will tend to cluster data that have local similarities, whereas PCA will prioritize to separate dissimilar data. With T-SNE, it will be easier to spot the local variances between the images, which is the desire for this project.
T-SNE calculates the similarity between two points in both the high-dimensional and low dimensional space and then uses 3 steps to optimize the two similarities using a cost-function.

- First the similarities in the high dimensional space are measured. Each data point becomes a center of a gaussian distribution and only similarities to other datapoints within this distribution are calculated. For each point a density is measured (the upper part of the fraction) and renormalized (lower part of the fraction):

$$p_{ij} = \frac{\exp\left(-\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2 / 2\sigma^2\right)}{\sum_k \sum_{l \neq k} \exp\left(-\left\|\mathbf{x}_k - \mathbf{x}_l\right\|^2 / 2\sigma^2\right)}$$

  This results in a probability p_{ij} that measures the distance between two points in the high dimensional space given as a probability. The bandwidth of the gaussian is set such that a fixed number of points falls within this bandwidth, which is because different parts of the space may have different densities. So Actually p_{ij} is computed like this:

$$p_{j|i} = \frac{\exp\left(-\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2 / 2\sigma_i^2\right)}{\sum_{j' \neq i} \exp\left(-\left\|\mathbf{x}_i - \mathbf{x}_{j'}\right\|^2 / 2\sigma_i^2\right)}$$

  Finally the conditional probabilities are symmetrized giving an overall average:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$$

- Each data point is now represented as a point in the low dimensional space and the

process is repeated, however with a student-t distribution, now calculating q_{ij}:

$$q_{ij} = \frac{\left(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2\right)^{-1}}{\sum_k \sum_{l \neq k} \left(1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2\right)^{-1}}$$

The reason for using the student-t distribution is the fact that is has a heavier tail that allows for better modeling of far apart distances.

- The desire is for q_{ij} to reflect p_{ij} as well as possible. If they are roughly the same, the structure must be roughly the same in both the high and low dimensionality. Kullback Liebler (KL) is used which is a measure for differences between probabilities:

$$KL(P\|Q) = \sum_i \sum_{j \neq i} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

The desire is to minimize this function, which is done using gradient descent which gradually moves the points around until KL is minimized. The reason for using KL is the fact that it penalizes when p_{ij} is large and q_{ij} is not, but not the other way around. This is exactly why T-SNE only preserves the local structure of the data.

For this project, T-SNE was used at the very end to visualize the output data from one of the finals layers in the network to investigate if there were differences in the image qualities between different splits of the data (field strength, gender, diagnosis..).

3 Methods

3.1 Datasets

ADNI

The datasets from ADNI used for this project were the ADNI1_Baseline_3T collection containing 199 3T images and all baseline 3T images from ADNI2. They had a total of 840 images. Finally, 740 1.5T images from ADNI1 where also included. The overview .csv file from downloading the data was used to classify each image as they included the subject ID of each patient along with their diagnosis, which could either be Healthy (CN) or mild cognitive impairment (MCI) or Alzheimer's (AD). Furthermore, the image ID from this file was used to ensure the identity of the image.
All subjects in ADNI2 that only had SAG_IR-SPGR scans where excluded, leaving 624 ADNI2 images for inclusion.
In the ADNI1 baseline3T collection, 48 subjects had two MP-RAGE images, where one was N3_SCALED_2, and the other was N3_SCALED. Each subject had an N3_SCALED, so for consistency, these were used for each subject. So in total, ADNI1 baseline3T consisted of 151 unique subjects.
All MCI patients were excluded as the model in this project only districts Alzheimer's patients from healthy patients.
All images were baseline 3D T1-weighted (MPRAGE) sequences.
The demographics of the entire dataset can be found in table 2.

AIBL

The Australian ADNI (AIBL) dataset was used for testing the final model. It consisted of 662 subjects whose demographics can be found in table 3. Only MCI images from this dataset were disregarded.

Italian ADNI

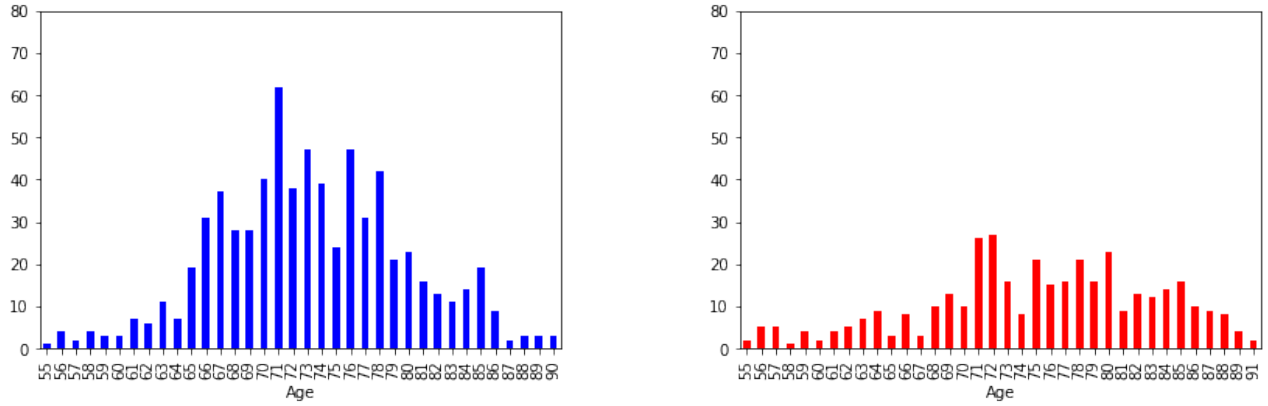
The I-ADNI dataset was used for testing the final model but never for training. It consisted of 181 subjects whose demographics can be found in table 4. There were no MCI patients in this dataset.

3.2 Demographics of the subjects

In table 2 and 3 and figure 5 and 6 the demographics of the subjects in the sites can be viewed. Considering that the MCI patients are excluded, is it observed that ADNI has quite an overflow of 1.5 T images, a quite good balance between men and women and in age, and again high imbalance in diagnosis. This case is even worse for AIBL that only has approximately 15% AD patients. AIBL also consists primarily of 3T images and has an acceptable balance in gender. I-ADNI whose demographics are in table 4 was also very imbalanced regarding diagnosis containing only two CN patients. An imbalance in gender was also present, only containing approximately 33% women. The age difference between AD and CN patients in I-ADNI is inconclusive because of the low number of CN patients.
The age distributions of all three sites are almost the same; however, the average age in AIBL and I-ADNI is slightly younger than ADNI. In both AIBL and ADNI, the age for AD patients is higher than the age for the healthy controls.
Lastly, the differences in the number of images for each site is noted. Where ADNI has over 1000 images excluding MCI, AIBL only has just above 500, and I-ADNI does not even have 200.

	CN	AD	MCI	other	Total
N	698	377	634	473	2182
Male/Female	300/398	196/181	389/245	243/229	1128/1053
Age avg.	75.1	73.24	74.1	71.2	73.3
1.5T/3T	505/193	235/142	564/70	104/369	1408/774

Table 2: Demographics of the subjects from ADNI1 and ADNI2.



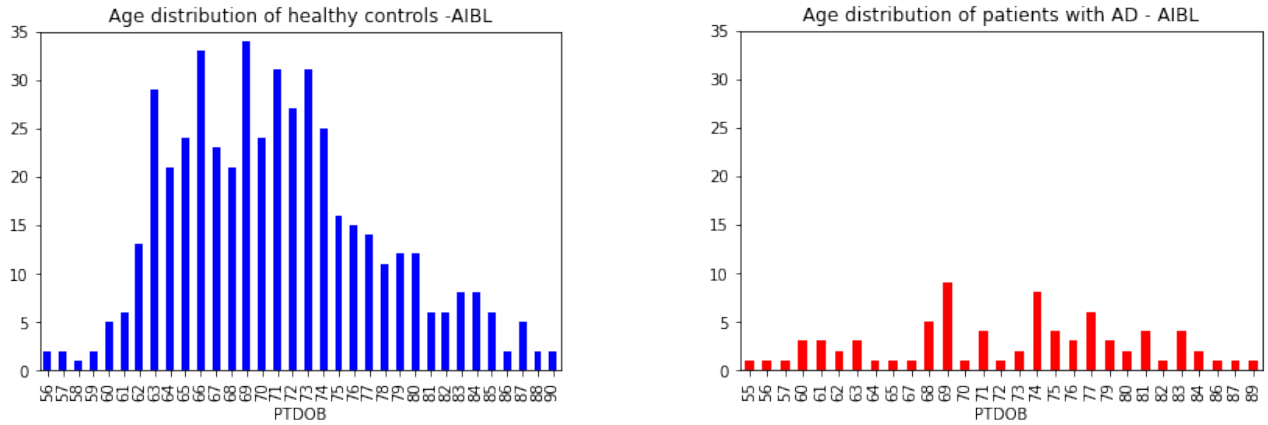
(a) **(b)**

Figure 5: Age distribution of healthy controls and Alzheimer's patients in ADNI

Australian ADNI - AIBL

	CN	AD	MCI	Total
N	479	79	104	662
Male/Female	203/276	33/46	56/48	292/370
Age avg.	70.9	72.4	73.52	71.53
1.5T/3.0T	87/392	12/67	17/87	116/549

Table 3: Demographics of the subjects from the australian ADNI.



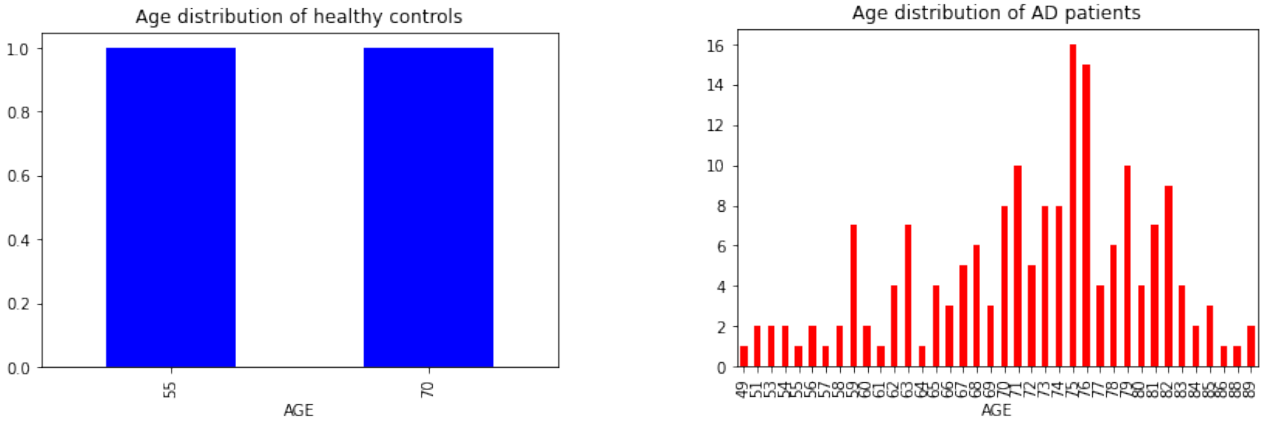
(a) **(b)**

Figure 6: Age distribution of healthy controls and Alzheimer's patients in the AIBL dataset

Italian ADNI

	CN	AD	MCI	Total
N	2	179	0	181
Male/Female	2/0	114/65	0/0	292/370
Age avg.	62.5	72.1	N/A	72.0
1.5T/3.0T	0/73	2/106	0/0	108/73

Table 4: Demographics of the subjects from the Italian ADNI.



(a) **(b)**

Figure 7: Age distribution of healthy controls and Alzheimer's patients in the Italian dataset

Demographics regarding magnetic field strength and gender

Since field strength and gender for each image are also factors tested regarding a bias, schemes are also made for an overview of their correlations. See table 5.

ADNI

Gender	1.5T/3T
F	405/174
M	335/161

AIBL

Gender	1.5T/3T
F	57/314
M	58/235

I-ADNI

Gender	1.5T/3T
F	64/50
M	44/23

Table 5: Demographics regarding gender and field strength.

3.3 The paper from Basaia et al.

The neural network in this project is heavily inspired by the network described in the paper by Basaia et al. called "Automated classification of Alzheimer's disease and mild cognitive

impairment using a single MRI and deep neural networks" [1]. The details of the architecture are described in section 3.7. The network is based on such a paper, instead of developing a model from scratch, since the desire is to investigate bias in current and active research and ensure that the evaluated model is relevant in the research field. Furthermore, this gives an intuition on what kind of architecture that works for this kind of problem.
Summarizing, Basaia et al. created a convolutional neural network which binary discriminated between CN, converting MCI (Mild Cognitive Impairment), stabile MCI and AD. They used all 3T T1-weighted baseline images from ADNI1 and ADNI2, and in total, they had 407 healthy controls, 280 converting MCI, 533 stable MCI and 418 AD patients.
They present models for converting MCI vs CN, stabile MCI vs CN, AD vs converting MCI, AD vs stabile MCI, and lastly converting MCI vs stabile MCI . Before they feed the images into the network, they are preprocessed in spm12, ultimately normalizing them to MNI305 space using the DARTEL algorithm, described in heavier detail in section 3.5. Afterward, they developed a convolutional neural network, ultimately performing with 99% accuracy on the model classifying CN from AD. The worst performing model, which was the last-mentioned performed with 75% accuracy. They validated all models using cross-validation and used transfer learning by using the weights from the CN vs. AD model to initialize the other models. This process of reproducing the network solving the AD vs. CN classification problem is one of the main goals for this project.

3.4 Tools

This section describes all tools that were used to complete this project. A full overview can be found in figure 8.

The Thinlinc server

Thinlinc is a cross-platform remote desktop server used by DTU for making remote access to their computer systems available to all students. The server was utilized to store all the data from the sites and store code and models. Since the data from ADNI could not be downloaded onto a local machine because of privacy issues and Thinlincs high storage space, the server was very beneficial for this project.

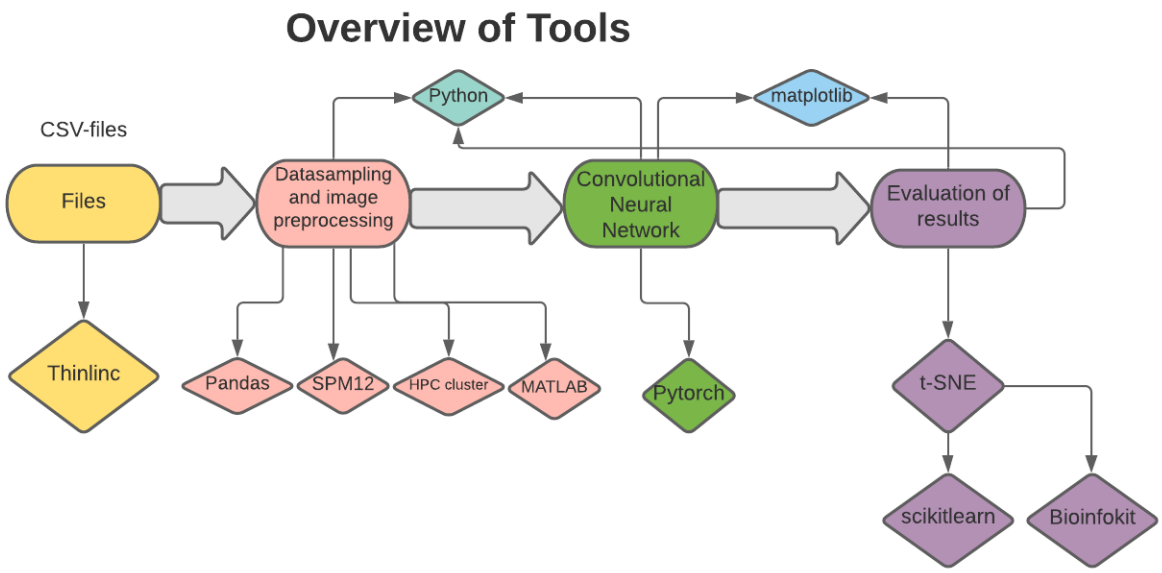


Figure 8: Overview of when each tool was used during the development of this project.

SPM12

Before the network can be applied, the data from ADNI had to be preprocessed using a tool called SPM [22], which is a software that contains many algorithms for structural and interface enhancements of MRI images. In this specific case, it was needed to normalize the images to MNI space using the DARTEL algorithm. The reason why the DARTEL algorithm is used for preprocessing is the allowance for a more accurate inter-subject registration of brain images. Studies have shown that the DARTEL method has a very good normalization performance [23].

HPC cluster

To run the code for this project, a HPC LSF cluster from DTU was used [24]. It ensures the distribution of the available resources between the users by a queuing system according to what the users need and what is available. HPC is a multiuser environment, where the user does not run the code on a local system but instead asks the cluster to run it utilizing a job script telling the cluster which application to run. Using the cluster was beneficial for this project as a powerful GPU was needed for the network, which was unavailable locally. Furthermore, some of the preprocessing in SPM12 took several days, and it would have been unfavorable to run interactively as that is much more vulnerable to crashes than the cluster.

Python and MATLAB

Python3 was used as the primary programming language for this project. The neural network, and all tests and visualizations connected to it, was written in python. Python was also used for several scripts to create the batch files for SPM12 and list the filenames for the preprocessed images to feed into the network. MATLAB was briefly used to run the batch files from SPM12 on the cluster.

PyTorch

PyTorch [25] is a library implemented in python for deep learning utilizing both GPU's and CPU's depending on how computationally heavy the deep learning task is. This library was used for implementing the network with the architecture described in 3.7 and in particular the following functions where used:
The 3D-convolutional-layers where implemented using the build-in function `Conv3d(in_channels, out_channels, kernel_size, stride, padding, ..)` whose return value can be defined as [26]:

$$\text{out}\left(N_i, C_{\text{out}_j}\right) = \text{bias}\left(C_{\text{out}_j}\right) + \sum_{k=0}^{C_{in}-1} \text{weight}\left(C_{\text{out}_j}, k\right) \star \text{ input } (N_i, k)$$

Where $\star$ is the is the valid 3D cross-correlation operator, C is the number of channels and N is the batch-size.
Another crucial functionality of PyTorch is the Dataloader and Dataset classes. The Dataset class allows for storing the data, and the Dataloader wraps an iterable around the data objects, making them easier to access. The entire data set was impossible to store as an array in memory due to its size, so utilizing these classes was essential for feeding the data to the network.
The build-in method for `Relu, batch-normalization, fully connected layers, sigmoid` and `dropout` was also used. All terms discussed above.

Further mentions

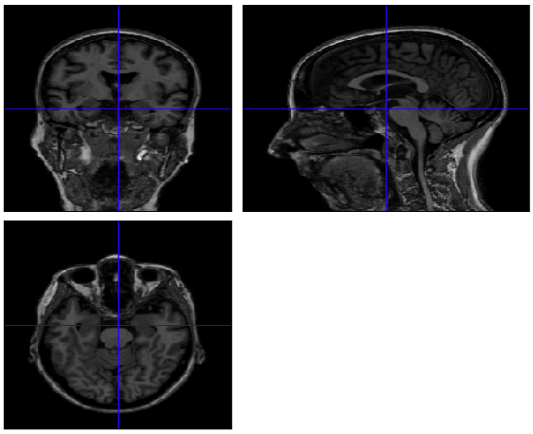
Matplotlib [27] was used throughout the entire project to visualize the loss plots. SKlearn [28] and bioinfokit [29] was used to do a T-SNE analysis at the very end of this project and visualize the results using different color splits. Pandas [30] was used for loading in data from the files containing information such as diagnosis, gender, and age.

3.5 The prepossessing steps

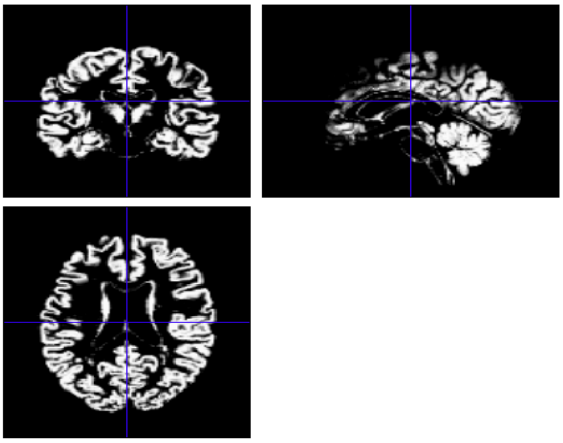
To follow the steps from the paper [1] the following steps had to be applied on each image [31] using SPM12:

- The T1-weighted images should be segmented to produce GM, white matter (WM), and cerebrospinal fluid (CSF) tissue probability maps in the Montreal Neurological Institute (MNI) space.
- The segmentation parameters obtained from the first step should be imported in DARTEL.
- The rigidly aligned versions of the images segmented in step 1 should be generated to create an "average space" over the entire dataset fitting the images to each other.
- The DARTEL template can now be created, and the obtained flow fields are applied to the modulated 3D T1-weighted images of single subjects (generated by the segmentation step). They are warped to the common DARTEL space and modulated using the Jacobian determinants.
- The modulated 3D T1- weighted images from DARTEL are normalized to the MNI template using an affine transformation estimated from the DARTEL GM template and then a priori GM probability map without resampling.

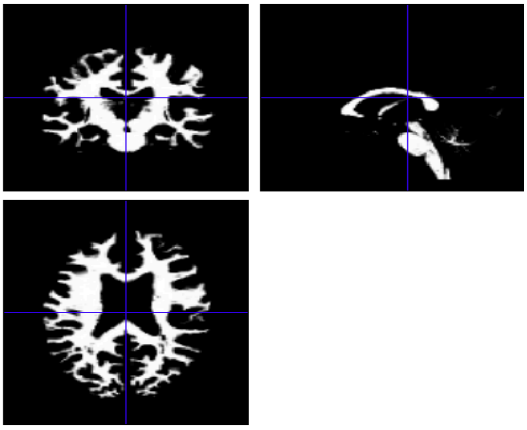
The images from each of the steps are shown in figure 9.



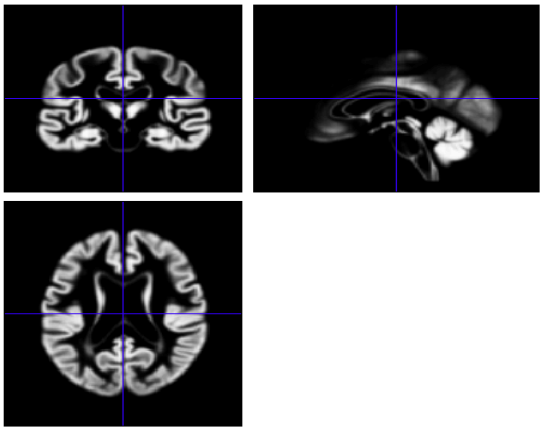
(a) The original MRI-image before any preprocessing had begun.



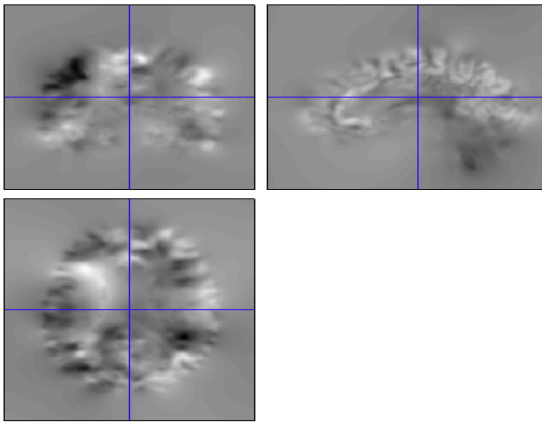
(b) DARTEL segmentation of grey matter.



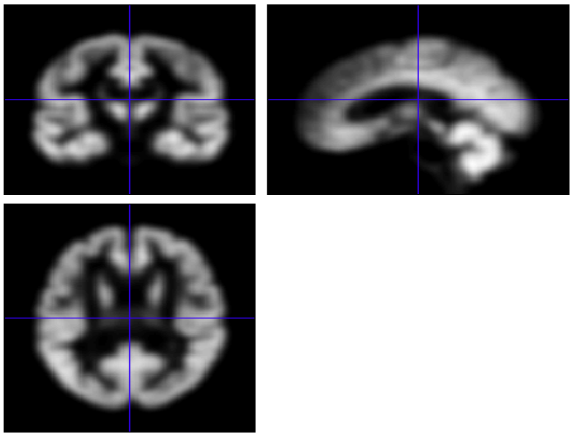
(c) DARTEL segmentation of white matter.



(d) The final average template (template 6) for all the 3.0 Tesla MRI images in ADNI1 and ADNI2



(e) The flow field.



(f) Final preprocessed image that is normalized to MNI space.

Figure 9: The steps of processing a single image from the original image to MNI. All the images are from the same subject

3.6 Batch script

SPM12 (Statistical Parametric Mapping) was used to complete the steps above. Karl Friston originally wrote SPM in MATLAB in 1991, and SPM12 is a significant update to the SPM software used in this project to analyze the fMRI images. In SPM12, it is possible to make a script applying different analytical and statistical processes to the images in a particular order which was precisely needed for this project to follow the steps from Basaia et al.
Firstly, a DARTEL segmentation (Diffeomorphic Anatomical Registration Through Exponentiated Lie Algebra [32]) of the grey and white matter on each of the 3D images was acquired. The batch editor was used to create the before mentioned script, and the field native tissue was set to "DARTEL imported" for the first two tissue types, which are the segmentations of the grey and white matter. Then this batch script was saved as a template, and a python script was written to produce a batch script containing all the filenames for the relevant images.
Running this batch-script on every image is computationally expensive, and therefore the HPC -cluster was used. Usage of this cluster is described in section 3.4. After acquiring the DARTEL segmentations, the DARTEL templates were created using every DARTEL -segmented image from the previous step to create an average template from the entire dataset. The template is modified for each image iteratively to align it to the dataset. So another batch script for generating the template was created, and a python script was written to include all of the paths for the image files.
Lastly, the images were normalized to MNI space using the template, the flowfields, and all of the segmentations of grey matter from the previous step. This required a third batch script.
It is an important decision which images should be preprocessed together and thereby influence the average template that the DARTEL algorithm creates. The following choices were made:

- All 3T images from ADNI were preprocessed together in one folder.
- All 1.5T images from ADNI are preprocessed together in a different folder.
- All AIBL images are preprocessed together.
- All I-ADNI images are preprocessed together.

These choices are a compromise of wanting the average template to represent the entire dataset and not be too influenced by data that is not present in the current experiment. 3T-images from ADNI were preprocessed together as the initial idea was only to use those. However, this split was not done for AIBL and I-ADNI since there were many cases where the test set was

a mix of field strengths. Furthermore, creating the templates is time-consuming, and using different templates for preprocessing each image, would easily create confusion in the structure of the data.

3.7 The architecture of the neural network

Parameters

For both the networks described in the following sections, some parameters were the same. The optimizer used was ADAM, the loss function was the (Binary) Cross-entropy Loss, the learning rate was 0.0001, and the batch size was 3. Basaia's model was trained for 200 epochs and the development model for 125 epochs.

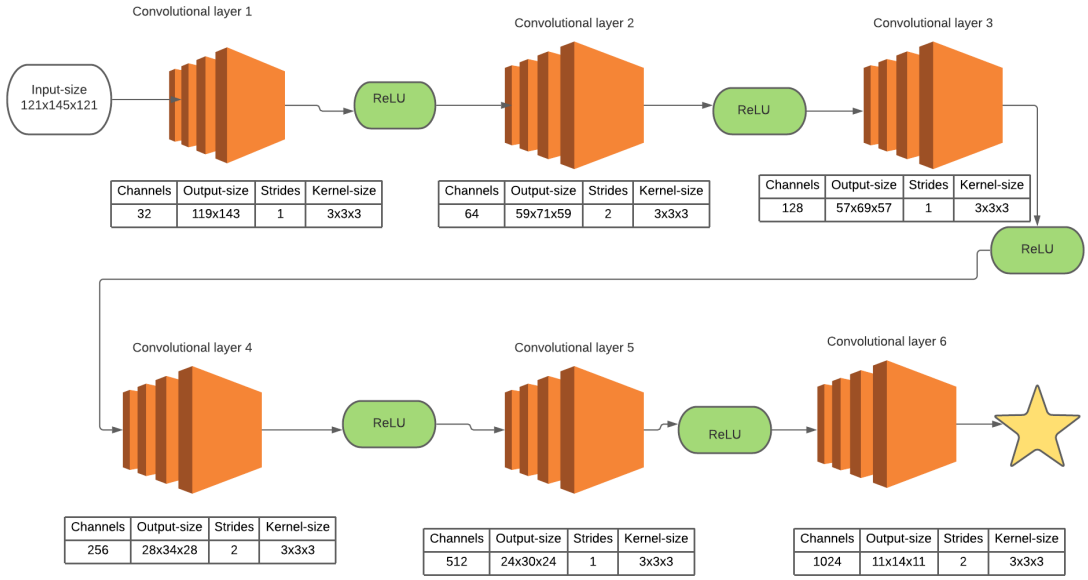
The network from Basaia et al.

The goal for the design of the network is to resemble the architecture described in the paper by Basaia et al. That network should consist of 12 convolutional layers, where the first two layers have 50 output channels with kernel-size $5 \times 5 \times 5$ with alternating strides 1 and 2. They should be followed by ten layers with 100 to 1600 channels with alternating strides 1 and 2 and a filter size of $3 \times 3 \times 3$. Each layer should use ReLU as an activation function, and lastly, there should be a fully connected layer and a logistic regression (sigmoid) layer as the output layer. The paper does not mention use of zero padding, but the kernels eventually become too small before the image is sent through every layer if zero padding is not applied. The network is analyzed and described layer by layer to understand each modification to the input. First assuming that zero padding is not applied.
Each image has dimensions 121×145121, and one channel since the image is grey-scale. If the first layer of the network has 50 kernels, a kernel size of 5, and a stride of 1, the output from that layer will be of size:

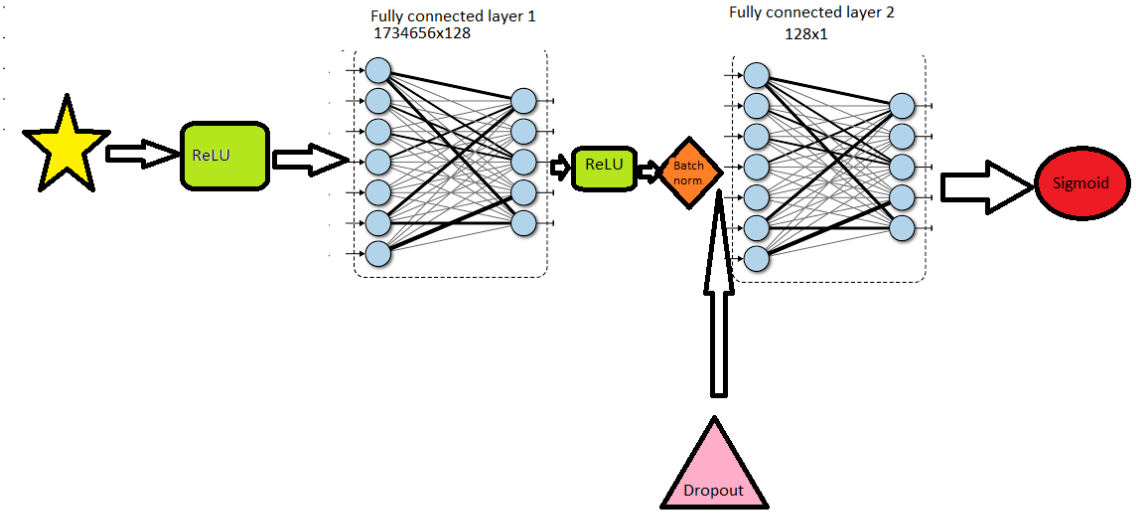
$$117 \times 141 \times 117 \times 50$$

using the formula $O_i = (((W_i - K + 2P)/S) + 1)$ where W is the input-size, K is the kernel size, P is the padding i is the dimension and S is the strides.
Applying the formula again but with stride two gives:

$$57 \times 69 \times 57 \times 50$$

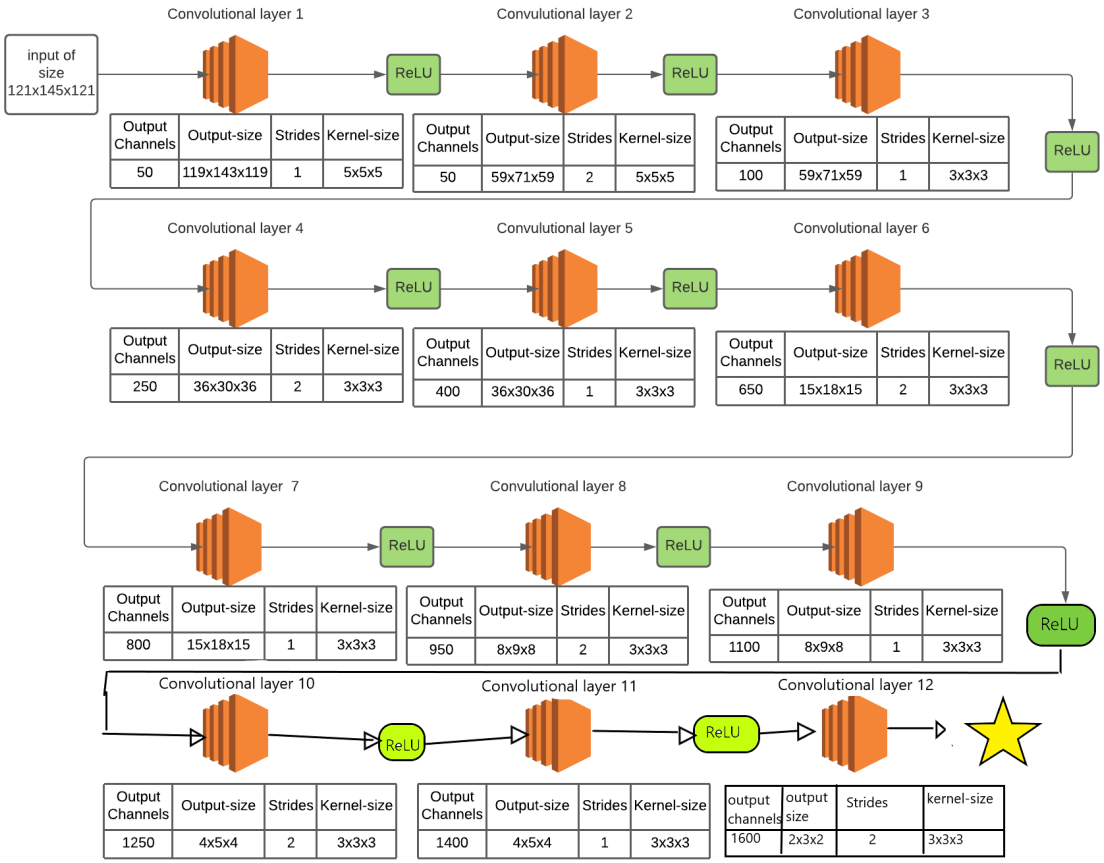


(a) The beginning of the network. The star marks the transition from the figure above to the figure below.

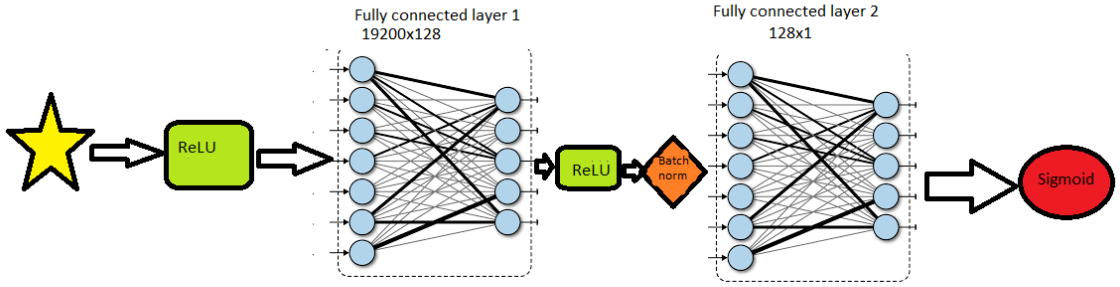


(b) The rest of the architecture.

Figure 10: The architecture for the Six-layer model



(a) The beginning of the network. The star marks the transition from the figure above to the figure below.



(b) The rest of the architecture.

Figure 11: Overview of the architecture of the model from Basaia.

It gets a bit unclear what exactly happens in Basaia's paper from this point, as there are no details on the exact architecture of the individual layers. However, it is interpreted as gradually increasing the number of channels up to 1600 through 10 layers with alternating strides 1 and 2. Increasing the number of kernels by 150 for each layer seems reasonable. So for the next layer, the filter size is $3 \times 3 \times 3$, there are 100 channels, and a stride of 1, resulting in an output of size:

$$55 \times 67 \times 55 \times 100$$

For the next layer there are 2 strides, a filter size of $3 \times 3 \times 3$, and 250 channels. So the output-size is:

$$27 \times 33 \times 27 \times 250$$

It is continued with a layer with filter-size $3 \times 3 \times 3$, stride 1 and 400 channels.

$$25 \times 31 \times 25 \times 400$$

Then continuing with a layer with filter-size $3 \times 3 \times 3$, stride 2 and 550 channels.

$$12 \times 15 \times 12 \times 550$$

And with a layer with filter-size $3 \times 3 \times 3$, stride 1 and 700 channels.

$$10 \times 13 \times 10 \times 700$$

And with a layer with filter-size $3 \times 3 \times 3$, stride 2 and 850 channels.

$$4 \times 6 \times 4 \times 850$$

And with a layer with filter-size $3 \times 3 \times 3$, stride 1 and 1000 channels.

$$2 \times 4 \times 2 \times 1000$$

Now the output is smaller than the filter, and no more layers can be added.
From these calculations, it must be assumed that Basaia used zero padding. With zero padding, the output size of the first layer will be:

$$119 \times 143 \times 119 \times 50$$

The formula from before is applied again. So for the next layer, the output has size:

$$59 \times 71 \times 59 \times 50$$

And continuing with the same approach as before:

$$59 \times 71 \times 59 \times 100$$

$$30 \times 36 \times 30 \times 250$$

$$30 \times 36 \times 30 \times 400$$

$$15 \times 18 \times 15 \times 650$$

$$15 \times 18 \times 15 \times 800$$

$$8 \times 9 \times 8 \times 950$$

$$8 \times 9 \times 8 \times 1100$$

$$4 \times 5 \times 4 \times 1250$$

$$4 \times 5 \times 4 \times 1400$$

$$2 \times 3 \times 2 \times 1600$$

When reaching the 12'th layer, the output size becomes smaller than the filter, accurately making this network possible to implement. There are a few good reasons for zero padding. It is easier to calculate the output dimensions from each layer, as showed above. Another is the possibility to make the network deeper, also shown above, and the last one is that padding keeps the information at the borders. This might not be so relevant for this problem as most borders are just assumed to be black.
The problem with zero padding is that the background and borders of the images are not that interesting for this problem. Zero padding makes these black background pixels have a larger influence on the final result, hypothetically making the network more unstable.
After all the convolutional operations, the network was finalized with a 19200×128 fully connected layer, batch normalization, and a 128×1 output layer activated by a sigmoid function. See an overview of all the layers in figure 11. The choice of the loss function, optimizer, learning rate, and the number of epochs is justified in the discussion, as those were also factors that were not mentioned in Basaia's paper.

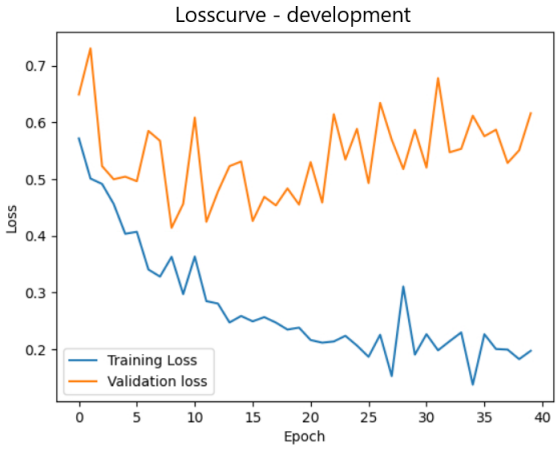


Figure 12: Plot showing the training-loss and validation-loss for each epoch during the development of the network. Since the training loss is decreasing and the validation loss is decreasing and then increasing, it is indicated that this model is overfitting.

A second network implemented for this project

The primary time on this project was spent on developing the neural network and training it. A straightforward network was implemented to begin with, to investigate which kinds of architecture improved performance, and get familiar with the data. The initial network was not inspired by Basaia, since Basaia's model was to complicated to start up with in the debugging-phase. The simple network was implemented using a tutorial [33]. It consisted of two sequences of convolutional layers with 32 and 64 output channels, respectively, followed by a ReLU activation function and 2×2 max-pool layers. Finally, a 258944×128 fully connected layer, followed by batch normalization, a dropout (0.15), and a final 128×2 fully connected layer, was added.

There were some concerns with Basaia's model which are elaborated in the discussion underlying the decision to include, evaluate and discuss this model as well.

To make sure that the network was learning correctly, a loss plot was created like the one in figure 12. The goal for any neural network is that the training loss should go down for each epoch as the network learns to recognize the training set. For as long as possible, the validation loss should follow, but at some point, it will start to go up again, indicating that the model is overfitting. An overfit model will perform very well on training data but not on unseen data, and the goal is to find a balance using, for example, early stopping.

This model was edited layer by layer to improve performance, and in the process, both dropout layers and batch normalization layers were also tested. The main inspiration was still Basaia's

network, so the aim was to shape it more and more into the same architecture. The number of channels and layers are different from Basaia's network, but otherwise, their principles are the same. Another difference is that this network does not implement zero padding.
The first layer is the development model is a CNN with 32 channels, a stride of 1, kernel size of 3, and therefore an output of size:

$$119 \times 143 \times 119 \times 32$$

The next layer is a CNN with 64 channels, a stride of 2, kernel size of 3, and therefore an output of size:

$$59 \times 71 \times 59 \times 64$$

The next layer is a CNN with 128 channels, a stride of 1, kernel size of 3, and therefore an output of size:

$$57 \times 69 \times 57 \times 128$$

The next layer is a CNN with 256 channels, a stride of 2, kernel size of 3, and therefore an output of size:

$$28 \times 34 \times 28 \times 256$$

The next layer is a CNN with 512 channels, a stride of 1, kernel size of 3, and therefore an output of size:

$$24 \times 30 \times 24 \times 512$$

The next layer is a CNN with 1024 channels, a stride of 2, kernel size of 3, and therefore an output of size:

$$11 \times 14 \times 11 \times 1024$$

Finally, it had a 1734656×128 fully connected layer, batch normalization, some dropout, and then a 128×1 output layer activated by a sigmoid function. See an overview in figure 10.
See pseudocode for both models in appendix 7

3.8 Data augmentation

When the model was developed and trained, the running-loss was measured for each epoch and plotted into figure 12. It was clear that the model was overfitting, so data-augmentation was added. For each data item fed to the network, there was a probability of 0.5 of editing the image. If not edited, the original image would just be returned for training. If edited, there

was a probability of 0,33 that it would either be rotated, flipped, or deformed with random values. If rotated, it could be rotated randomly from 0 to 30 degrees. If deformed, the control points were set between 7 and 11, and the maximum displacement between 7 and 16. The numbers were chosen to ensure the image were still a realistic representation of the kind of data the model could be presented to in the future. Making each augmented image depending on changing random numbers, produces a more extensive and different data set for each epoch, explaining why the loss curves are still bouncing for the new model. However, it reduced overfitting tremendously to augment the data.

3.9 Splits of the data

For training and evaluating the model for this project, the data set was split into three different parts - training, validation, and testing. The idea here is to use the validation part for evaluating the model through the development and finetune parameters and layers. The test-set is saved for last, when the network is finished, to make a final conclusion on performance based on completely unseen data, which has not influenced the development of the model. This ensures that the results are representable to the model's performance when used in the future on new data.
Data augmentation was not added to either the test set or the validation set.

4 Results

The models were evaluated by doing 6-fold cross-validation of the dataset and exclude 15 % for final testing. The final test set was not involved in the CV.
The model was trained with the six different sets for 125 epochs, and six different accuracies were acquired on the different splits such that an average could be found. For the CV, the validation and training set was balanced.
The CV for the simple Six-layer model yielded an accuracy of 81,62 %. The worst performance was 76.6 %, and the best performance was 89.8%. See table 6.
The CV for Basaia's model yielded an accuracy of 83,3 %. The worst round yielded 76.9 %, and the best round had an accuracy of 92.3 % See table 7.

	Accuracy	Specificity	Sensitivity
Round 1	84.6 %	80.0 %	92.9 %
Round 2	76.9 %	100.0 %	67.9 %
Round 3	76.9 %	85.7 %	72.0 %
Round 4	79.5 %	75.0 %	84.2 %
Round 5	82.0 %	93.8 %	73.9 %
Round 6	89.8 %	88 %	92.9 %
Average	81.62 %	87.1 %	80.6 %

Table 6: Results of the cross-validation of the simple model with six convolutional layers

	Accuracy	Specificity	Sensitivity
Round 1	79.5 %	76.0 %	85.7 %
Round 2	76.9 %	67.9 %	100%
Round 3	82.1 %	85.7 %	80.0 %
Round 4	89.7 %	94.7 %	85.0 %
Round 5	79.5 %	78.3 %	81.3 %
Round 6	92.3 %	96.0 %	85.7 %
Average	83.3 %	83.1 %	86.3 %

Table 7: Results of the cross-validation of the model from Basaia et al. [1].

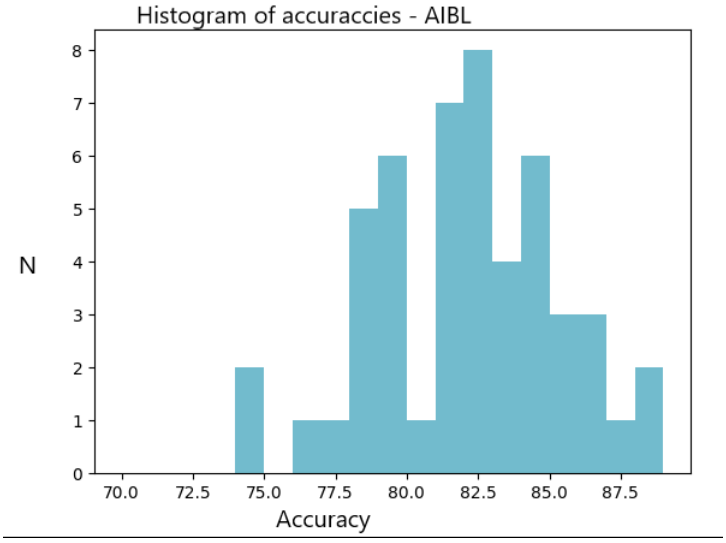


Figure 13: Plot showing the histogram over the accuracies achieved from testing 50 permutations of data from AIBL on the Six-layer model. The data was equally distributed for all permutations.

4.1 Testing the model only trained on 3T images

The accuracy for the test set from ADNI was 83,83 for the Six-layer model and 80.8 % for Basaia's model. In both cases, the last model from the CV was used.
To test AIBL, 50 permutations of data were initialized with the same distribution of CN and AD patients as there was in the test set for ADNI. All 50 permutations of data were tested on the same model and rounded to one decimal, and the output was the histogram in figure 13. The average accuracy, sensitivity, and specificity of these runs are given in table 8.

The average accuracy for testing AIBL 50 times with 50 different samples on the same model (the simple Six-layer model) was 81,65 %. See the details in table 8 All of these test sets had a balance: 74 CN and 23 AD patients for measuring accuracy to ensure that the accuracy was not biased. However, the balance for measuring specificity/sensitivity is given in the table as all images not involved in training or validation could be used to measure those.

Overall for these two models, they perform very well on ADNI and AIBL. They both achieve accuracies above 80 %.

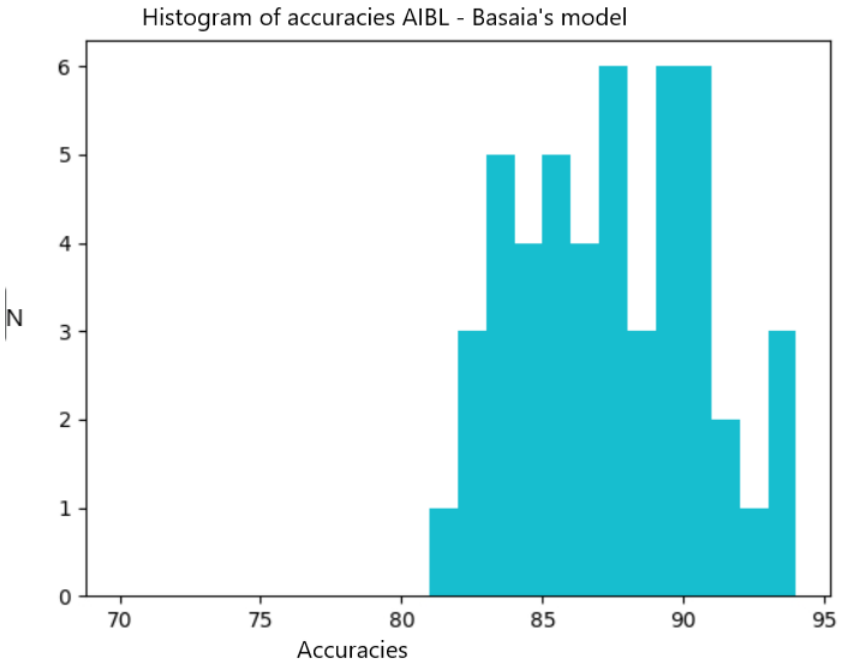


Figure 14: Plot showing the histogram over the accuracies achieved from testing 50 permutations of data from AIBL on Basaias model. The data was equally distributed for all permutations.

	Accuracy	Specificity	Sensitivity	CN/AD	TP	TN	FP	FN
ADNI	83.8 %	89.1 %	65.2 %	74/23	15	66	8	8
AIBL	81.26 %	81.4 %	80.1 %	392/67	54	319	73	13
Italian ADNI	68.4 %	N/A %	68.4 %	0/73	50	0	0	23

Table 8: Result for the final testing of the Six-layer model- All images were 3T. The accuracy from AIBL is an average of the 50 test runs with different permutations of 23 AD and 74 AD patients such that the accuracies are comparable. The specificity and sensitivity is measured on the part of the AIBL dataset that was 3T.

	Accuracy	Specificity	Sensitivity	CN/	TP	TN	FP	FN
ADNI	81.8 %	87.8 %	65.2 %	74/23	15	65	9	8
AIBL	87.7 %	91.1 %	73.3 %	392/67	49	357	35	18
Italian ADNI	57.5 %	N/A %	57.5 %	0/73	42	0	0	31

Table 9: Result for the final testing of the model from Basaia et al. - All images were 3T. The accuracy from AIBL is the average of the 50 test runs with different permutations of 23 AD and 74 AD patients such that the accuracy compares. The specificity and sensitivity is measured on the entire AIBL dataset that was 3T.

Group	Sex	Dataset	Age	Sex	Dataset
AD	F	ADNI1	73.893617	47	47
		ADNI2	74.255814	43	43
	M	ADNI1	75.905660	53	53
		ADNI2	76.105263	57	57
CN	F	ADNI1	73.098039	51	51
		ADNI2	72.072727	55	55
	M	ADNI1	73.265306	49	49
		ADNI2	77.000000	45	45

Figure 15: Demographics for the training set of the mixed model.

4.2 Training model on both 3T and 1.5T images

The Six-layer model was also trained using the entire dataset of ADNI1 and ADNI2 (except for the SAG_IR-SPGR scans). Then it was tested on different datasets, both consisting of 1.5T and 3T images, and datasets that only had one kind. This was to investigate if there was a bias between 1.5T and 3T and see if performance could be improved if significantly more data was included. The training set had 200 1.5 Tesla images and 200 3 Tesla images and consisted of 200 patients with AD and 200 healthy controls. The demographics of the training data can be seen in figure 15.

Looking at the results in table 10 it is observed that, in general, the performance goes up even when the test-set is only one kind of field strength, and the general ability to diagnose patients correctly for this model is over 80 %. The models seem to perform best on ADNI, a little worse on AIBL, who, however, has better sensitivity than ADNI, and even worse on I-ADNI, whose metrics are all low. The ability to diagnose a patient as healthy (specificity) is higher for 3T images, whereas the ability to diagnose an AD patient is higher on the 1.5T images. It could be indicated that the models tend to overestimate 1.5T images and underestimate 3T images. For the Italian ADNI, the sensitivity is generally lower, and the specificity was impossible to measure because of the lack of healthy controls in this site.

The same thing was done for Basaia's model, see figure 11, and many common patterns were observed. However, Basaia's model seems to have a tough time predicting AD patients since every sensitivity is very low, and thus for I-ADNI, the performance corresponds to random classification. Furthermore, Basaia's model seems to have significantly better performance for 1.5T images. In general, the results for Basaia's model are more unstable as higher differences are observed both between sites, field strengths, and specificity versus sensitivity.

	Accuracy	Specificity	Sensitivity	N 1.5T/3T	N AD/CN	TP	TN	FP	FN
Mixed testset ADNI	83.8 %	84.1 %	83.5 %	357/140	170/327	142	275	52	28
AIBL (mixed)	79.2 %	77.0 %	92.4 %	0/558	79/479	73	369	110	6
AIBL (only 3T)	77.3 %	75.0 %	91.0 %	0/459	67/392	61	294	98	6
AIBL (only 1.5T)	87.9 %	86.2 %	100 %	99/0	12/87	12	75	12	0
ADNI1 (Only 1.5T)	82.4 %	82.1 %	83.0 %	357/0	106/251	88	206	45	18
ADNI1 (Only 3T)	87.6 %	90.8 %	84.3 %	0/140	64/76	54	69	7	10
Italian ADNI mixed	78.7 %	N/A	78.8 %	108/73	179/2	141	1	1	38
Italian ADNI (3T)	73.3%	N/A	73.3 %	0/73	73/0	53	0	0	20
Italian ADNI (1.5T)	82.4 %	N/A	83.0 %	108/0	106/2	88	1	1	18

Table 10: Result for the final testing - The 6 layer model trained on both 1.5 T and 3T images

	Accuracy	Specificity	Sensitivity	N 1.5T/3T	N AD/CN	TP	TN	FP	FN
Mixed testset ADNI	84.6 %	93.3 %	67.6 %	357/140	170/327	115	305	22	55
ADNI1 (Only 3T)	78.0%	93.4 %	59.4 %	0/140	64/76	38	71	5	26
ADNI1 (Only 1.5T)	87.1 %	93.2 %	72.6 %	357/0	106/251	77	234	17	29
AIBL (mixed)	91.9 %	96.2 %	65.8 %	0/558	79/479	52	461	18	27
AIBL (only 3T)	91.72 %	96.2 %	65.7 %	0/459	67/392	44	377	15	23
AIBL (only 1.5T)	92.9 %	96.6 %	66.7 %	99/0	12/87	8	84	3	4
Italian ADNI mixed	57.9 %	N/A	58.1 %	108/73	179/2	104	2	0	75
Italian ADNI (3T)	44.0 %	N/A	44.0 %	0/73	73/0	31	0	0	42
Italian ADNI (1.5T)	69.4 %	N/A	68.9 %	108/0	106/2	73	2	0	33

Table 11: Result for the final testing - Basaia's trained on both 1.5 T and 3T images

4.3 Investigating gender bias

It was also investigated whether there was a difference in performance between genders for AIBL and I-ADNI. For this case, the model was trained on all AD or CN patients in ADNI apart from a small validation set. Therefore there is no test set for ADNI.

	Accaracy	Specificity	Sensitivity	CN/	TP	TN	FP	FN
AIBL - Females	94.39	98.2 %	71.8 %	275/46	33	270	5	13
AIBL - Males	91.6 %	96.1 %	63.6 %	203/33	21	195	8	12
AIBL - All	93.2 %	97.3	68.4	479/79	54	466	13	25
I-ADNI - Females	57.9 %	N/A	57.9 %	0/114	66	0	0	48
I-ADNI - Males	62.3 %	N/A	63.1 %	2/65	41	2	0	24
I-ADNI - All	60.7 %	N/A	59.8 %	2/179	107	2	0	72

Table 12: Test-results regarding potential gender-bias for the Six-layer model model trained on all the data in ADNI.

This model revealed no significant differences between the genders overall. However, it is observed that the performance is slightly better for the females in AIBL, and for I-ADNI, the better performance is for the males. It is noted that the general difference in performance for AIBL and I-ADNI becomes even more evident.

5 Discussion

5.1 Data selection

One of the first significant decisions for this project was how to choose the data. Only 3T baseline images was used initially since this would ensure consistency in the data as all images would have similar quality.
The point of training the Six-layer model on both 1.5T and 3T images afterward was to observe if a bias was evident when only evaluating on either 1.5 T or 3.0 T images and to investigate the general performance when adding more training data. The hypothesis was that the difference in signal-to-noise ratio caused by the different magnetic field strength would cause the network perform better on the 3T images with the high signal-to-noise ratio. Only MPRAGE images were used for this project as it was a concern that different modalities in the scanners might affect the network in an unforeseen way. This was why all ADNI2 subjects that only had SAG_IR-SPGR scans were excluded.

5.2 Challenges from the datasets

Imbalanced data

Since the main topic of this project was to test for demographic bias, it proved quite the challenge that both AIBL and I-ADNI were so imbalanced. Even more so because they were imbalanced in different ways - Where AIBL had many controls (almost 84 %), I-ADNI had only two healthy controls in total. This fact made the accuracy impossible to compare, and therefore the sensitivity for I-ADNI was the only reasonable performance measure. However, for AIBL, the sensitivity was based on very few samples (79/479), and therefore, it is hard to ensure the robustness of the performance.
The amount of data for training was also challenging, especially for the models only trained and tested on 3T images. After the exclusion described in section 3.1, only 335 3T (142 AD, 193 CN) images were left for training, validation, and testing. Furthermore, the training set was chosen to be balanced to eliminate potential bias, and some AD patients should be in all three splits to test appropriately.
The data augmentation should remedy this to some extent for the training set. However, the imbalance of the test sets that the balancing of the training set creates for 3T ADNI images is still an issue. The low data sample caused a worrying compromise between the robustness of the measured performance and the amount of training data.

Magnetic Field strength

As argued earlier, it is essential for training and testing that the magnetic field strength is the same for all the images. Throughout this entire project, it has been quite challenging to decipher the field strength of a given image. They were not present in the essential .csv files, holding diagnosis and demographics.
For ADNI2, all images were 3T [34]. A part of the ADNI1 collection was named "baseline3T" on the ADNI website, so all images in that collection are assumed to be 3T as well, even though none of the images could be looked up using the advanced search on the ADNI webpage to ensure this. Their image IDs had no matches. The remaining images from ADNI1 are assumed to be 1.5T, and many attempts were made to ensure this. The ADNI webpage was searched for a datasheet containing this information which should be possible to find using the advanced search function. However, for ADNI1, the advanced search does not seem to work correctly, as an image cannot be found on its ID; there were simply no matches for many of the images. The only kind of information that could be found was the scheme shown in 21 which is not specific for each image but suggests that some images in ADNI1 are 3T. This should be noted when evaluating the results.
However, for AIBL, this information was found. Using the advanced search on the ADNI website for all baseline 3D images from AIBL, it was possible to view the magnetic field strength and Image ID for each Image. That could be downloaded as a .csv. Merging that .csv, with the dataset from AIBL on Image IDs and Subject, gave a complete overview of the magnetic field strength of AIBL.
For I-ADNI, the task was finally straightforward as the field strength was given in the filename for each subject.

Cleaning up and finding information

Much time for this project has been used to clean up and rearrange the non-image data. Demographics and group size from the ADNI papers [3] [4] [5] did not match up to what was present in our downloaded set, and finding information on why this was the case, was not possible. Finding a diagnosis for each image also proved challenging as the csv files that followed the downloaded images from, for example, AIBL, did not hold this information. The webpage

for AIBL had to be investigated to find this. Furthermore, it was of uttermost importance that the image ID for an image was present with the diagnosis in the .csv-files as this was the only factor that could ensure that the diagnosis and image matched up.
I-ADNI was the most organized dataset. Every needed information was present in either the filename for the image or the .csv file. However, the different structures of the image files had to be adapted to ADNI and AIBL for ease of implementation.
The .csv files for each site were extended with columns containing the path to the preprocessed image, the magnetic field strength, and the site. This was to ensure consistency and ease of implementation.

5.3 Implementation choices

A very crucial part of this entire project was to implement the neural network. During the implementation, many choices had to be made regarding design and architecture. In the following subsections, the essential choices are justified.

The design of the network

As stated in section 3.7, the basics of the network are entirely based on the paper by Basaia et al. [1] because the study showed that this architecture had a very high performance on this specific problem. So layer by layer Basaia's model was implemented, constantly testing that the current model was learning satisfyingly by checking that the loss consistently decreased for each epoch. Zero padding had to be added at some point to create more layers, as discussed in 3.7.
Getting started on the network was difficult. The architecture from Basaia's model is very complicated and challenging to implement in one go. Memory limitations were also quickly reached, and a data loader from PyTorch had to be utilized. Therefore a choice was made to implement two networks. Basaia's model was put away, and the tutorial from [33] was implemented instead. It has a straightforward structure that was learning from the beginning, even though the performance was not very good. Much time was used to improve this model and turn it gradually into Basaia's while maintaining acceptable performance; however, it was discovered by a lot of trial and error that the model did not seem to benefit from having 12 layers. Adding layers sustained learning until a given point (6 layers), and then it did not improve performance anymore. Contrarily the results became more unstable, and the

computation time was increased.
It can be hard to predict the final performance precisely in the development phase, as many parameters can be tweaked outside the layers, which might influence performance. Therefore, the choice was made to continue with Basaia's model even though it did not improve performance and keep the development model. It was kept in mind that the architecture of Basaia's network was never clear and that some vital parameter tuning might be missing.
During development, the biggest challenge was the run time of training and the availability of the GPU on the cluster. Each model took a steady amount of time to train and test, and for each minor tweak of a parameter, training had to be repeated to make sure what tweak actually impacted performance. The GPU queue was often occupied, and sometimes it could take over 10 hours before a job was able to run and then another 10 hours to train the network. Eventually, it was decided that the performance was acceptable and that sufficient time had been spent experimenting. It is entirely possible that more tuning of the network could have made it perform even better. Finally, a more explicit description from Basaia et al. of the network might have given more accurate results and saved time on testing architectures. It was even tried to ask Basaia et al. for clarification; however, no answers were received.
Beneficially, much was learned from this process about this problem.
Most of the tuning in development was done based on experience and trial and error, but some of it was chosen based on different arguments presented in the below sections.

Max-pooling versus increasing strides

As discussed earlier, the purpose of a convolutional layer in a network is to summarize the features of the image. However, these summaries can be very sensitive to the location of the features in the input, and there can be a desire to make this more robust since this will have a significant effect if an image is shifted or rotated just a fraction. One can achieve this by downsampling the image and keeping essential signals using pooling. For convolutional networks, max pooling is used as the interest lies in the crucial features. This is a very cheap operation that's easy to implement.
An essential part of the architecture in Basaia's network and the one built for this project is that they do not contain max-pooling layers. The usual way to construct a network is to include a series of sets consisting of a convolutional layer, an activation layer, and a pooling layer to downsample. However, for this project, the sets only contain the convolutional layers with alternating strides. The fact that the strides of each convolutional layer alternate between 1 and 2, ultimately also downsamples the image. Research has shown that constructing a network

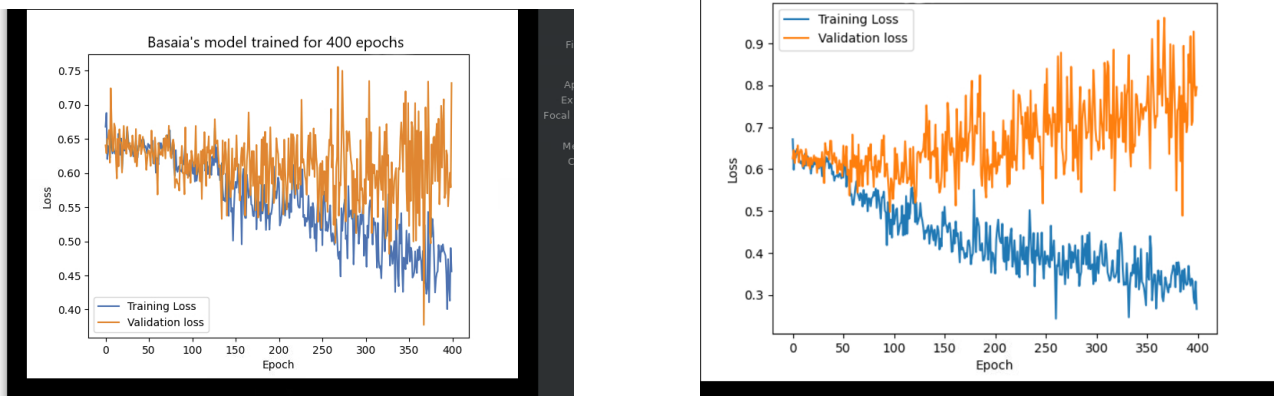
like this ending with a few fully connected layers can actually maintain or even improve the final accuracy on well-known deep learning architectures like CIFAR-10 and ImageNet [35]. In general, it should be noted that convolutional layers has learnable parameters, and therefore are more computationally expensive. Furthermore, the architecture is simplified as different types of layers is not needed. However, max-pooling is a cheap and straightforward operation that makes the network faster.
The idea to use alternating strides originated from Basia's paper [1].

Number of epochs

In order to determine the number of epochs for both models, early stopping was used. The data-set was split into a training and validation set (only 3T images from ADNI were used for this part), and the model was trained for 400 epochs at first. Then a plot of the training-loss and validation-loss was used to determine when to stop. As long as both losses are decreasing, it will make sense to continue training, but at some point, the model will overfit the training data and inherit the noise and potential imbalance of the training set. This will result in a lower ability to generalize, and therefore performance will decrease on unseen data increasing the validation loss. This is the point to stop training in simple cases. However, in practice, the training of a neural network is stochastic and noisy. The performance of the validation set can be bouncy between epochs, which was also the case for this project. Therefore the point for stopping was eyeballed by looking at the general tendency of the curves. Based on this, 125 epochs were chosen for the Six-layer model and 200 epochs for Basaia's model. See figure 16.

Noisy loss curves

Due to the noise of the loss curves, pinpointing the exact fitting number of epochs was challenging. A reason for the noise could very well be the small batch size of 3 and the fact that the training data changes for each epoch due to the augmentation. However, since each data point was a 3D image containing $121 * 145 * 121 = 2.122.945$ pixels, memory restrictions did now allow for a bigger and more generalizing batch size. Another option could also be that the initial learning rate is still too high even though it is adaptive. ADAM might still need too many epochs to find a reasonable learning rate in each direction. However, considering that training for 200 epochs is already time-consuming and that lowering the learning rate would mean training for even more epochs, the value of the learning rate was maintained.



(a) The loss curves for Basaias model

(b) The loss curves for the 6 layer model

Figure 16: Plot showing the training loss and validation loss for each epoch during the development of the networks. It is observed that both loss curves are noisy, but it is possible to spot a general tendency, and at around 125 and 250 epochs, the validation loss seizes to decrease, so this is where to stop. The lowest validation loss is roughly the same for each model.

Batch normalization

The two final layers of both models were fully connected layers separated by a batch normalization layer. The experience proved that not doing the batch normalization caused the model to classify everything as the same class regardless of training time. This would suggest that some neurons have inappropriately large weights, causing them to eliminate smaller but important signals. This is a pretty common problem [36].

Learning-rate and optimizer

Adam was chosen as the optimizer because it has both an adaptive learning rate and momentum, as discussed in the background section. It is the most used optimizer for neural networks. However, recent optimization algorithms have proven to be better and solved some of the pitfalls of ADAM using other adaptive momentum approaches. However, their implementation into the machine learning world is going slow, and therefore, they are not used. They could be considered implemented for this network in the future [37].
The learning rate was based purely on previous experience with convolutional neural networks. Even though ADAM can adapt it during training, the initial choice is still relevant. Initializing far away from the optimal learning rate(s) would increase the time to find the optimal rate for each data point, and there are many to consider. Higher and lower initial learning rates were

tested, but based on the behavior of the loss functions, a learning rate of 0.0001 was chosen.

5.4 Metrics for evaluation of results

It proved to be a challenge to measure the performance of the models accurately, such that an eventual bias would be clear to spot. Looking at the results for the cross-validation (see 6 7) and the loss curves in, for example, figure 16 it is observed that the performance for each epoch differs a lot. Testing models trained for the same amount of epochs on the same test-set revealed very unstable accuracies ranging from 75% to 90%. This is a consequence of the noisy loss curve that is explained in section 5.3. In an attempt to reduce this instability and ensure that good accuracies were not just pure luck for the choice of test-set, cross-validation was done. This was also why 50 different permutations of data from AIBL were tested on the final model. It made the results a lot more generalizable. Doing 50 permutations of I-ADNI with a comparing distribution of AD and CN patients was not possible due to the lack of CN patients. The way cross-validation was used to evaluate the network is a little different from the way cross-validation is typically used. Conventionally it is used to tune a parameter. However, one is limited when working with such a large dataset and a deep and complicated network. Many epochs are needed to gain the desired accuracy, and even doing one epoch is computationally expensive and takes much time. Many of the last tested networks close to the final ones took over 10 hours to train. K-fold cross-validation requires k training runs which would then take days for each parameter test, so this was deselected. However, when the model was finished, and it was decided that the performance was acceptable based on the validation set, the model was trained for a fixed number of epochs six times to ensure the final accuracies were representable to future test-runs.

5.5 The two different networks

In the following, it is finally described what was learned from the results of the two networks.

Evaluating the model from Basaia et al.

In the paper written by Basaia et al., they report an accuracy of around 99% for the model described and implemented in their project. As shown in the results reported above, the model implemented for this project cannot match that accuracy. There can be several reasons for this. Firstly, there may be differences in the architectures of the two models since Basaias model is not described in heavy detail in the paper. The channels and strides of each layer are

never described, only that the strides should be alternating between one and two and that the channels should range from 100-1600. The choice of the optimizer, loss function, initial learning rate, and number epochs is not mentioned either. Furthermore, zero padding is only assumed because of the argument presented in the architecture section 3.7. Also, Basaia et al.'s paper never mention the use of batch normalization. However, the experience was that it had to be introduced very early in development in order for the network to learn.
Secondly, it should be noted that the two models that are presented in this project do not differ a lot in general performance, but they do differ a lot in complexity and robustness. Basaia's model has so many layers that it can only be implemented with zero-padding, suggesting that their signal will be heavily dominated by 0. This means that the last layers will not contribute with any new information. This is also what the results suggest since a much simpler network with fewer layers can match the performance. Challenging Basaia's network with different sites causes heavier problems than the Six-layer model show. Thirdly it must, with caution, be stated that accuracy of 99% must be too good to be true. Alzheimer's is a clinically diagnosed disease; therefore, the MRI scan will never be sufficient to make a diagnosis. An accuracy this high suggests that this is possible. However, without clinical symptoms as features, there should be several false positives and false negatives among the patients in the test-set. There will be cases where there are only clinical symptoms and cases where the MRI suggests Alzheimer's, but there are no clinical symptoms present. Fourth, it is also noted that Basaia had access to a lot more 3T images for training and testing, which could also improve their performance.

Evaluating the simple 6 layer model.

Since the performance of the Six-layer model matched the performance of Basaia's model and showed significantly more robustness, it was the primary model used for the different experiments described in this project. The experience with Basaia's model was that even minor tweaks or imbalances could have significant impacts on performance. Therefore it did not contribute to the gender experiment. A reason for the instability could again be the complexity of the model - usually, adding more layers will increase performance as it will allow for more features to be picked up, but only up to a certain point. After that, it will tend to overfit the data since it might pick up on features that are not relevant for the given problem. Also, for obvious reasons, the simple model was a lot faster to train several times and made more time for experimenting with the training sets and architecture.

5.6 Overall performance

In general, the pattern for both models and the different ways they were trained was that they had more challenges diagnosing sick patients than classifying healthy ones (low sensitivity, high specificity). This problem was especially significant in Basaia's model and almost eliminated in the Six-layer model trained on both 1.5T and 3T. Due to the imbalance of data and especially the fact that I-ADNI had almost no healthy controls, the best measure for the bias is not the accuracy, but the sensitivity, which measures the model's ability to predict if someone has AD. It is noted that the sensitivity of the Six-layer model trained on 3T images is low both for ADNI and I-ADNI. However, the lack of AD patients in the ADNI data might have made the result for ADNI a bit unrepresentative. However, since there were only 142 AD patients with 3T images in ADNI, it was a compromise that had to be made in order to ensure that there was also enough training data.
Nevertheless, AIBL has a very high sensitivity compared to I-ADNI, which both have a sufficient amount of AD patients in their dataset to make them comparable. It is cautiously concluded that the Six-layer model has a better performance on AIBL than on I-ADNI.
It is the same case for Basaia's model only trained on 3T images. However, it seems that it overfits a bit more on ADNI in general regarding sensitivity than the Six-layer model since both the sensitivity for AIBL and I-ADNI drops. The accuracy for AIBL is higher than ADNI, which intuitively seems a bit odd, but there could be many reasons for this. There could be an unforeseen bias that makes the data in AIBL easier to predict. This could be many cases where the brain and mental status of the patient correspond, unexpected differences in modality between scanners resulting in different qualities, or a human bias in doctors' evaluation. It is noted that that the difference in ethnicity between AIBL and ADNI does not seem to have a bad influence on performance.
The investigation for ethnic bias becomes a lot more tangible when looking at the Six-layer model trained on a more extensive data set, including 1.5 T images. The data set was balanced both on magnetic field strengths, diagnosis, and genders. It is observed that the performance on I-ADNI is significantly worse, especially on the 3T images, and that the general performance is stable and pretty good. As before, it is not safe to conclude that there is an ethnic bias, but it is safe to state that the results indicate that the model has a hard time on I-ADNI, and one reason could be an ethnic bias.
There seems to be some difference in performance for the two field strengths. In particular, the model from Basaia distinguishes comprehensively in performance, and surprisingly the 1.5T images seem to perform best. In comparison, the predictions for 3T images, especially

I-ADNI, are tremendously wrong, being even below null accuracy. It might be for the same reasons that AIBL performs better than ADNI, and it should also be noted that the test sets are not balanced regarding diagnosis, making the results harder to interpret. It should also be considered that the performance of the Six-layer model for this experiment is, in general, better, especially the sensitivity, and that the difference is not so significant here as it is for Basaias model, just like with the ethnic bias. It is observed that the Six-layer model trained on the same data only inherits the problem vaguely and even has better performance on 3T for ADNI. A part of the reason for that could be that it is not ensured that there are not some 3T images among the set of 1.5T images in ADNI (see secton 5.2). However, this still disturbs the pattern of 1.5T images performing better since it should be ensured the there are no 1.5T images in the 3T part of ADNI.
Nevertheless, this should be investigated further with more data and other models. It could also be tested for models trained on only one kind of field strength using the same principle as for the testing of ethnic bias.
The last experiment was to investigate a potential gender bias. However, this did not yield any significant differences. The experiment does not confirm that no gender bias is present - that would require more data and a more extensive quality check of the training data, which is not evenly balanced between genders. The choice to include all the data was made since this would resemble what is done in practice, as more training data seems to improve and stabilize overall performance. Since the number of males and females is 496 males and 579 females, the imbalance is also not significant. With this experiment, a significant difference in performance between AIBL and I-ADNI, is observed which magnifies the results from the two previous models, suggesting that an ethnic bias is present.

5.7 Improving overall performance

It seems that more training data continues to improve the performance of the models. In general, the best performing model was the one testing for gender bias, which was also trained on the most data. However, one should be careful with this as the training data is not balanced in any way. Conclusively, it is indicated that including more balanced training data might improve performance while maintaining robustness.
Another way to improve performance could be to consider cropping the dark areas around the brains away. This would eliminate the possibility that noise in these areas is picked up as features and center the convolution on the essential structures. Another option could be to add learned data augmentation. Studies [38] show that using pairwise alignments of images

from the same class to make a learned transformation can improve performance on CNN's.
Apart from the design of the networks, which can always be explored even further, there are some factors of AD whose presence might have improved the performance.
The fact that AD cannot be diagnosed solely from a brain scan should be considered. There may be several cases in the training set where these two factors do not add up. It could be a speculation that there might be an overflow of patients, especially in the 3T images with diseased brains, and a low disability score in the training set. This would mean that the network has been exposed to many diseased brains classified as healthy, and therefore will have a higher tolerance on features that indicate illness in the test set. In general, the network will have a hard time with patients whose mental status does not correspond to the level of disease in their brains, and the distribution of these cases is unknown in these datasets. The overall performance of the models could probably be improved by including the clinical symptoms as features and make it a multi-input network.
However, clinical symptoms can be tough to translate onto an AI . Evaluation practices should be very consistent for this to have maximum effect, and given that each doctor will have some amount of human bias, this can be very hard in practice. Even if many tests are standardized, doctors still have to evaluate parts of the process. A suggested way to accomplish some of this is suggested in the following paper [39] where they propose an AI-ecosystem that ensures cooperation within the field. This should ensure that the clinical evaluation, the data collected, and the developed models are presented in a common framework such that everything is kept as consistent as possible all over the field. However, with the rapid development that AI is going through, an idea like this could decay only to be an admirable effort. Such a framework would easily take so long to implement that its methods will be outdated before then.
Being aware that the clinical symptoms might also hold some bias and the risk of a human bias might be more prominent for this area, they should still be considered essential features.

5.8 Preventing unforeseen bias

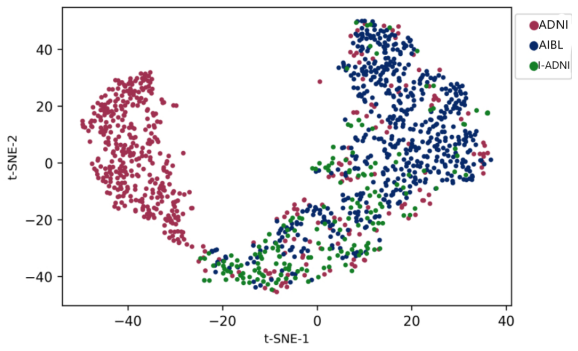
Even though this project's entire scope is wrapped around algorithmic bias and fairness, it is quite possible this it is still biased in an unforeseen way. This is one reason why the conclusion that an ethnic bias is present is made with much caution. There can be a bias in the demographic data that was not investigated (which could cause an aggregation bias in the model), and there might also be an unforeseen bias in the demographics that where not accessible. For this project, and in general, it is often impossible to balance data on every

demographic factor because that would cause too much data to be excluded. Also, even when balancing perfectly on each factor, there might exist problems because one side of a balanced split might have more detailed features than the other. This was another reason why the magnetic field strength of each image was considered essential for this project, and there might be other factors with similar problems.
It could be speculated that *Measurement bias* is also present since all SAG_IR-images were excluded because there was a hypothesis that the difference in modality might affect performance in an unforeseen way. This should have been investigated further to make a safe conclusion.
In each case, numerous considerations were made to prevent unforeseen bias as much as possible. Both ethnicity, gender, and magnetic field strengths were tested for bias, and for the mixed data trials, field strength and gender were also balanced in the training set. AIBL and I-ADNI were kept out of all training to avoid the risk of inducing a bias in ethnicity, as this bias was the whole starting point for this project. In practice, one should be aware of this balance as well. However, ensuring a bias-free training set is challenging in practice. It is not safe to conclude that all patients in the American data-set are, in fact, American natives, and the same goes for AIBL and I-ADNI. Keeping track of the demographics for both training, validation, and test sets has significantly been prioritized during this project, but it is doubtful that everything has been considered. In practice, some of the demographic information might be unavailable or impossible to interpret.
Human bias should also be considered. Even though clinical tests for AD are standardized, there still might be differences in how the doctors diagnose the patients across the world, and some doctors might be more likely to give the diagnosis than others.

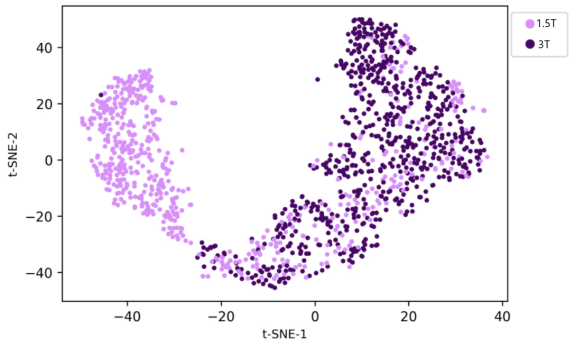
5.9 T-SNE

T-SNE was used to visualize the data after the first fully connected layers activation function, where the output was a vector of 128 features. This was to get an overview of the variances between the different splits of data. There has been much speculation for this project whether the image quality and performance of the network differs between sites, tesla-units, genders, and diagnosis. T-SNE provides an option to examine this visually. The input to T-SNE was the entire concatenated test set consisting of all three sites and both 1.5T and 3T images. The two networks were trained on a balanced set of 1.5T and 3T images.
The results of the T-SNE can be found in figure 17 for the Six-layer model. T-SNE was also run for Basaia's model, see figure 18. Both yielded a lot of the same patterns. However, Basaia's model was a bit more significant in the variances, and it was very clear from the plot of the

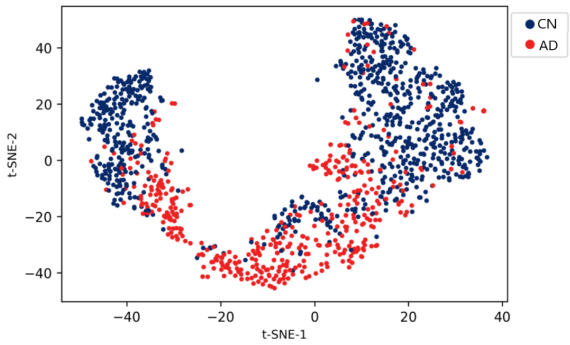
confidences and diagnosis that Basaia's model had significant issues with the AD patients, particularly the ones from I-ADNI as they shared color patterns. This is not the case for the Six-layer model, which has a very different confidence plot.
The highest confidences for the Six-layer model are found in the middle of the "clusters," indicating that the confidence levels do not relate to the splits but have a unique pattern. It seems that images that are locally close to many other images are easier to predict.
Taking point in the plots from the Six-layer model but describing patterns that the models share, it is observed that minimal variance is found between the genders, which would also be expected after investigating gender bias. However, much variance is found in all other splits. It is observed that the two "sides" of the figure mainly represent variance between the sites and between 1.5T and 3T. The entire left side is 1.5T images from ADNI, whereas the top and bottom seem to mark the diagnosis groups' variance. It should be noted, however, that almost no CN patients are present in I-ADNI, which could camouflage the tendency between AIBL and I-ADNI as variance in sites. The variance might only occur because there are an overflow of CN in AIBL and vise versa in I-ADNI. Conclusively, the T-SNE analysis backs up the results suggesting variances in image quality between all splits besides the gender. The variances in performance are more distinguishable for Basaia's model, which was also what the results showed.



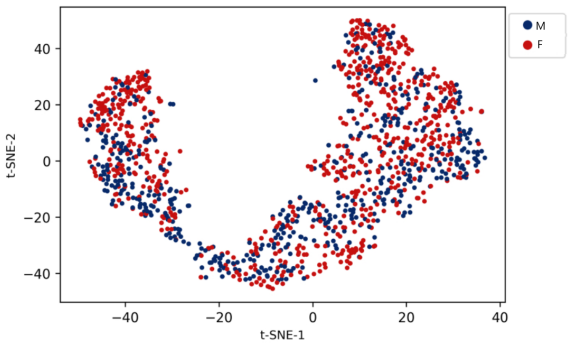
(a) T-SNE colored by dataset



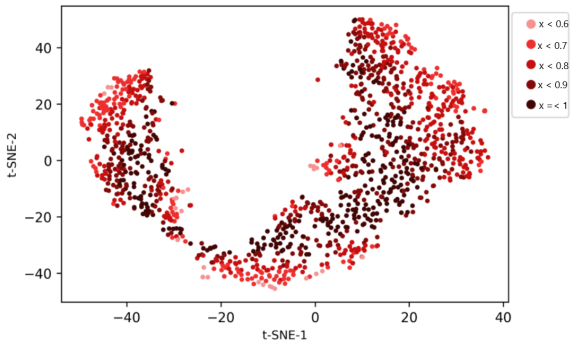
(b) T-SNE colored by Field Strenghts



(c) T-SNE colored by diagnosis

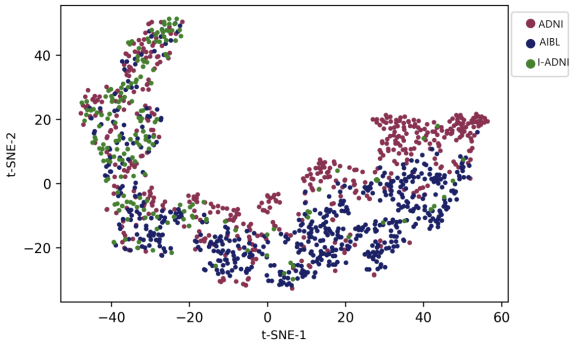


(d) T-SNE colored by gender

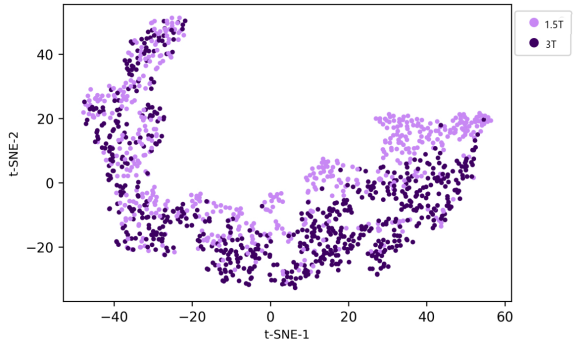


(e) T-SNE colored by confidence

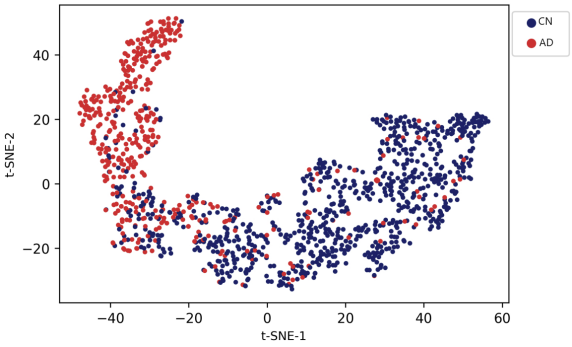
Figure 17: T-SNE plots for the entire collections of tests sets using the output from the first fully connected layer from the Six-layer model as features.



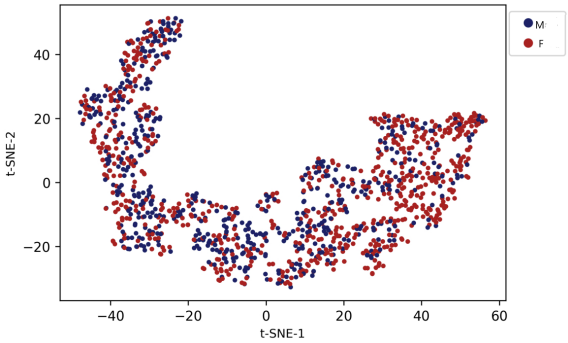
(a) T-SNE colored by dataset



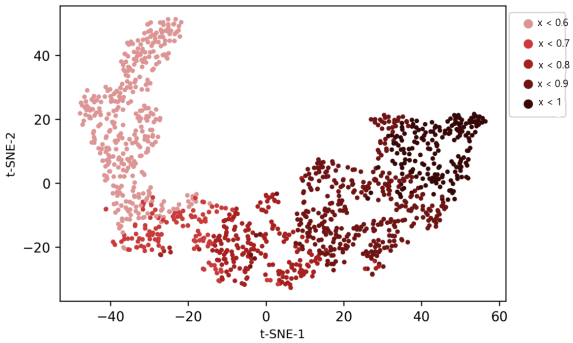
(b) T-SNE colored by Field Strenghts



(c) T-SNE colored by diagnosis



(d) T-SNE colored by gender



(e) T-SNE colored by confidence

Figure 18: T-sne plots for the entire collections of test sets using the output from the first fully connected layer from Basaia's model as features.

5.10 A discussion on bias and fairness

Before wrapping this project up, the concept of a bias in algorithms for diagnosing patients should be discussed. In the background section, the term "bias in algorithms" is defined, but where does this bias originate, and how could it be remedied?

This project is showing concerning effects of a demographic bias, and other concerning bias-

problems are countless. Studies have shown that diagnosing idiopathic pulmonary fibrosis has a gender bias [40], that skin lesions can be harder to detect on black skin [41] or that the impact Alzheimers can have on patients can be affected by their IQ [12]. The common factor for this research is that the bias originates already in (maybe unknown) factors from the subjects' psychical and mental state, which could raise a very critical ethical problem when selecting patients for a trial. Suppose the desire was to create a data set with no bias, then *all* bias-potential factors should be recorded for each patient. Their IQ should be recorded, their skin color should be examined, and their amount of breast tissue should be determined. Given that the before mentioned examples are far from the only ones, the sky is the limit on what could bias the performance of diagnostic tools, and each factor that requires a doctor's examination could induce a human bias. For this project, a similar problem was observed since the patients' clinical symptoms were not taken into account. They could hold an unknown bias if the data set contains many patients with high IQ or many patients whose mental status does not correspond with their MRI. Especially the IQ might not even be possible to record if the patient is already showing symptoms.

Even if data is available, it can still be hard to find and understand. As it was seen with the difficulty of finding the relevant information on field strengths for each image, it can also be challenging to retrieve the information even if it is known to have relevance and is available. A data set like ADNI has many columns that are not self-explanatory, especially if the knowledge about medicine is limited. Given that studies [2] also show that demographics are rarely recorded in papers based on diagnosing tools, this problem of retrieving the relevant demographics might be a common issue. This is speculation originating from the experience of this project.

For this discussion, it should also be considered that remedying a bias might have a cost. For example, the central portion of patients in some given country is of that country's origin, so it would not affect them that a model might hold an ethnic bias. However, it would affect the natives if remedying the bias causes the performance of the natives to drop for the ethnic bias to disappear. There is always a limit to the amount of training data and information that one model can hold, and a compromise here might have an unforeseen cost. A possible solution is elaborated on in the next section. Conclusively, ensuring that no bias is present in a data set is nearly impossible. However, a great step towards this is to raise awareness and investigate the effects of a bias in common models, which was also the motivation for this project.

5.11 Suggestions for remedying biased data sets

Finally, now that it is concluded that limited data sets and algorithms are creates tremendous problems that might be hard to solve, suggestions for remedying this and the potential downfalls of those ideas are presented.

As suggested in [2] it could be considered to standardize the information that needs to be present in a data set in order to include it in a paper. However, this would require a massive amount of cooperation worldwide and would not necessarily eliminate the bias. It would make it general - one mistake in this standardization would impact everyone, but the mistake would be more straightforward to fix - a general change would remedy it. This is a limited approach that should be considered carefully. As discussed earlier, eliminating every bias is almost impossible even if it is tried to be aware of everything.

Another suggestion could be to use a toolkit like Aequitas [42] that can test your data set for bias if you use it as input. Although this might sound like a magical solution, this again raises the question of which factors one should be aware of and if these factors are present in the data set. This toolkit should also follow some conventions that are said to be the "ground truth," but how to ensure they did not miss something? A third way could be to be aware of the potential bias even before data collection begins. Even though it might be hard to know which demographics are relevant for a potential bias, some beforehand knowledge could be considered. For example, it would be intuitive that skin color could affect the ability to detect skin lesions or that gender (more specifically, breast size) could affect a lung scan. A way to improve it might be to balance how much of such data that would be needed to balance out the problem - maybe more data for the weak class should be provided as each data point would contribute with fewer features. However, the potential issue would be to find that balance and not unintentionally bias the algorithm in another direction trying to do this. The last thought could be related to the discussion above about an intentional bias. In some cases, it might be more beneficial to have several AI's that are "biased" in different ways and then pick the AI based on the demographics for the patient. This could make room for an AI to specialize in one particular problem like detecting lesions on black people and also remedy the fact that finding a lesion on light skin and dark skin might not be similar problems according to an AI . Even the architecture of the AI might need to be different. It could be speculated that the CNN for dark skin might need more layers since the problem is more complicated.

A problem that could arise from this would be that it requires more resources to train and develop different AIs, and it might also not always be clear which AI to use, which could result in a human bias when selecting the model.

6 Conclusion

This project aimed to investigate whether a bias in a training data set would propagate as a bias in a deep learning model.
Two neural networks with similar structures, one with fewer layers and comparable performance, were successfully implemented and trained on the American ADNI dataset. Firstly only 335 3T images were used for both training, validation, and testing, suggesting a difference in performance between ADNI/AIBL and I-ADNI. However, not nearly enough test data was present to make any safe conclusions for effect of ethnic bias.
So to investigate further, the models were trained on a training set balanced on both magnetic field strength and diagnosis consisting of 400 images. Here it was observed that, especially for the simplified model, the results became more stable because of the large data increase. It became clear that for further investigation, a simplified model trained on balanced sets of magnetic field strengths and diagnosis with as much training data as possible, yielded the most consistent results.
A general variation in performance was detected for this model. The difference was around 5 % between ADNI and I-ADNI (sensitivity), and no stable difference between ADNI and AIBL was observed suggesting that an ethnic bias is present when testing on I-ADNI. Furthermore, the effect of field strengths was investigated, and it was found that the performance was generally higher for 1.5T images differing up to 10 %. Gender bias was also briefly investigated but showed no significant difference. However, the inconsistency between the results and the low amount of test data makes these results very vulnerable.
In conclusion, it is stated with caution that the results of the experiments for this project suggest a concerning presence of an ethnic bias. It should provide great motivation for paying attention to the ethnic demographics of a training set in the future.

6.1 Further work

More experiments could be investigated for the influence of bias, starting with the other models that Basaia is presenting, differentiating, i.e., MCI and AD patients. Data sets from more countries should also be acquired, especially countries where racial differences are more significant. This could be Japan or India, where ADNI data sets are also available.
In this project, the important part of the demographic data was the origin of the subjects. However, an important question to be asked should be - which demographics are relevant? Studies have shown that many medical papers never reveal demographics from the data that

they use [2], and as we've argued, bias can be present in many unexpected demographic factors. The entire aspect of this should be investigated more thoroughly.
Finally, the European Commission is setting up some rules for the usage of AI [43]. They should ensure the security of humans and companies' basic rights, such that confidence for the performances of the AI's is ensured by ranking them on risk factors, where one of them is bias in data sets and models. It should be clearly explored what demands a data-set should fulfill to ensure such requirements and critically evaluated if such rules are even realistic to enforce in practice.

References

[1] Silvia Basaia, Federica Agosta, Luca Wagner, Elisa Canu, Giuseppe Magnani, Roberto Santangelo, and Massimo Filippi. Automated classification of alzheimer's disease and mild cognitive impairment using a single mri and deep neural networks. *NeuroImage: Clinical*, 21:101645, 2019.

[2] Samaneh Abbasi-Sureshjani, Ralf Raumanns, Britt E. J. Michels, Gerard Schouten, and Veronika Cheplygina. Risk of training diagnostic algorithms on data with demographic bias, 2020.

[3] Aisen P. S. Beckett L. A. Donohue M. C. Gamst A. C. Harvey D. J. Jack C. R. Jr Jagust W. J. Shaw L. M. Toga A. W. Trojanowski J. Q. Weiner M. W Petersen, R. C. The alzheimer's disease neuroimaging initiative (adni): Mri methods. 2010.

[4] Kathryn Ellis, Ashley Bush, David Darby, Daniela Fazio, Jonathan Foster, Peter Hudson, Nicola Lautenschlager, Nat Lenzo, Ralph Martins, Paul Maruff, Colin Masters, Andrew Milner, Kerryn Pike, Christopher Rowe, Greg Savage, Cassandra Szoeke, Kevin Taddei, Victor Villemagne, Michael Woodward, and David Ames. The australian imaging, biomarkers and lifestyle (aibl) study of aging: Methodology and baseline characteristics of 1112 individuals recruited for a longitudinal study of alzheimer's disease. *International psychogeriatrics / IPA*, 21:672–87, 06 2009.

[5] E. Cavedo, A. Redolfi, Francesco Angeloni, C. Babiloni, R. Lizio, L. Chiapparini, M. Bruzzone, D. Aquino, U. Sabatini, M. Alesiani, A. Cherubini, E. Salvatore, A. Soricelli, F. Vernieri, F. Scrascia, E. Sinforiani, P. Chiarati, S. Bastianello, P. Montella, D. Corbo, G. Tedeschi, S. Marino, A. Baglieri, S. De Salvo, F. Carducci, C. Quattrocchi, M. Cobelli, and G. Frisoni. The italian alzheimer's disease neuroimaging initiative (i-adni): validation of structural mr imaging. *Journal of Alzheimer's disease : JAD*, 40 4:941–52, 2014.

[6] What is alzheimer's disease? `https://www.nia.nih.gov/health/what-alzheimers-disease`.

[7] Facts: Alzheimer's disease. `https://www.beingpatient.com/alzheimers-disease-facts/`.

[8] Statistics: Alzheimer's disease. `https://www.alzheimers.net/resources/alzheimers-statistics/`.

[9] Diagnosing alzheimer's: How alzheimer's is diagnosed. `https://www.mayoclinic.org/diseases-conditions/alzheimers-disease/in-depth/alzheimers/art-20048075`. Accessed: May 8th, 2021.

[10] Takashi Ohnishi, Hiroshi Matsuda, Takeshi Tabira, Takashi Asada, and Masatake Uno. Changes in brain morphology in alzheimer disease and normal aging: Is alzheimer disease an exaggerated aging process? *American Journal of Neuroradiology*, 22(9):1680–1685, 2001.

[11] Taking care of our brain is just as important as taking care of other parts of our body. `https://www.identifyalz.eu/en/home/brain-health.html?utm_source=google-ads&utm_medium=search&utm_campaign=id-ad-general-public&utm_content=how-to-prevent-alzheimers&gclid=CjwKCAjwqcKFBhAhEiwAfEr7zVM6d6Y-gyVy-5PHl1ntmDwCc-JympOTW3zNN5-4C1Z_4EZP4T_8RRoChKkQAvD_BwE`. Accessed: May 8th, 2021.

[12] Yaakov Stern. Cognitive reserve in ageing and alzheimer's disease. *The Lancet Neurology*, 11(11):1006–1012, 2012.

[13] World Health Organization.

[14] Mary Reagan. Understanding bias and fairness in ai systems. `https://towardsdatascience.com/understanding-bias-and-fairness-in-ai-systems-6f7fbfe267f3`.

[15] Klaus Möllenhoff, Ana-Maria Oros-Peusquens, and N. Shah. Introduction to the basics of magnetic resonance imaging. 1971, 01 2012.

[16] The mni brain and the talairach atlas. `https://brainmap.org/training/BrettTransform.html`. Accessed: May 8th, 2021.

[17] Brodmann's areas of the cortex. `https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-79948-3_301`. Accessed: May 8th, 2021.

[18] Neural networks and their application on images. `https://github.com/camilla1237/Project-outside-course-scope---Neural-networks-and-their-application-on-images`.

[19] Adam: A method for stochastic optimization. `https://arxiv.org/pdf/1412.6980.pdf`. Accessed: May 8th, 2021.

[20] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[21] Sidharth Mishra, Uttam Sarkar, Subhash Taraphder, Sanjoy Datta, Devi Swain, Reshma Saikhom, Sasmita Panda, and Menalsh Laishram. Principal component analysis. *International Journal of Livestock Research*, page 1, 01 2017.

[22] K.J. Friston, J. Ashburner, S.J. Kiebel, T.E. Nichols, and W.D. Penny. *Statistical Parametric Mapping: The Analysis of Functional Brain Images*. Academic Press, 2007.

[23] Arno Klein, Jesper Andersson, Babak A Ardekani, John Ashburner, Brian Avants, Ming-Chang Chiang, Gary E Christensen, D Louis Collins, James Gee, Pierre Hellier, Joo Hyun Song, Mark Jenkinson, Claude Lepage, Daniel Rueckert, Paul Thompson, Tom Vercauteren, Roger P Woods, J John Mann, and Ramin V Parsey. Evaluation of 14 nonlinear deformation algorithms applied to human brain mri registration. *NeuroImage*, 46(3):786—802, July 2009.

[24] Batch jobs under lsf 10. `www.hpc.dtu.dk/?page_id=1416`. Accessed: May 16th, 2021.

[25] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[26] Pytorch documentation: conv3d. `https://pytorch.org/docs/stable/generated/torch.nn.Conv3d.html`. Accessed: May 8th, 2021.

[27] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[29] Yu Huang, Jian Yong Wang, Xiao Mei Wei, and Bin Hu. Bioinfo-kit: A sharing software tool for bioinformatics. In *Mechanical Science and Engineering IV*, volume 472 of *Applied Mechanics and Materials*, pages 466–469. Trans Tech Publications Ltd, 3 2014.

[30] The pandas development team. pandas-dev/pandas: Pandas, February 2020.

[31] Normalize dartel to mni. `HTTP://BRAINMAP.WISC.EDU/NORMALIZEDARTELTOMNI`. Accessed: April 8th, 2021.

[32] John Ashburner. A fast diffeomorphic image registration algorithm. *NeuroImage*, 38:95–113, 11 2007.

[33] Pytorch: Step by step implementation 3d convolution neural network. `https://towardsdatascience.com/pytorch-step-by-step-implementation-3d-convolution-neural-network-8bf38c70e8b3`. Accessed: May 8th, 2021.

[34] Mri acquisition. `http://adni.loni.usc.edu/methods/mri-tool/mri-acquisition/`. Accessed: May 20th, 2021.

[35] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net, 2015.

[36] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

[37] An updated overview of recent gradient descent algorithms. `https://johnchenresearch.github.io/demon/`. Accessed: May 8th, 2021.

[38] Søren Hauberg, Oren Freifeld, Anders Boesen Lindbo Larsen, John Fisher, and Lars Hansen. Dreaming more data: Class-dependent distributions over diffeomorphisms for learned data augmentation. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 342–350, Cadiz, Spain, 09–11 May 2016. PMLR.

[39] Nils R Winter, Micah Cearns, Scott R Clark, Ramona Leenings, Udo Dannlowski, Bernhard T Baune, and Tim Hahn. From multivariate methods to an ai ecosystem. *Molecular psychiatry*, May 2021.

[40] Johannson KA Wells AU Walsh SLF Assayag D, Morisset J. Patient gender bias on the diagnosis of idiopathic pulmonary fibrosis. 2020.

[41] ANGELA LASHBROOK. Ai-driven dermatology could leave dark-skinned patients behind. `https://www.theatlantic.com/health/archive/2018/08/machine-learning-dermatology-skin-color/567619/`.

[42] Pedro Saleiro, Benedict Kuester, Abby Stevens, Ari Anisfeld, Loren Hinkson, Jesse London, and Rayid Ghani. Aequitas: A bias and fairness audit toolkit. *CoRR*, abs/1811.05577, 2018.

[43] Europa klar til den digitale tidsalder: Kommissionen foreslår nye regler og tiltag for ekspertise i og tillid til kunstig intelligens. `https://ec.europa.eu/commission/presscorner/detail/da/ip_21_1682`. Accessed: May 8th, 2021.

[44] Adni webpage. `http://adni.loni.usc.edu`.

7 Appendix A - Details of the implementation

Both of the neural networks were implemented in python with PyTorch [25] and pseudocode can be seen in figure 19 and 20. The network itself was implemented as a class where each layer is an attribute, and the forward function determines which layers the input should be sent through and in what order. Their architecture is matching what was described in section 3.7. The full code with a readme for execution can be found here: `https://github.com/camilla1237/Master-thesis-code` and the trained models are available on the Thinlinc server.

```
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.conv_layer1 = self._conv_layer_set(1, 32,3, 1)
        self.conv_layer2 = self._conv_layer_set(32, 64, 3, 2)
        self.conv_layer3 = self._conv_layer_set(64, 128, 3, 1)
        self.conv_layer4 = self._conv_layer_set(128, 256, 3, 2)
        self.conv_layer5 = self._conv_layer_set(256, 512, 5, 1)
        self.conv_layer6 = self._conv_layer_set(512, 1024, 3, 2)
        self.conv_layer7 = self._conv_layer_set(1024, 2048, 3, 1)
        self.conv_layer8 = self._conv_layer_set(2048, 4096, 3, 2)
        self.fc1 = nn.Linear(1734656, 128) #6 layers
        self.fc2 = nn.Linear(128, num_classes)
        self.relu = nn.ReLU()
        self.out = nn.Sigmoid()
        self.batch=nn.BatchNorm1d(128)
        self.drop=nn.Dropout(p=0.15)

    def _conv_layer_set(self, in_c, out_c, ks, strides, batch=False):
        conv_layer = nn.Sequential(
                nn.Conv3d(in_c, out_c, kernel_size=(ks, ks, ks), padding=0, stride=strides),
                nn.ReLU(),
            )
        return conv_layer

    def forward(self, x):
        out = self.conv_layer1(x)
        out = self.conv_layer2(out)
        out = self.conv_layer3(out)
        out = self.conv_layer4(out)
        out = self.conv_layer5(out)
        out = self.conv_layer6(out)
        out = out.view(out.size(0), -1)
        out = self.fc1(out)
        out = self.relu(out)
        out = self.batch(out)
        out = self.drop(out)
        out = self.fc2(out)
        return out
```

Figure 19: The code for the simple model that was created during development.

```
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.conv_layer1 = self._conv_layer_set(1, 50,5, 1)
        self.conv_layer2 = self._conv_layer_set(50, 50, 5, 2)
        self.conv_layer3 = self._conv_layer_set(50, 100, 3, 1)
        self.conv_layer4 = self._conv_layer_set(100, 250, 3, 2)
        self.conv_layer5 = self._conv_layer_set(250, 400, 3, 1)
        self.conv_layer6 = self._conv_layer_set(400, 650, 3, 2)
        self.conv_layer7 = self._conv_layer_set(650, 800, 3, 1)
        self.conv_layer8 = self._conv_layer_set(800, 950, 3, 2)
        self.conv_layer9 = self._conv_layer_set(950, 1100, 3, 1)
        self.conv_layer10 = self._conv_layer_set(1100, 1250, 3, 2)
        self.conv_layer11 = self._conv_layer_set(1250, 1400, 3, 1)
        self.conv_layer12 = self._conv_layer_set(1400, 1600, 3, 2)
        self.fc1 = nn.Linear(19200, 128) #Basaia layers
        self.fc2 = nn.Linear(128, num_classes)
        self.relu = nn.ReLU()
        self.out = nn.Sigmoid()
        self.batch=nn.BatchNorm1d(128)
        self.batch1=nn.BatchNorm1d(512)
        self.drop=nn.Dropout(p=0.15)

    def _conv_layer_set(self, in_c, out_c, ks, strides, batch=False):
        conv_layer = nn.Sequential(
                nn.Conv3d(in_c, out_c, kernel_size=(ks, ks, ks), padding=1, stride=strides),
                nn.ReLU(),
            )
        return conv_layer
    def forward(self, x):
        out = self.conv_layer1(x)
        ...
        out = self.conv_layer12(out)
        out = self.fc1(out)
        out = self.relu(out)
        out = self.batch(out)
        out = self.fc2(out)
        out = self.out(out)
        return out
```

Figure 20: The code for the neural network inspired by the model by Basaia.

8 Appendix B - ADNI study schedule

Study Schedule	ADNI1																					
	Screen			Baseline			Month 6			Month 12			Month 18			Month 24			Month 30		Month 36	
	CN	MCI	AD	CN	MCI	AD	CN	MCI	AD	CN	MCI	AD	CN	MCI	AD	CN	MCI	AD	CN	MCI	CN	MCI
Explain Study	x	x	x																			
Obtain Consent	x	x	x																			
Demographics, Family History, Inclusion and Exclusion Criteria	x	x	x																			
Medical History, Physical Exam, Neurological Exam, Hachinski	x	x	x																			
Vital Signs	x	x	x	x	x	x	x	x	x	x	x	x		x		x	x	x			x	x
Height																						
Screening Labs	x	x	x																			
APOE	x	x	x																			
DNA Sample Collection for APOE Genotyping and GWAS																						
Cell Immortalization Sample Collection																						
American National Adult Reading Test				x	x	x																
Mini Mental State Examination	x	x	x				x	x	x	x	x	x		x		x	x	x			x	x
Logical Memory I and II	x	x	x							x	x	x				x	x	x			x	x
Everyday Cognition (Ecog)																						
Montreal Cognitive Assessment (MoCA)																						
Digit Span				x	x	x	x	x	x	x	x	x		x		x	x	x			x	x
Cateogory Fluency				x	x	x	x	x	x	x	x	x		x		x	x	x			x	x
Category Fluency (Animals)																						
Trails A & B				x	x	x	x	x	x	x	x	x		x		x	x	x			x	x
Digit Symbol				x	x	x	x	x	x	x	x	x		x		x	x	x			x	x
Boston Naming Test				x	x	x	x	x	x	x	x	x		x		x	x	x			x	x
Auditory Verbal Learning Test				x	x	x	x	x	x	x	x	x		x		x	x	x			x	x
Geriatric Depression Scale	x	x	x							x	x	x				x	x	x			x	x
Clock Drawing				x	x	x	x	x	x	x	x	x		x		x	x	x			x	x
Neuropsychiatric Inventory																						
Neuropsychiatric Inventory Q				x	x	x	x	x	x	x	x	x		x		x	x	x			x	x
ADAS-Cog				x	x	x	x	x	x	x	x	x		x		x	x	x			x	x
ADAS-Cog 13 (with Delayed Word Recall and Number Cancellation)																						
Clinical Demetia Rating Scale	x	x	x				x	x	x	x	x	x		x		x	x	x			x	x
Activities of Daily Living (FAQ)				x	x	x	x	x	x	x	x	x		x		x	x	x			x	x
Plasma and Serum Biomarker Collection																						
RNA Sample Collection																						
Collect and Process Biomarkers				x	x	x	x	x	x	x	x	x		x		x	x	x			x	x
Concomitant Medications	x	x	x	x	x	x	x	x	x	x	x	x		x		x	x	x			x	x
Subject Payments	x	x	x	x	x	x	x	x	x	x	x	x		x		x	x	x			x	x
Phone Contact													x		x				x	x		
Adverse Events	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Diagnostic Summary				x	x	x	x	x	x	x	x	x		x		x	x	x			x	x
MRI (1.5 T) (100%)	x	x	x				x	x	x	x	x	x		x		x	x	x			x	x
MRI (3 T) (25%)				x	x	x	x	x	x	x	x	x		x		x	x	x			x	x
MRI (3 T) (100%)																						
PET (50%)				x	x	x	x	x	x	x	x	x		x		x	x	x			x	x
^{18}F-AV-45 Amyloid Imaging (100%)																						
FDG-PET Imaging (100%)																						
LP (minimum of 20%)				x	x	x				x	x	x										
CSF Collection by Lumbar Puncture (100%)																						

Study Schedule	ADNI GO: New Participants						ADNI GO: Follow-Up from ADNI1			
	Screen	Baseline	Month 3	Month 6	Month 12	Month 18	Visit 1	Visit 2	Visit 3	Visit 4
	EMCI	EMCI	EMCI	EMCI	EMCI	EMCI	CN/MCI	CN/MCI	CN/MCI	CN/MCI
Explain Study	x						x			
Obtain Consent	x						x			
Demographics, Family History, Inclusion and Exclusion Criteria	x									
Medical History, Physical Exam, Neurological Exam, Hachinski	x						x			
Vital Signs	x	x		x	x		x		x	
Height	x									
Screening Labs	x									
APOE										
DNA Sample Collection for APOE Genotyping and GWAS		x					x		x	
Cell Immortalization Sample Collection		x								
American National Adult Reading Test		x								
Mini Mental State Examination	x			x	x		x		x	
Logical Memory I and II	x				x		x		x	
Everyday Cognition (Ecog)		x		x	x		x		x	
Montreal Cognitive Assessment (MoCA)		x		x	x		x		x	
Digit Span										
Cateogory Fluency										
Category Fluency (Animals)		x		x	x		x		x	
Trails A & B		x		x	x		x		x	
Digit Symbol										
Boston Naming Test		x		x	x		x		x	
Auditory Verbal Learning Test		x		x	x		x		x	
Geriatric Depression Scale	x			x	x		x		x	
Clock Drawing		x		x	x		x		x	
Neuropsychiatric Inventory										
Neuropsychiatric Inventory Q		x		x	x		x		x	
ADAS-Cog										
ADAS-Cog 13 (with Delayed Word Recall and Number Cancellation)		x		x	x		x		x	
Clinical Demetia Rating Scale	x			x	x		x		x	
Activities of Daily Living (FAQ)		x		x	x		x		x	
Plasma and Serum Biomarker Collection		x		x	x		x		x	
RNA Sample Collection		x			x		x		x	
Collect and Process Biomarkers										
Concomitant Medications	x	x		x	x	x	x	x	x	x
Subject Payments										
Phone Contact										
Adverse Events	x	x		x	x	x	x	x	x	x
Diagnostic Summary		x		x	x		x		x	
MRI (1.5 T) (100%)							x		x	
MRI (3 T) (25%)										
MRI (3 T) (100%)	x		x	x	x					
PET (50%)										
^{18}F-AV-45 Amyloid Imaging (100%)		x					x			
FDG-PET Imaging (100%)		x					x			
LP (minimum of 20%)										
CSF Collection by Lumbar Puncture (100%)		x					x			

Study Schedule	ADNI2: New Participants																		ADNI2: Follow-Up from ADNI1/GO		
	Screen		Basline		Month 3		Month 6		Month 12		Month 18		Month 24		Ongoing 6 Month Interim (phone)		Ongoing Annual		Initial	Ongoing 6 Month Interim	Ongoing Annual
	CN EMCI/LMCI	AD	CN EMCI/LMCI	AD	CN EMCI/LMCI	AD	CN EMCI/LMCI	AD	CN EMCI/LMCI	AD	CN EMCI/LMCI	AD	CN EMCI/LMCI	AD	CN EMCI/LMCI	AD	CN EMCI/LMCI	AD	CN EMCI/LMCI	CN EMCI/LMCI	CN EMCI/LMCI
Explain Study	x	x																	x		
Obtain Consent	x	x																	x		
Demographics, Family History, Inclusion and Exclusion Criteria	x	x																	x		
Medical History, Physical Exam, Neurological Exam, Hachinski	x	x																	x		
Vital Signs	x	x	x	x			x	x		x				x			x		x		x
Height	x	x																			
Screening Labs	x	x																			
APOE																					
DNA Sample Collection for APOE Genotyping and GWAS			x	x															x		
Cell Immortalization Sample Collection			x	x																	
American National Adult Reading Test			x	x																	
Mini Mental State Examination	x	x					x	x		x				x			x		x		x
Logical Memory I and II	x	x								x						x	x		x		x
Everyday Cognition (Ecog)			x	x			x	x		x				x			x		x		x
Montreal Cognitive Assessment (MoCA)			x	x			x	x		x				x			x		x		x
Digit Span																					
Cateogory Fluency																					
Category Fluency (Animals)			x	x			x	x		x				x			x		x		x
Trails A & B			x	x			x	x		x				x			x		x		x
Digit Symbol																					
Boston Naming Test			x	x			x	x		x				x			x		x		x
Auditory Verbal Learning Test			x	x			x	x		x				x			x		x		x
Geriatric Depression Scale	x	x					x	x		x				x			x		x		x
Clock Drawing			x	x			x	x		x				x			x		x		x
Neuropsychiatric Inventory			x	x						x				x			x		x		x
Neuropsychiatric Inventory Q							x	x				x			x	x				x	
ADAS-Cog																					
ADAS-Cog 13 (with Delayed Word Recall and Number Cancellation)			x	x			x	x		x				x			x		x		x
Clinical Demetia Rating Scale	x	x					x	x		x				x			x		x		x
Activities of Daily Living (FAQ)			x	x			x	x		x				x			x		x		x
Plasma and Serum Biomarker Collection			x	x			x	x		x				x			x		x		x
RNA Sample Collection			x	x						x				x			x		x		x
Collect and Process Biomarkers																					
Concomitant Medications	x	x	x	x			x	x		x		x		x	x		x		x	x	x
Subject Payments																					
Phone Contact										x					x	x				x	
Adverse Events	x	x	x	x			x	x		x		x		x	x		x		x	x	x
Diagnostic Summary	x	x	x	x			x	x		x				x	x		x		x		x
MRI (1.5 T) (100%)																					
MRI (3 T) (25%)																					
MRI (3 T) (100%)	x	x			x	x	x	x		x				x			x		x		x
PET (50%)																					
^{18}F-AV-45 Amyloid Imaging (100%)			x	x										x			x		x		x
FDG-PET Imaging (100%)			x	x										x			x		x		x
LP (minimum of 20%)																					
CSF Collection by Lumbar Puncture (100%)			x	x										x			x		x		x

Figure 21: Overview of the ADNI-data from the ADNI webpage [44].