Explainability of Deep Learning Models trained on MRI scans for Dementia identification

Charles Gallay

charles.gallay@epfl.ch

cg@visium.ch

Thesis submitted for the EPFL degree

Master of Science in Data Science

March 13, 2020

Under the joint supervision of

École polytechnique fédérale de Lausanne (EPFL)

Signal Processing Laboratory 5 (LTS5)

Prof. Jean-Philippe Thiran
jean-philippe.thiran@epfl.ch

Visium, Lausanne

Research and Development(R&D)

Axel Uran
au@visium.ch

CHUV

Laboratoire de recherche en neuroimagerie (LREN)

Kherif Ferah, PhD
ferath.kherif@chuv.ch

Acknowledgments

I would like to thank Ferath Kherif for the supervision he gave me on the project, especially with his medical expertise on dementia diseases.
I would also like to thank Timon Zimmerman for his helpful advice on training deep neural networks, Axel Uran for his supervision during my time in the company and all my colleagues at Visium[1] for the hints or ideas they might have provided me.
I would like to thank Pierre Gallay, Lucas Massemin and Matteo Togninalli for reading this thesis and for their feedback.
Last but not least, I would like to thank Lionel Clavien from InnoBoost[2] for providing me access to their computational resources.

[1] https://visium.ch/
[2] http://www.inno-boost.com/

Abstract

Dementia is a disease that specialists still have trouble understanding. Predicting whether a patient has dementia is done by clinicians based on cognitive tests and a scan of their brain.
We propose a method based on deep learning that automatically gives a diagnostic and a heatmap highlighting the regions of the brain that our model points to be responsible for the dementia. The explanation obtained by our model gave expected results, notably by highlighting the hippocampus of patients with dementia.
Therefore, we provided a proof of concept that machine learning technique can give a valuable output and helps clinicians in their decision process.

Table of Contents

1 Introduction

1.1 Motivation

According to the World Health Organization (WHO)[3], there are currently around 50 million people suffering from dementia around the word. Despite the already high number of cases and an expectation of 152 million patients by 2050, there is, as of today, no treatment to cure the disease or slow its progression down. However, the quality of life of the patients can be improved when the disease is detected in an early stage. There is therefore a need to build tools that can predict whether a person has the disease or not. Additionally, there is currently no clear understanding of the causes of dementia. Overall there is a need for a better understanding of the disease.
The recent improvements on the machine learning models and the overall better explainability of their outputs motivates their application to the field of dementia detection.

1.2 Problem Statement

This thesis aims at provides a machine learning model to automatically detect dementia. The outcome model has the constrain of having reasonable performances in terms of the different losses and metrics defined in section 2.3 and must be able to explain its predictions.
In our approach, we chose to work with a three-dimensional scan of the brain as input. Namely the raw T1-weighted Magnetic Resonance Images (MRI) of the patient brain. This data is a scan that encodes the anatomy of the brain.
Our final goal is in a first phase to feed these MRI to the Convolutional Neural Network defined in section 4.1 in order to obtain a prediction. In a second phase, we explain the previously obtained prediction using the fullgrad algorithm explains in section 2.4.3. Finally, the outcome of the two previous phases are visualized using the brain viewer from annex A. This brain viewer can be used by a clinician to see the original individual scan of the brain, the attention map and the predicted diagnostic.

1.3 Research Question Contributions

This thesis tackles the following questions:

- How can raw MRI scans be used as input to a deep learning model to predict dementia?
- Which information does a deep learning model use in order to make its prediction?

In order to answer these questions, the thesis provides the following contribution

- We present a prepocessing pipeline that prepares the MRI brain scans for a deep learning model.
- We introduce a deep learning model that predicts if a brain scans presents dementia.

[3] https://www.who.int/news-room/fact-sheets/detail/dementia

- We further present a model that visually shows the region of the brain the model used in order to give its prediction
- We develop a brain viewer tool that any non-machine learning expert can use to analyze the output and the explanation of the model.

1.4 Thesis Structure

This thesis starts with chapter 2 which quickly introduces the reader to the dementia diseases in section 2.1 before listing the different machine learning blocks that will be used during the entire work. This chapter end itself by describing, in section 2.4, the different techniques used to gain a better understanding of the model's outputs. The following chapter 3 describes the type of data used and how it has been preporcessed for the models. The thesis continues with chapter 4, which presents the different models and their architecture. In chapter 5 we briefly explain how the models were trained and the performances obtained from them. This chapter ends by showing the results on models explainability.
Finally, in the last chapter 6 we conclude the thesis and propose some future work that could be done to improve either the performance or the explanabilty of the models.

2
Background

2.1 Medical Background

This section introduces the reader to the dementia disease, some of its symptoms and explain why it is not a normal stage of life.

2.1.1 Dementia

Dementia is not a specific disease but an overall term used to group all diseases that are characterized by a decrease in memory, thinking capabilities, language and other skills related to the thought. With between 60 and 70 percents[4] of the cases, Alzheimer's Disease (AD) is the most common disease-causing dementia. Despite the fact that dementia is affecting mostly old people, it is not a normal stage of life and therefore should not be seen as severe ageing.

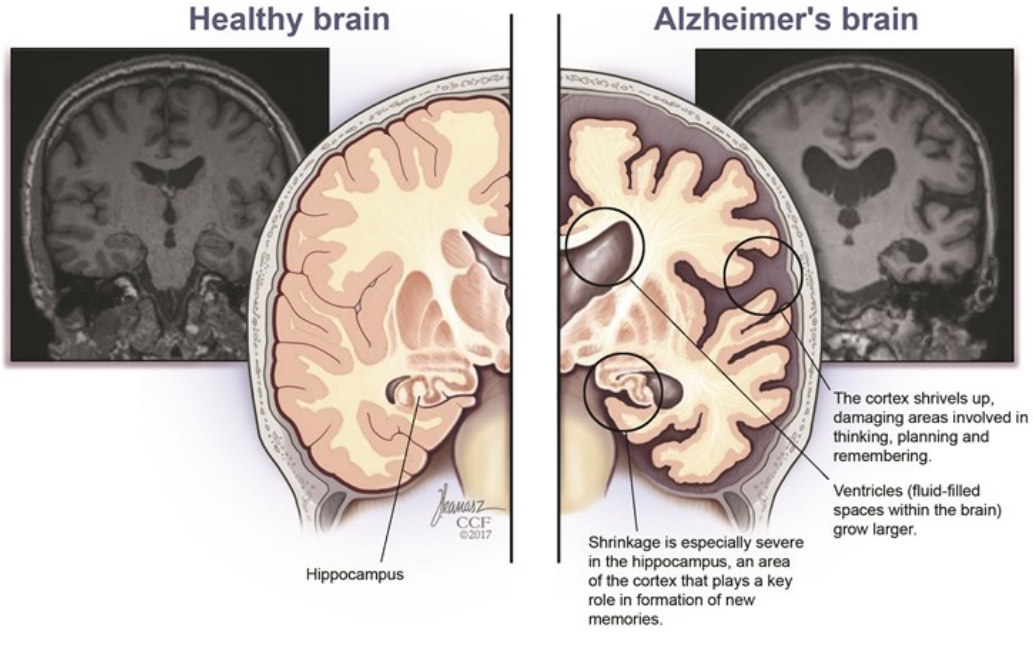


Figure 2.1: Illustration[5]of the differences between a healthy brain and the brain of someone with Alzheimer disease.

Figure 2.1 highlights the visual differences that practitioners can use to diagnose a Alzheimer patient. Knowing which part of the brain is affected by the disease will help us evaluate our model. We expect a good model to focus his attention on these brain areas in order to make its prediction. In fact, one would more easily trust a model that has the same attention locations as a clinician, in agreement with

[4] https://www.who.int/news-room/fact-sheets/detail/dementia
[5] https://www.keepmemoryalive.org/brain-science/alzheimers-brain

the medical knowledge.

2.1.2 Current Methods

Most current methods in dementia research are based on the on comparing MRI scans from healthy participants to the scans from patients with dementia. this required that all the brain scan are aligned to a common template, as describe in section 3.3.3 The goal is that a voxel at a specific coordinate is mapping to the same brain region on both images. This also allows to remove the large differences of brain anatomy between subjects. For completeness of the study, we performed this method called Voxel-Based Morphometry[1] in annex C.
Some classical machine learning[2] approach such as SVM[3], logistic regression and Random Forest[4] have been use to detect dementia. These methods require the data to be preprocessed and leak at explaining their outputs.

2.1.3 Age vs. Dementia

It is tempting to see dementia damaged brain as an extreme ageing process. In both cases, we observe a loss of white matter due to neuron death. Even though scientists currently do not know exactly what causes the disease, they do observe the presence of biomarkers such as plaque and tangle [5].
Despite the differences, researcher[6] have built simple dementia detectors by training an age predictor on healthy brains. It appears that their predictor performs badly on dementia's brains and tends to overage people with the disease. As we trained an age predictor on healthy brains, in annex B we took the opportunity to try it on our data. In fact, as shown in the annex, it turns that our predictor is not able to distinguish between healthy and sick brains as it can be seen in figure B.2. This tends to show that indeed the assumption is wrong and that the people with dementia do not have an overaged brain. In general, it would be more interesting to build a model that is able to detect dementia regardless of the patient's age, therefore reducing as much as possible the bias of the model towards age.

2.2 Model Blocks

This section defines fully connected and convolutional layers which are the two machine learning blocks that we are using in this project.

2.2.1 Fully Connected

One of the most basic components when building a deep learning model is the fully connected (FC) layer. Mathematically it is defined as:

$$Y = X * W + b$$

Where the outputs Y are expressed as a linear combination of the inputs X using a weight matrix W and a bias b. It has been proven by the well-known *universal approximation theorem*[7] that such layers combine with non-linear functions can approximate in theory any continuous functions.

2.2.2 Convolutional Neural Networks

The fully connected layer presented above is really general and works well with a large variety of data. Unfortunately, for certain types of data, using FC layers can quickly become unscalable. For example, when dealing with images of size 200 by 200 in color (RGB), a layer with 100 features output would already require 12'000'100 parameters to be learned (including bias).
Convolutional Networks[8] were introduced by LeCun to minimize the processing done on the input image

and let the network learn the right set of features. For it to be working, convolutional layers have to make assumptions on the data. One is spatial locality, where the pixels close to each other are more likely to be correlated than distant ones. In fact, CNN can be seen as small FC layers that are applied at multiple locations across the image. These are usually called kernels. The assumptions made by CNN reduce the number of parameter of the network. This helps for regularization and the network requires less data to be train.
This way of computing the inputs does have multiple advantages, one of them being weight-sharing. The kernel being applied at multiple locations, the weights learned to extract a feature in one location are by construction reused to extract the same features everywhere else in the image. Another property of CNN that comes directly from their formulation is the fact that translating the input, results in a translated output, this is called translation equivariance. Nowadays, CNN has become the standard way of processing images.

In our case, we are dealing with 3D images (MRI) that are scientifically larger than standard 2D ones, therefore it makes even more efficient to use 3D convolutions. As illustrated by figure 2.2, 3D CNN are really similar to 2D CNN. The only difference between 2D CNN and 3D CNN is the shape of the learned kernels, which are three-dimensional in order to fit the input.

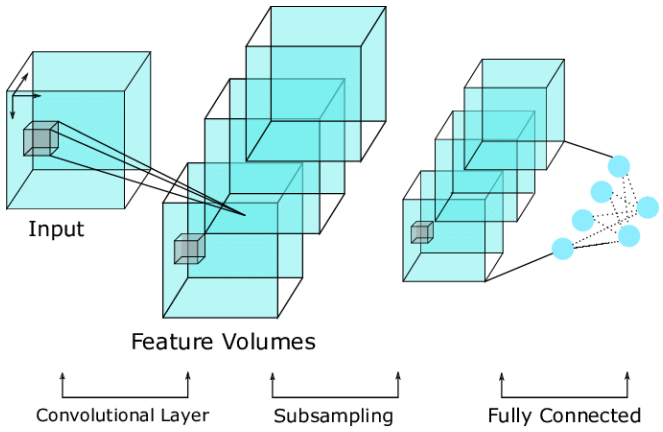


Figure 2.2: Illustration[6] of a 3D CNN

2.3 Losses and Metrics

This section lists the different metrics, and losses used during this project.

2.3.1 Binary Cross Entropy

This loss is used to measure the error made on binary classification.

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

Where p is the predicted probability of belonging to the class made by network and y the binarized true label.

2.3.2 Mean Square Error (MSE)

This loss is used to penalize errors made by a network for example in the case of a regression. It works with continuous values and is defined as:

$$MSE = \frac{\sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2}{n}$$

Where $\hat{Y}_i$ is the predicted value and Y_i is the target value, n is the total number of samples and i the running index of a sample in the summation.

[6] https://www.researchgate.net/publication/330912338_ECNN_Activity_Recognition_Using_Ensembled_Convolutional_Neural_Networks

2.3.3 Mean Absolute Error (MAE)

This metric is really similar to the MSE, but we take the absolute value of the error instead of taking the square.

$$MAE = \frac{\sum_{i=1}^{n} |(Y_i - \hat{Y}_i)|}{n}$$

Where $\hat{Y}_i$ is the predicted value and Y_i is the target value, n is the total number of samples and i the running index of a sample in the summation.

2.3.4 Accuracy

The accuracy is one of the simplest metrics to evaluate a model. It is computed by counting the number of correctly classified samples divided by the total number of samples.

2.3.5 Precision and Recall

While accuracy is simple to understand and to visualize, it can often fail at evaluating the performance of a model when the dataset has unbalanced classes. To overcome this issue other metrics can be computed based on the true/false positive and true/false negative.

- **True positive (TP)**: Positive sample correctly classified as positive.
- **False positive (FP)**: Negative sample badly classified as positive.
- **True negative (TN)**: Negative sample correctly classified as negative.
- **False negative (FN)**: Positive sample badly classified as negative.

Precision can thus be computed as below and gives a sense of how precise the classifier is on dementia predicted brains.

$$Precision = \frac{TP}{TP + FP}$$

Recall on the other hands can be computed as below and gives a sense of how likely the model is to detect dementia when the patient is indeed ill.

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

In a medical situation, it might be more interesting to have a model with a good recall at the expense of precision, namely having more false positive. The consequences of being detected as healthy when in fact the patient suffers from the disease are often worse than the opposite.
Accuracy can also be computed with these terms.

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

2.3.6 ROC - AUC

Receiver Operating Characteristic allows for plotting a chart like figure 5.4a that visualizes the true positive rate against the false positive rate at different thresholds. From this curve, we can compute the Area Under the Curve (AUC) which is a good metric to compare different models. Usually, a bigger AUC means a better model, the rise of the curve is even more important. In an ideal situation, we expect the curve to touch the top left corner and have an AUC of 1.

2.3.7 PR - Curve

The Precision and Recall curve is often used in replacement of the ROC curve when dealing with unbalanced data. It plots the precision and recalls of the model is computed at different thresholds. An example of PR curve can be seen in figure 5.4b.

2.4 Model Explanation

There exist multiple ways to explain the output of a model. We tried different approaches that are either specific to computer vision tasks or more general and compared them for our specific task of dementia prediction.

2.4.1 Shap

Shap is an algorithm based on Game Theory that aims to predict the contribution of a feature to increase the confidence of a model. Its mathematical background lies on the Shapley value which basically is the average of the marginal contributions across all permutations of features. Figure 2.3 shows what the outcome of the shap algorithm look link on image data.

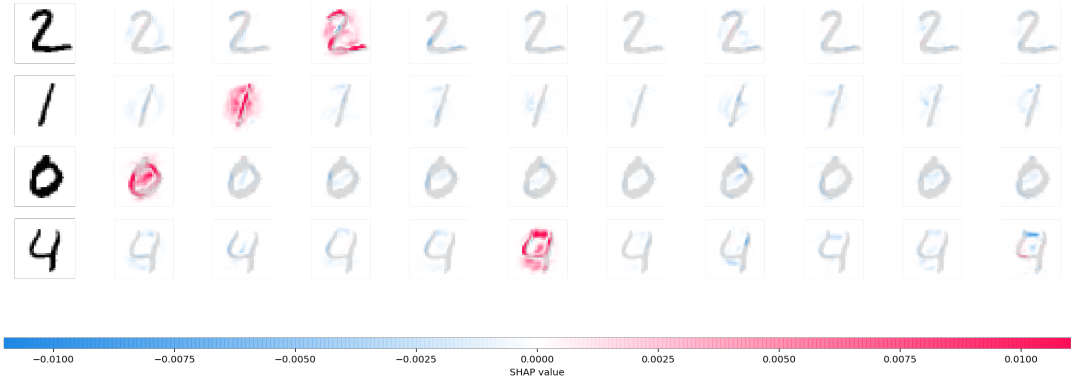


Figure 2.3: Example of shap value from the Shap repo[7]. Each column represents the shap output for a specific class (in a sorted order).

The library[9] works with different kinds of models but it was tested with the GradientExplainer as DeepExplainer is not fully compatible with Pytorch yet. Compared to some other explainer algorithm, Shap presents the advantage of giving a negative value for a feature which has a negative correlation with the output. The output obtained by this process can be seen in figure 5.5, but we can see that it does not give a good interpretation in comparison with other techniques.

2.4.2 Grad-Cam

When the input is an image, analysts are interested in finding out which part of the image better explains the prediction. In his paper[10] Zhou, proposes a specific model for which he could create a *class activation maps*. This idea has been generalized to work with any model by the authors of grad-cam.
The Grad-Cam[11] algorithm is looking for activation of the neurons at a specific layer. To do so the input is processed in a forward pass by the network until the final prediction is made. The gradient of the predicted class with respect to the layer of interest is then computed. It is then pooled across its spatial dimension in order to get a value of layer importance per channel. The features extracted from the image are then multiplied channel-wise by the pooled gradient in order to get an activation map (also called

[7] https://github.com/slundberg/shap

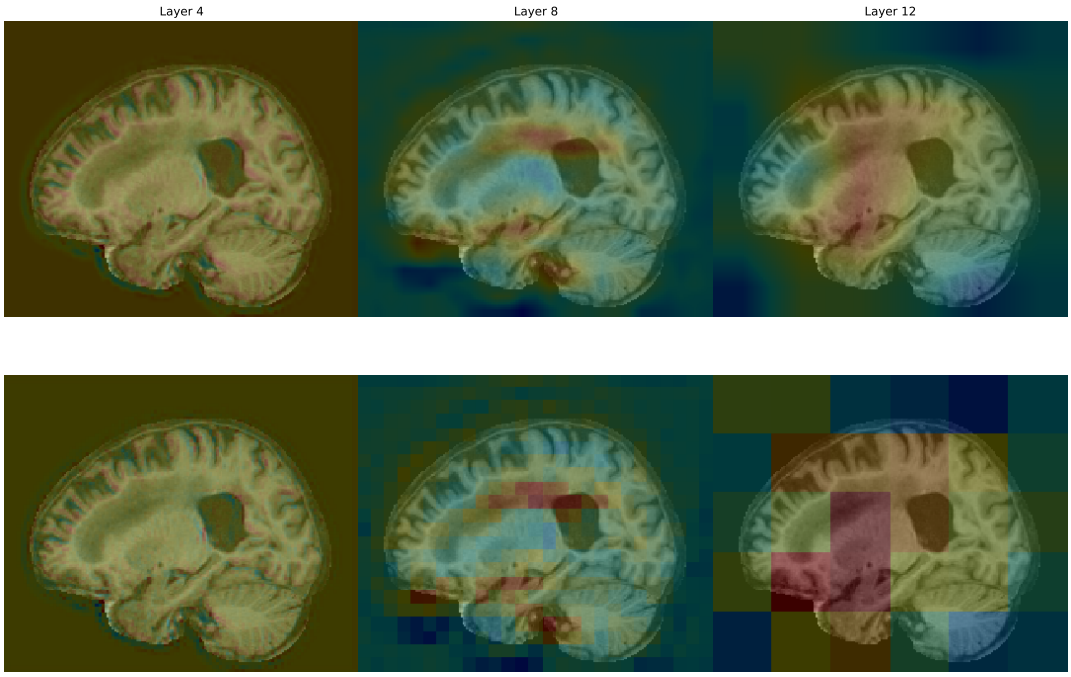


Figure 2.4: GradCam saliency map seen at different layers. Trilinear interpolation is used in the first row and nearest interpolation in the second one.

saliency map) of the same shape at the output of the layer of interest. This can be mapped to the original image shape by interpolation. For visual purposes, we choose to do a trilinear interpolation, which is the extension of the bilinear interpolation in three dimensions. A visual comparison of the outcome can be seen in figure 2.4. This algorithm is well schematized in figure 2.5.

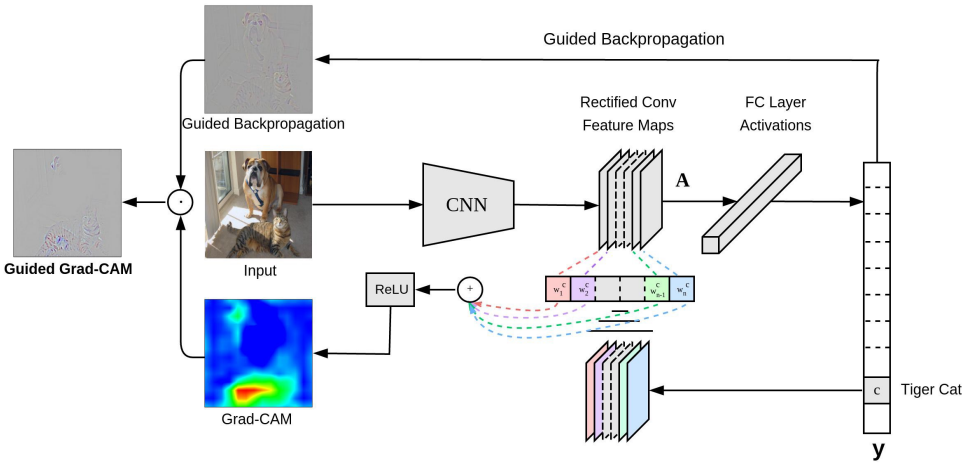


Figure 2.5: Schema of the GradCam algorithm taken from the paper[11]

When the model goal is to perform multi-class classification, it makes sense to apply a Relu function to the activation map in order to remove the negative value that could be explaining any other classes in the image. In our case, as we are dealing with a binary classification task, it might make sense not to remove negative values and let a clinician decides which map is more useful for him.
The output size of the layer one tries to explain will determine the granularity of the activation map. The artifact due to shrinkage of the network can be seen in figure 2.4. This imposes a trade-off when constructing the model, either last convolution layer output is too small in order to keep a good focused explanation or it is too big but this induces a bigger model which then needs more data to be trained.

Guided backpropagation[12] is another visualization to explain what a model has learned. Compared to grad-cam it gives a sharper explanation map but fails at giving an explanation for a specific class. To get the best of both, it is possible to multiply pointwise the two outputs as illustrated on the left side of figure 2.5.

2.4.3 FullGrad

Developed by Sriniva and Fleuret at EPFL, FullGrad[13] introduce a new tool for neural network interpretability that satisfies both *completeness* and *weak dependence on inputs*. Completeness can be seen as the property that the saliency map contains all the information necessary to compute the output of the model.

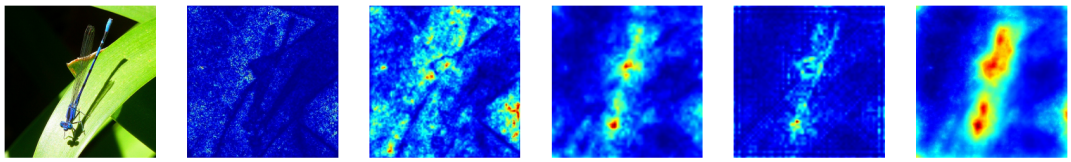


Figure 2.6: Visualization[13] of bias gradient at different layers and the output of fullGrad which is an aggregation of the input gradient and all the intermediate layers.

A saliency map is complete if there exists a function ϕ such that:

$$\phi(S(x), x) = f(x) \quad \forall x, f$$

Weak dependence on inputs, on the other hand, is a property that one gets when slightly changing an important pixel drastically affects the output of the model. Previous methods were not able to have these two properties at the same time.

As visualized in figure 2.6 the output of the fullGrad algorithm is an aggregation of the gradient at multiple layers. Therefore, compared to GradCam there is no need to define a layer of interest which is usually set to the last convolutional layer. This also makes the algorithm less sensitive to the potentially small size of the last convolutional layer.

3 Data

The data used for the analysis of the brain are Magnetic Resonance Imaging (MRI) scans. This type of scanning is often used as it captures the structure of the brain without the need for exposing the patient to radiation (in contrast to X-ray).

3.1 MRI Images

MRI scans are interested in detecting hydrogen in the body. In fact, this element is widely present in water and fat which makes it interesting in order to analyze the inner workings of the human body.

In normal state protons inside the hydrogen nucleus are spinning in a random direction, but when placed into a strong magnetic field (usually between 0.2 and 3 teslas which is about 10'000 times stronger than magnet people usually use on their fridge). They will align their spin with the field.
Approximately half of them will end up facing the field and the other half in the same direction as the field. In fact, a few more protons will line up their spin in a low configuration. These extra protons, despite being only a few, are the interesting ones.
The second phase consists of sending a specific radio frequency (RF) pulse. The extra protons currently in low-energy configuration will absorb this pulse and flip on their axes. When the RF pulse is stopped, the protons will return into their low-energy configuration and doing so emit RF waves. As different tissues of the brain contain different hydrogen density, this map highlights the different tissues of a brain well, especially the white and gray matter.

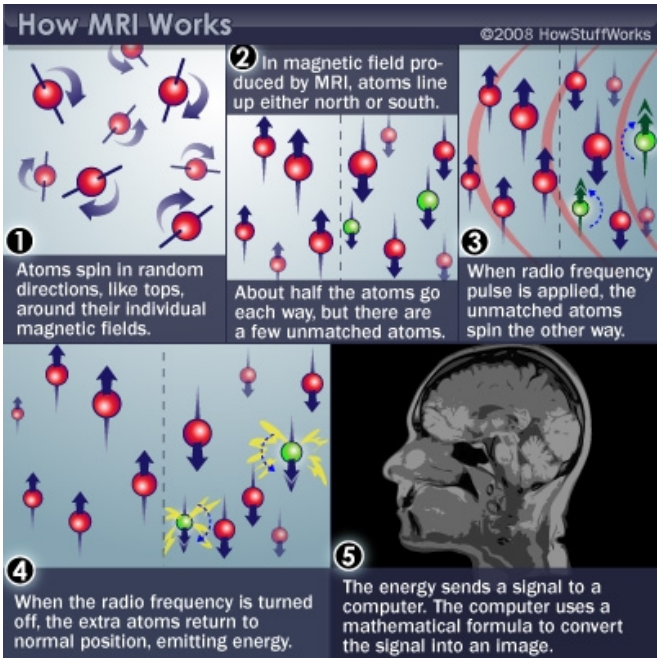


Figure 3.1: Steps to acquire an MR Image. Credit howstuffworks.com[8]

Some parameters of the machine can be tuned in order to produce different maps (also called modalities). One of the parameters that can be tuned is the amount of time between two pulses are sent. In fact, two of the most often used modalities (T1 and T2) differ by this parameter. For the rest of the report, when talking about MRI we are indeed referring as T1-weighted images.
As shown later, knowing how these images were formed is quite important, when preprocessing them.

[8] https://science.howstuffworks.com/mri3.htm

3.2 Datasets

This section explains the different datasets used as well as some basic statistics about them.

3.2.1 IXI

IXI dataset consists of approximately 600 MR Images from healthy patients. Each in T1, T2 and PD-weighted modalities, but for our application, we will concentrate on T1-weighted images only. The data has been collected in 3 different hospitals. For each image, we have the age of the patient together with other information such as if the patient is right or left-handed and his sex. As visualized on figure 3.2, the scans come from different hospitals. Further analyses as highlighted in figure 3.3 show that there is a bias on age due to data coming from different hospitals. This could let the model learn where the IRM comes from in order to predict the age. For example, by detecting that an IRM comes from the IOP hospital, the model could output an age around 30 years old without being terribly wrong. Of course, as we do not want this, it is important to normalize our data as illustrated by figure 3.7 of the preprocessing pipeline.

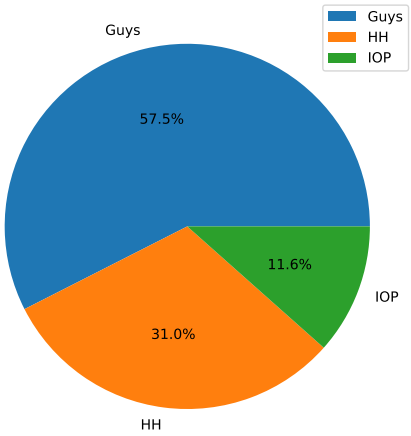


Figure 3.2: Scans percentage per hospital.

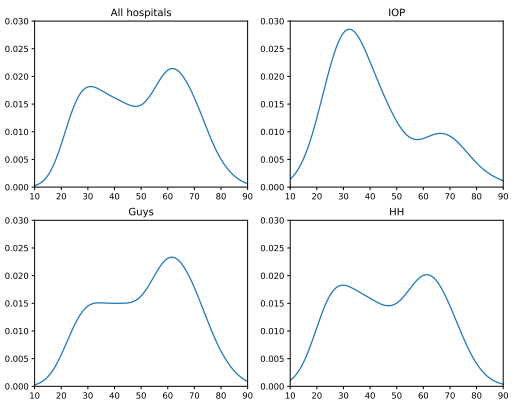


Figure 3.3: Age distribution for each hospital.

3.2.2 OASIS

OASIS[14] is the main dataset we used in order to train and conduct our analysis of dementia detection and explanation. It consists of 1098 patients for which we can have multiple scans across time. This has to be taken into account when splitting the dataset into train and test sets. If this information is omitted, half scans of a patient could end up in the train set and the other half into the test set, leaking some information between the two sets that are not independent anymore, heading to unreliable evaluation metrics.

Having multiple samples for one patient comes from the longitudinally of the dataset, this influenced the way we attribute labels to a sample. In the dataset, one patient can have multiple diagnoses depending on when he was diagnosed. For example, on the first check up the patient might have been diagnosed healthy, but when checked again 5 years later, his diagnosis might have changed and he might now be classified with dementia. We choose to deal with this situation by taking for each patient his latest (worst) diagnosis as the label for each sample. This choice is motivated by two reasons, firstly we want to build a model that detects dementia as early as possible. Therefore these samples where human failed

at making a correct early diagnosis are helping the model into learning features to detect the illness even in situations where the human eye might fail. Secondly, it has been shown by researchers [15] that the changes in the brain already occur 10 to 20 years before the first symptoms are observed.

	max CDR					
min CDR	0	0.5	1	2	3	Grand Total
0	609*	192	39	12	2	854
0.5		66	61	45	5	177
>1			31	31	5	67
Grand Total	609	258	131	88	12	1098

Figure 3.4: Number of patients per CDR in the OASIS dataset[9].

The dataset does not contain a final diagnosis of whether or not a patient has dementia with a certainty of 100 percent. The label associated with each scan is called *clinical dementia rating* (CDR)[10]. This particular diagnosis is based on an interview of both the subject, his caregiver and a clinician. It aimed at testing different cognitive domains such as memory, problem-solving or orientation. The results range from 0 to 3, where 0 stands for no dementia, 0.5 questionable dementia, 1 mild, 2 moderate and 3 severe cognitive impairment. For simplification, we decided to go for only two classes, either dementia or no dementia. Images with CDR 0 are labeled as control image (no dementia) and images with CDR either one two or three were labeled as having dementia. We choose to discard images with a CDR of 0.5 as these might be confusing for the model. The distribution of CDR can be observed in figure 3.5

Dementia being often diagnosed for old people, we want to check how this is affecting our dataset. Figure 3.5 clearly highlights something important about the age distribution in the dataset. We see that people with dementia tend to be older. To mitigate this effect while training, we build as described in section 4.3 a model that tries to extract features from the image that contains enough information to predict dementia but do not allow the model to predict the age of the patient.

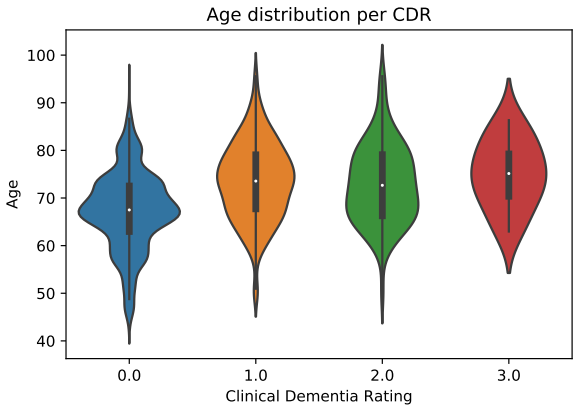


Figure 3.5: Distribution of patient age for each class of dementia rating. The width of a violin represents the number of persons for a specific age.

3.3 Preprocessing

Preprocessing is especially important in medical imaging. For example, a model could easily detect from which hospital/machine the data has been acquired, just by looking at the intensity distribution of voxels if no preprocessing is done. In the case where the data is biased on some hospitals, as it is the case in figure 3.3, the model could easily overfit on this bias. In fact, due to the expensive price of acquiring data, datasets are often created by merging data from multiple hospital institutes. In addition, we do not want our model to work well on one hospital only, but should generalize as much as possible for other hospitals too. The section below describes in detail the different stage of our preprocessing pipeline to overcome these issues. The overall preprocessing pipeline has been visualize in the figure 3.8.

[9] https://www.oasis-brains.org/files/OASIS-3_Imaging_Data_Dictionary_v1.5.pdf
[10] https://www.sciencedirect.com/topics/medicine-and-dentistry/clinical-dementia-rating

3.3.1 Resampling

MRI scanners can have different resolutions that directly come from the machine used to acquire the data. To overcome this we use the metadata present in the MRI header to resample the images such that the effective distance between two neighboring voxels is exactly 1mm. This might require a drop of resolution or interpolation in the case where the original image was sampled at a lower resolution.

3.3.2 Bias Field Correction

In medical imaging, one would logically expect that the intensity of a certain type of tissue should always be the same across the image. For example, the value of a voxel representing white matter should be the same independently of its location in the image. However this expectation does not match the reality, there are numerous undesirable artifacts implied by the way the data is recorded. These are often referred to as bias field, and sometimes as shading, intensity non-uniformity/inhomogeneity. Fortunately for us, this non-uniformity can be assumed to be of low-frequency and thereby to be smooth across the image, which means that it can be estimated and corrected or at least mitigated.

One can try to correct the bias by modeling the recorded image as a combination of 3 components, the clean image I, the bias field of low frequency B and an independent Gaussian noise σ. This formula is one possible way to combine these components. $S(x, y, z) = I(x, y, z) * B(x, y, z) + \sigma(x, y, z)$. The algorithm we used, assumes this formula, for bias field correction. The low-frequency bias field B can be seen in red in figure 3.6. It is an improvement over the *non-parametric nonuniform normalization (N3)* and is therefore named N4. The exact computation of the bias goes beyond the scope of this thesis, but the reader can find the details in the original paper [16].

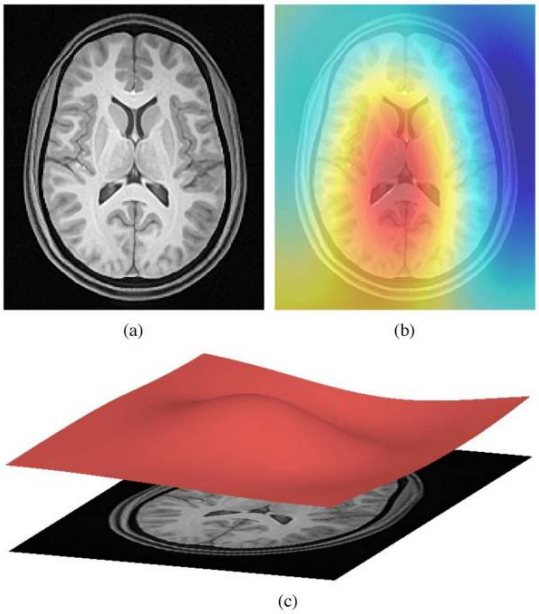


Figure 3.6: Bias field correction by N4ITK[11]

3.3.3 Co-registration

In the 3D space, the brain can be located anywhere and more precisely rotated or scaled by any factor. As all the models used in this study are based on convolutional layers, it is well known that they can be equivariant or even invariant to translation, but they are not rotation nor scale equivariant. Thus it would greatly help the model to transform the brain scans from all patients so that they are all realigned into a common space.
Here we chose to align the brain into the MNI152 space[12]. The template we used consists of an averaging of 152 healthy brains (healthy subjects) that have been matched using a 9-parameter affine transform. It is commonly used as a standard template. In addition to the averaged brains, this space does provide some useful masks, notably interesting for us a mask of the skull and one of the hippocampus.
Some specialists call this process normalization, while co-registration is used to describe the process by which different modalities from the same patient are realigned. This process is slightly easier as the type of transformation between images can be restricted to affine transformation (linear mapping method that

[11] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3071855/
[12] https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/Atlases

preserves points, straight lines, and planes)
Registration can be viewed as an optimization problem formally defined below. Find p that satisfies

$$\min_{p} M(I, J, \omega(p))$$

Where ω is a parametric transformation from a set of allowed transformation. Usually, the metric used to compare the two images is mutual information. The higher the mutual information is between two images the closer they are (similar look).
The algorithm used internally by ANTs[13] is the following:

1. Find the optimal, rigid transform that minimizes the negative Mutual Information.
2. Find the optimal, affine transform that minimizes the negative Mutual Information.
3. Find the optimal diffeomorphism that minimizes the negative Mutual Information.

As only invertible transformations are applied, the resulting output from this algorithm is a register image and an invertible transformation. The fact that the transformation is invertible is quite important as it means that, in total, the information has not been lost but it is now shared between the registered image and the transformation matrix.
Unfortunately, some useful features for the task could now be encoded into the transformation and dealing with the registered image only could lead to bad performances. For this reason, the next steps of the pipeline are going to be executed on both the registered image and the non-registered one. In further work, it would be interesting to train the model on the non-registered image to compare the performances.
The optimization task cannot be implemented in one step due to the high number of parameters to tune. Therefore an iterative approach based on the gradient is implemented. This solution might not be the optimal one but does converge towards it. This makes it by far the most computation expensive step of the pipeline, it takes roughly 10 minutes per image to be realigned with the common brain. For this reason, we had to parallelize the preprocessing pipeline using programs such as GNU parallel[14].

3.3.4 Skull stripping

This step removes the skull from the image. Doing so is quite easy once the brain has been remapped into MNI152 space, we can simply multiply the voxel value with the inverted skull mask to obtain the brain without a skull.
By doing so we want to ensure that the model will not try to learn information from the skull as we make the assumption that dementia to be located in the brain.

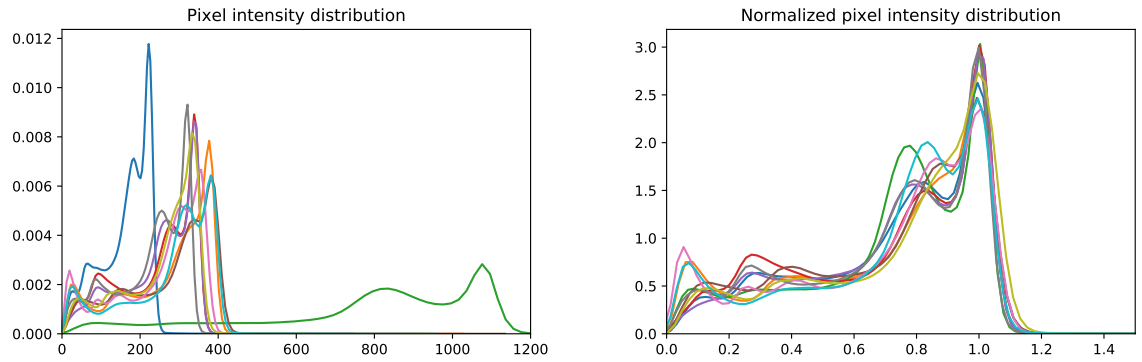
3.3.5 Intensity normalization

During the N4 bias correction step, we took care of uniforming the intensity of a given tissue across the whole image. The goal of this step is to do the same but across the whole dataset. To do so we will detect the intensity of a white matter voxel and set it to one. We can see in figure 3.7a how much the intensity of the voxels can vary from two samples taken by different machines. The output of the normalization as shown in figure 3.7b is now much more uniform across machines and therefore less prone to overfitting.

3.4 Data augmentation

Data augmentation is an extra step of the training pipeline that is done right after the data has been preprocessed. One technique to overcome overfitting is to create new training samples for which we already

[13] https://stnava.github.io/ANTs/
[14] https://www.gnu.org/software/parallel/



(a) Before voxel intensity normalization

(b) After voxel intensity normalization

Figure 3.7: Voxel intensity distributions of 10 randomly chosen samples from the OASIS dataset.

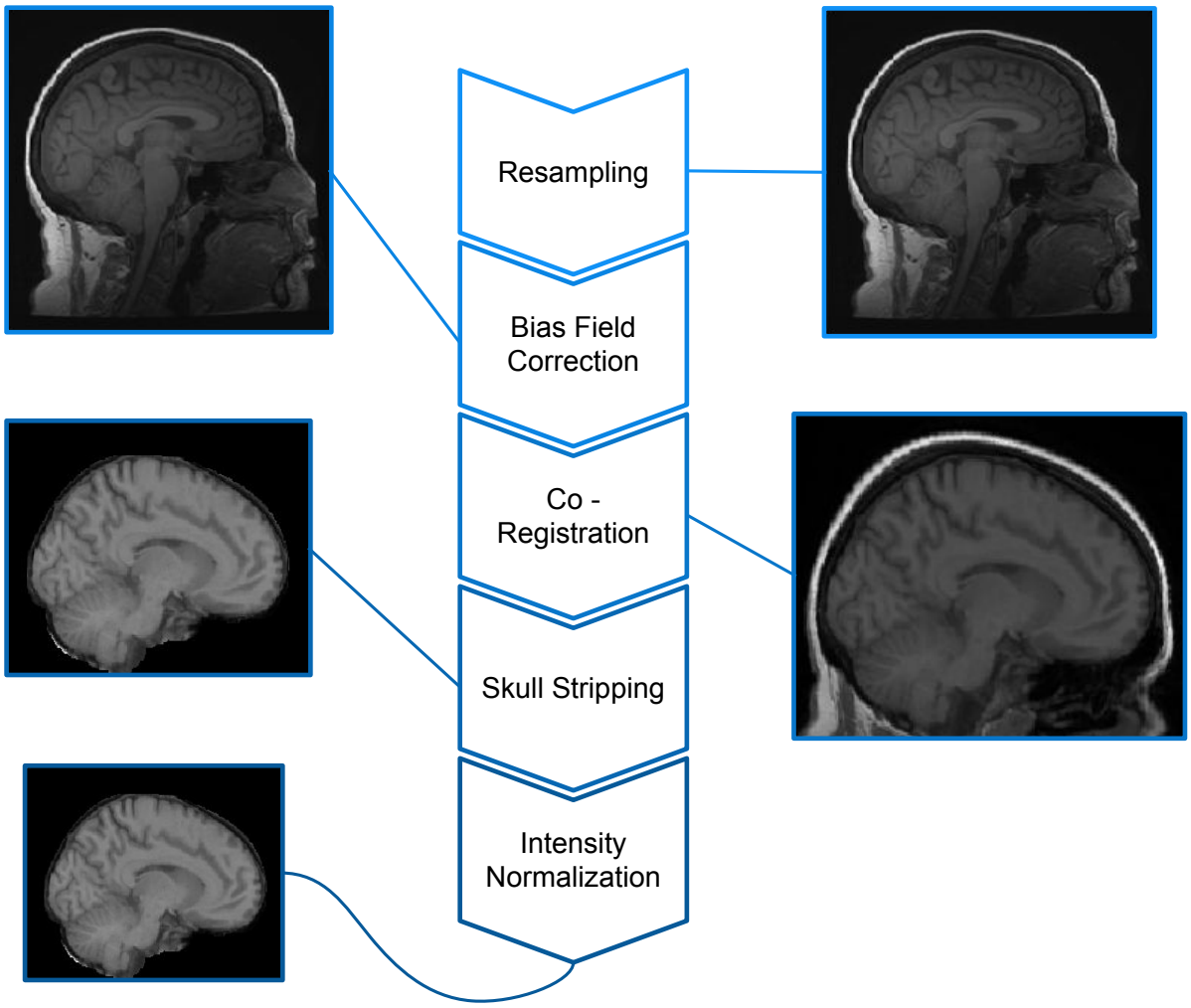
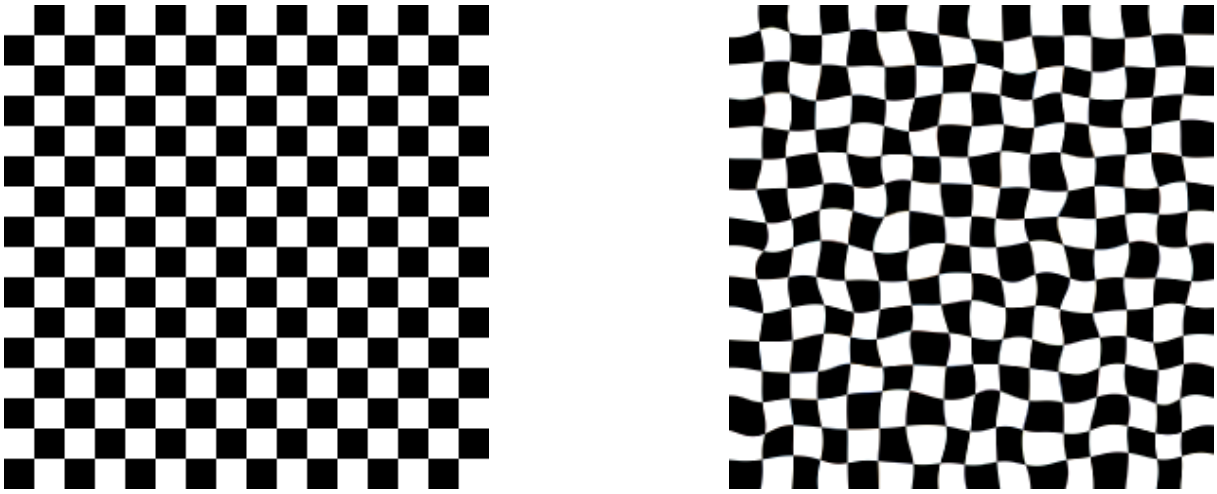


Figure 3.8: Preprocessing pipeline and its outputs after each steps.

know the labels. This trick can be applied to any type of data, but it is often used when dealing with pictures. Usually one applies a transformation to the input image, such that the model activity changes when the newly generated image is shown to it, but the transformation should be labels invariant. For example, it is known that CNN are translation equivariant, but not equivariant to rotations. Therefore AI expert often apply small random rotation to the image in order to augment their training set. Note that the transformation applied should be chosen carefully, as for example when dealing with a dataset



(a) Input grid (b) transformed grid

Figure 3.9: Example[15] of the elastic transformation applied to a grid (with alpha=991, sigma=8).

such as MNIST if a rotation of more than 90 degrees is applied, it might become impossible for the network to distinguish between a rotated 6 and a rotated 9. In our case, we choose to apply random noise, small voxel intensity variation, flipping, small rotation and elastic transform. Figure 3.10 shows the different effect of the transformation on a brain scan.
While most of the transformation applies are common, the elastic one is less known. It consists of applying a smooth deformation to the image. Figure 3.9 illustrates this transformation on a grid.

[15] https://gist.github.com/erniejunior/601cdf56d2b424757de5

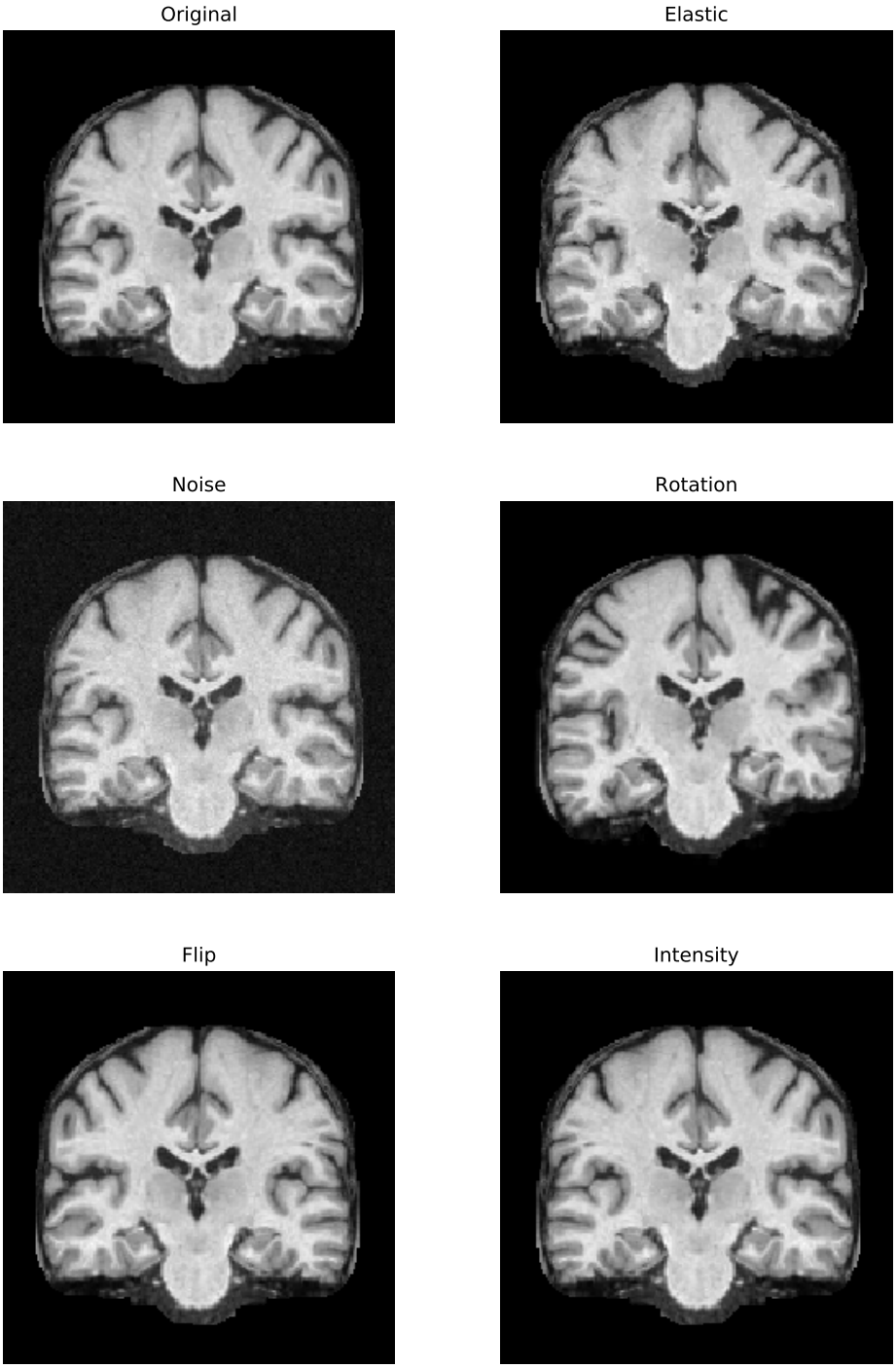


Figure 3.10: Examples of applying data augmentation to a sample brain.

4
Models

This section exposes the reader to the different model architectures used during the project. We decided not to go for a complex deep learning algorithm such as ResNet[17] for simplicity but it would be interesting to try them with more data to see the impact.

4.1 Standard 3D CNN

The model we choose for classification is composed of 12 convolutional layers each of size 3 by 3 by 3. The number of output channels gradually increases from 1 (grey-scale) to 32. As we are dealing with 3D images, it is not scalable to increase the number of channels without reducing the image size as it would require a lot of GPU memory. Therefore, we added a MaxPooling layer after every two convolutional layers. This technique is used to reduce the image size, but this is not the only way of reducing the image size. In fact, when performing standard convolution with a kernel of size k.
The output shape in each dimension will be of $s_{out} = \frac{s_{in}-k+2*pad}{stride} + 1$, where k is the kernel size, pad the added padding and s_{in} the input size. In this work, we focused on building a model that could explain its prediction as much as possible. During the project, we have realized that it was important to keep the shape of the image unchanged after each convolution layer, otherwise, we got some alignment issue between the image and the saliency map once interpolated to the image dimensions. The image is therefore padded with zeros all around in order to prevent it from these artifacts.
Between every layer, we choose to activate the layer output with the classical Relu[18] function. The main reason behind this choice is the need for easy interpretability with the FullGrad algorithm (the current implementation only supports the Relu activation function). The convolutional layers are good for extracting relevant features for the task. These features are then fed into a classifier composed of two linear layers. A dropout[19] layer has been added in between these layers in order to reduce overfitting. The architecture is summarized in figure 4.1.

4.2 3D CNN With Global Average Pooling

The model described above has quite a lot of parameters to learn especially due to the last linear layers. In fact, when flattening the reduced image it becomes a high-dimmensinal vector. This has the disadvantages of being harder to train and demands a larger amount of training data. As already stated at the time of writing we do not have a lot of data to train our model.
To overcome this, we slightly changed the design of our network and replaced the flattening layer by a *Global Average Pooling Layer*[20] (GAP). This change dramatically reduced the number of parameters of the classifier which allowed us to increase the number of output channels of our features extractor to 128.
This technique is often performed in computer vision and has the advantage of allowing the network to infer on data of almost any size. By construction, it gives the network a translation invariance property in the opposite of being only equivariant to translation for standard 3D CNN.
With regards to our preprocessing this might, in fact, be a feature not well suited, as all the brain scans

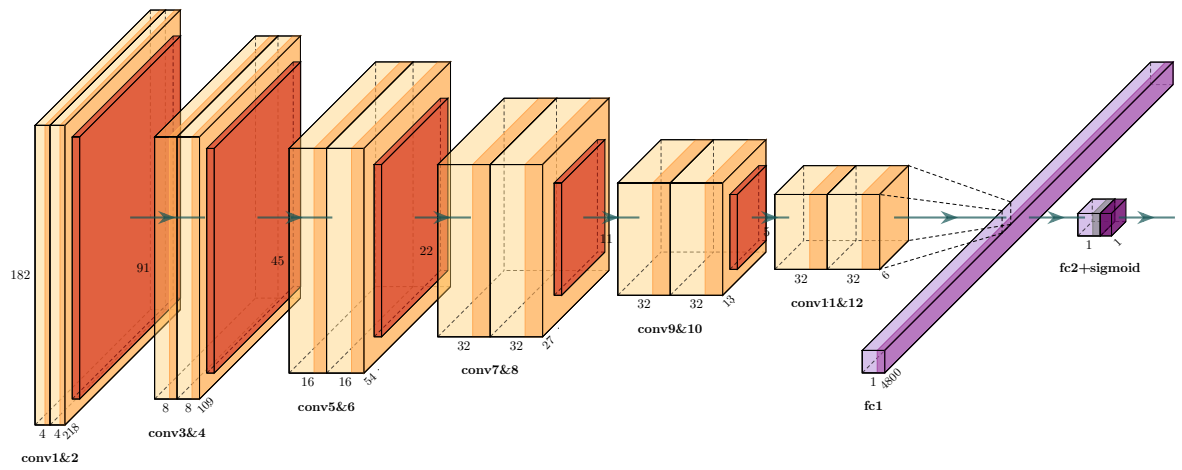


Figure 4.1: Architecture of the implemented network. For visual purpose the MRI image of shape (182, 218, 182) is represented as an image of shape (182, 218).

are registered into the same space meaning that a specific voxel is going to encode a specific location inside the brain. This global average pooling has, in fact, the disadvantage of losing some spatial location property.

4.3 Unbiased Model

Often the data has some underlying bias. For example, in the case of Alzheimer's disease, it appears that old people are more likely to be diagnosed with the disease than young ones (see figure 3.5). Experiments on age prediction in annex B, show that age is a hidden feature well present in MRI scan.

Such bias can, of course, fool the model, that might be overconfident on classifying old people as having dementia or even worse, young people as healthy just because in the training set demented samples were mostly old people. Note that by building the model below we are indeed only interested in mitigating the bias due to age, but this model can be easily modified in order to mitigate other potential biases as long as the label is provided for the bias. Examples of other bias might be sex, left- or right-handed, or even the hospital/machine used for scanning.

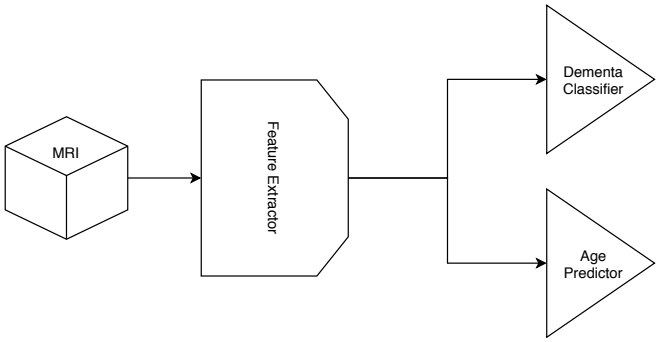


Figure 4.2: Unbiased Model Schema.

As illustrated by figure 4.2, the model is composed of 3 elements. The *feature extractor* extract a fixed number of features from the MRI using convolutional layers in a similar manner as the other models described in chapter 4. The *dementia classifier* and *age predictor* are both composed of fully connected layer, with one feature as output. The dementia classifier has a sigmoid function on top of it as its task is a binary classification while the age predictor does not have an activation function on its output.

What makes this model interesting is that we had to use adversarial training. For this, we need two optimizers, one that works on the weights of the feature extractor and the weights of the age classifier. While the other works on the weights of the age predictor only. In addition to the classic dementia loss (BCE) and the age predictor one (MSE), we define a new loss L_3 to be: $L_3 = L_{dementia} - \lambda L_{age}$. The

final goal is a Min-Max game where the first optimizer tries to minimize the L_3 loss while the second one tries to minimize the L_{age} one. This model is inspired by the GAN [21] architecture and requires a lot of fine-tuning to get stable training.

4.4 Autoencoder for Transfer Learning

Autoencoders are a class of neural networks, usually used to encode the data into a latent space. As illustrated in figure 4.3, the network is made of two components, an *encoder*, which maps the original data to the latent space and a *generator* which maps back from the latent space to the original space of the data. Note that the latent space must be of lower cardinality than the input, otherwise the encoder could simply learn the identity function. The training goal is to reconstruct the input data as accurately as possible. To ensure this, the loss to use can be either the mean square error with no activation to the last layer. Or in the case where the input data has been normalized into the range $[0, 1]$, by applying a *sigmoid* function to the last layer. Binary cross-entropy loss can be used as an alternative, note that in this case the optimal loss is not expected to be zero.

The loss is obtained by comparing the input data with the network output, thus making Autoencoders fall into the category of models that can be trained in an unsupervised manner. This is especially interesting for problems where a lot of data of data is provided but only a few of them are labeled. Once trained on unlabeled data, the weights of the encoder block can be reused in combination with another network for a classification task on similar data.

Another similar network is the denoising Autoencoder [22]. It is trained to remove artificially added noise from its inputs. It has the advantage of not requiring compression of the input in dimensions and is also known to extract better features.

This model has been implemented, but as we did not get many unlabeled data, we did not see any improvement when using it.

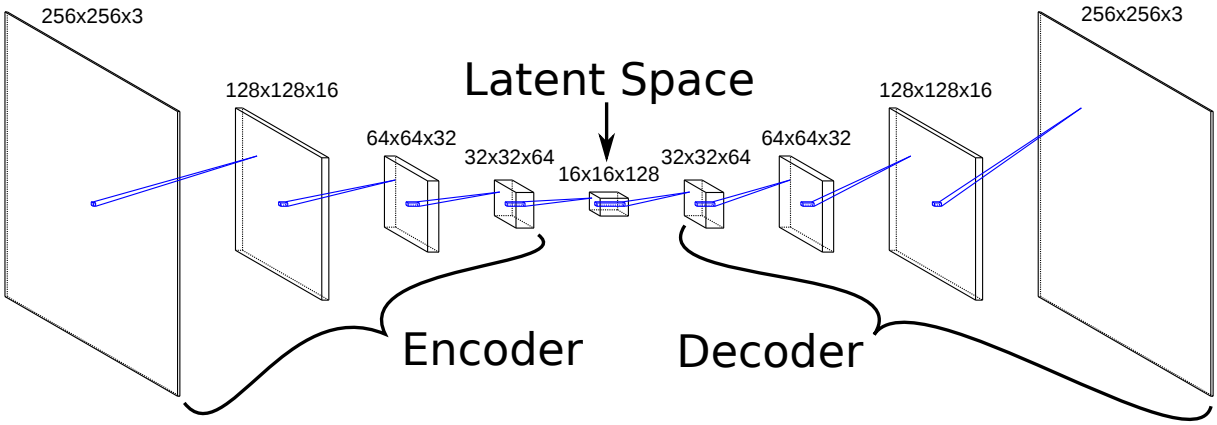


Figure 4.3: AutoEncoder architecture[16] for images.

[16] https://awesomeopensource.com/project/yu4u/convnet-drawer?categoryPage=11

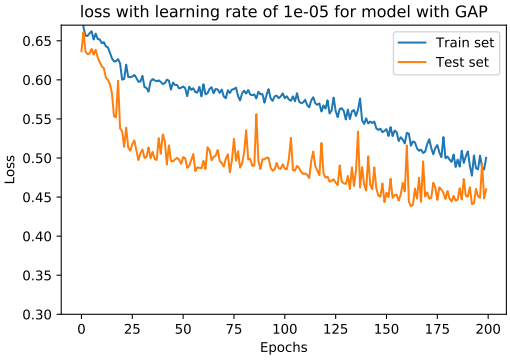
5 Experiments

This chapter explains how the models were trained in order to obtain the best performances and an explanation map.

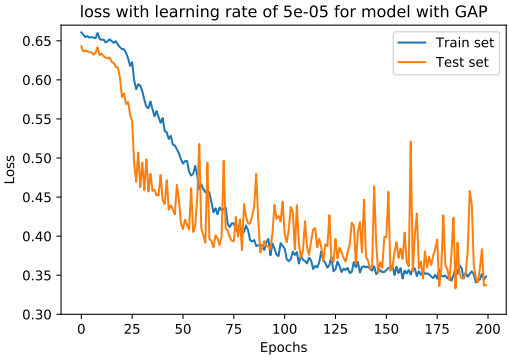
5.1 Training

Training of the models was done using the Adam[23] optimizer and the pytorch[24] library. An implementation in tensorflow[25] has been done but PyTorch was chosen as we needed to explain the model using the fullGrad[13] GitHub repository. For the loss we chose Binary Cross Entropy as the task we are solving is a binary classification.

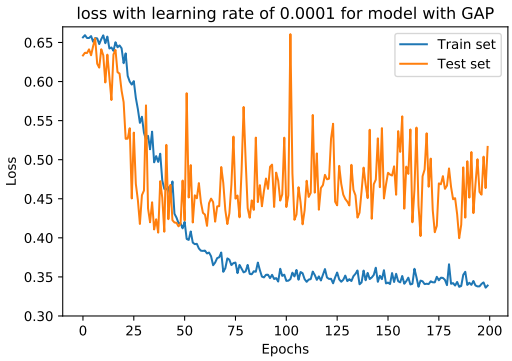
Training the different models required to fine turn some hyperparameter. Figure 5.1 shows how the test loss can be used to compare the performances of the model when trained with different learning rates. Here we see that choosing a learning rate of 5e-05 seems to be the best decision to train the model.



(a) lr=1e-05



(b) lr=5e-05



(c) lr=0.0001

(d) lr=0.0002

Figure 5.1: Comparison of the train and test loss for different learning rate across 200 epochs of training.

Once trained, a network should be able to predict the correct class for any input. Unfortunately, the prediction of a model cannot be blindly trust. Nonetheless, the output of the network can be interpreted

as a confidence score on the prediction of the network. Therefore even in the case where the network outputs 0.45, which means that the patient does not have dementia, it can still be worth double-checking his case. In fact, for this particular prediction, the network is not so confident and has more chances of being wrong. Unfortunately, when trained for too long, the network tends to become overconfident and its outputs become almost binary (values close to zero or close to one), losing part of its interpretability. Label Smoothing [26] aims at reducing the overconfidence of the network by changing the label of the train set from 0 to $0 + \epsilon$ and from 1 to $1 - \epsilon$. For the experiments, we chose ϵ to be 0.1.
The loss used for training is Binary Cross Entropy. But as shown in figure 3.4 the data we are training on is unbalanced. Instead of training the model with the standard binary cross-entropy loss as defined in section 2.3.1, we used a variant that penalizes more the mistakes made on the underrepresented class. In our case, as we have approximately 70 percent of control samples and 30 percent of samples from people with dementia, the loss for a demented sample would be weighted by $\frac{1}{0.3} = 3.33$ and the other by $\frac{1}{0.7} = 1.43$.
Note that more could have been done in order to better train the model such as trying another optimizer or even using a learning rate scheduler, but we preferred to focus on explainability instead of performances.

5.2 Evaluation

The performance of the different model being very similar, we will perform the evaluation on the standard convolution model described in section 4.1. This section purely evaluates the model's performance and resistance to age bias but is not interested in the explainability of the model. For evaluation of the age predictor please refer to annex B.

5.2.1 Bias due to age

As shown in the section 3.2.2, the OASIS dataset presents a bias to age. To evaluate the model bias we can look at figure 5.2. We can see a trend that the model tends to more easily predict dementia to an old person than a young one.

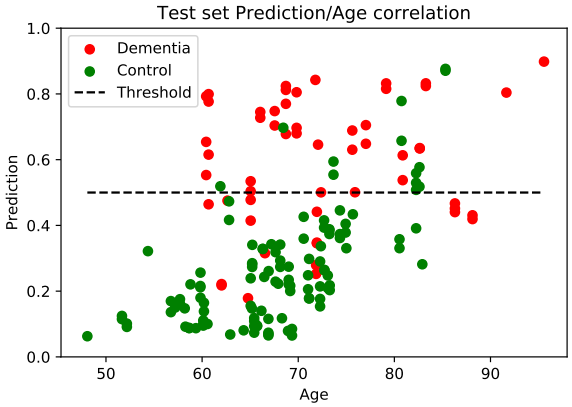
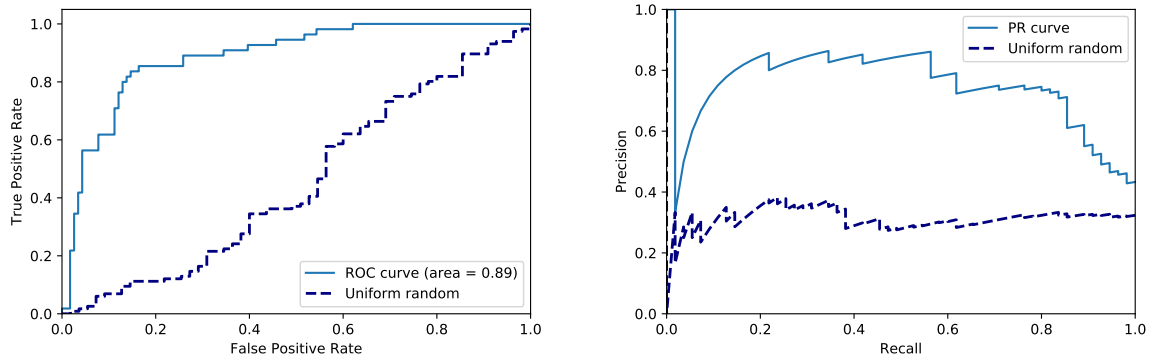


Figure 5.2: Prediction of dementia per age.

5.2.2 Metrics

In order to evaluate and compare the different model, we used the metrics defined in section 2.3. The confusion matrix in figure 5.3 allows us to compute the accuracy which is of 81.87%. This metric is quite confusing as it seems to be a good performance, but random guessing already has an accuracy of 70.76%. Instead, we can compute the precision which is of 74% and the recall to be of 67.27%.
Figure 5.4 shows the ROC and PR curves to gain a better sense of the model performance. Note that as the dataset is unbalanced it is recommended to look at the PR curve.



(a) ROC curve.

(b) Precision and Recall curve.

Figure 5.4: Evaluation curves.

5.3 Model Output Explanation

As shown in figure 2.1, Alzheimer disease tends to severely damage some specific regions of the brain. One of these regions is the hippocampus which serves at the creation of new memories. We would expect from a good explainer that indeed it highlights this specific region when explaining the output for a damaged brain. Figure B.2 compares the outputs of the 3 algorithms we implemented on different slices of a damaged brain. In fact, in figure 5.5 we chose slices where the hippocampus is visible and observed that while Shap explanation is difficult to interpret, both GradCam and FullGrad focus on the hippocampus. It is especially visible with the output of the fullgrad algorithm where its maximal attention on this slice is actually located inside the hippocampus. For the rest of the report, we chose to work with fullgrad as its output makes more sense to us and is also sharper. The brain viewer implemented in annex A is designed to work with a fullgrad saliency map.

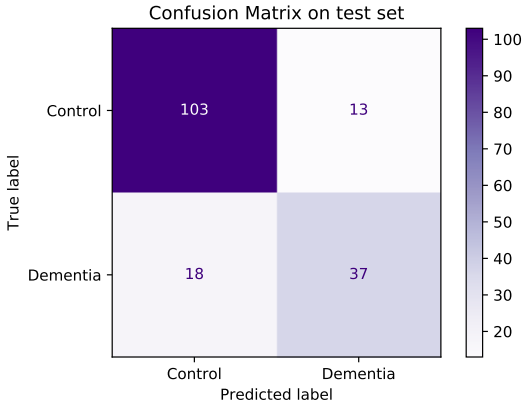


Figure 5.3: Confusion Matrix on the test set.

Looking at the fullgrad output in figure 5.6, we realized that the model tends to focus more on the right hippocampus that the left one. In fact, the analysis performed in annex C.2 seems to confirm the difference with respect to the right hippocampus between healthy and demented patients.

Further analysis of the fullgrad outputs show the focus that the model has on the ventricles that typically grows larger for damaged brains.

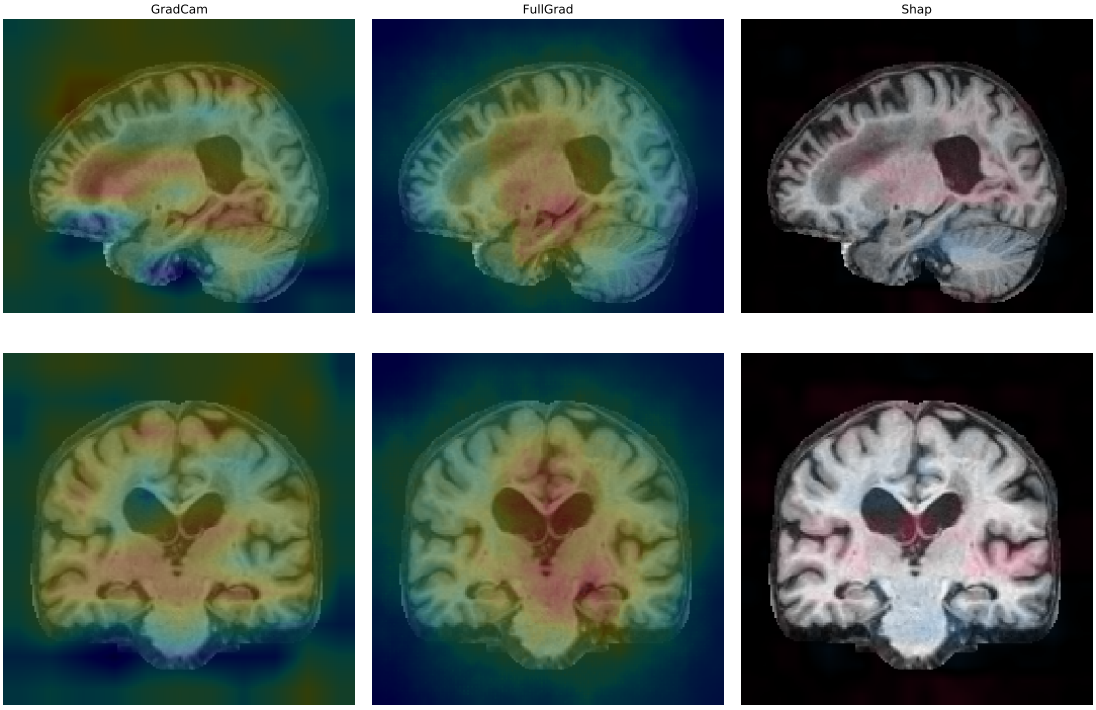


Figure 5.5: Outputs of the explainer algorithm on one patient with dementia.

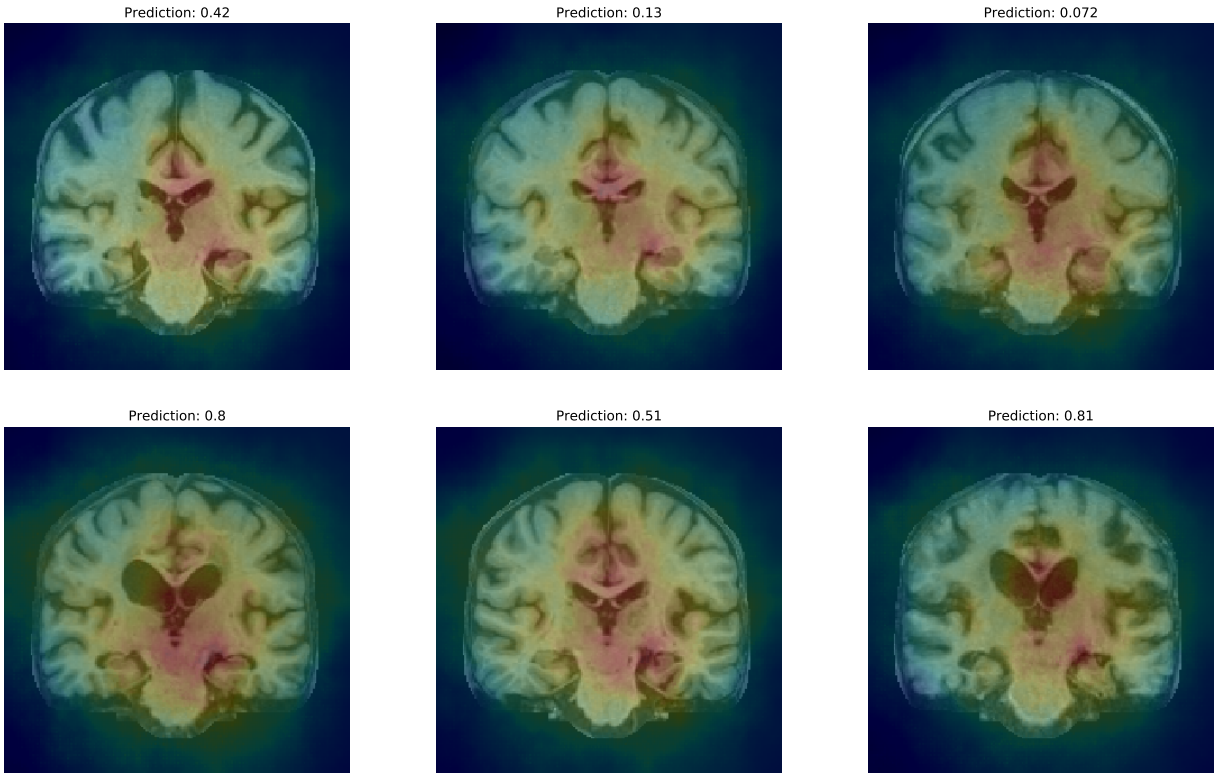


Figure 5.6: Outputs of the fullgrad algorithm,the first row is composed of control patients and the second one of dementia patients.

6
Concluding Remarks

6.1 Conclusion

In this study, we built a complete reusable pipeline composed of preprocessing, training, evaluation and explanation to detect dementia from raw MRI scans. The models obtained by training on the OASIS dataset did not attain state-of-the-art performances but have the advantage of providing not only a diagnostic but an explanation about which region of the MRI made the model do such a prediction. The model explanation we obtained coincides with the consequences of dementia that specialists observe. In particular, fullgrad was able to spot the importance of the hippocampus in the input image when predicting dementia.

In addition to that, we built a responsive and easy-to-use web application that allows any clinician without any machine learning background to quickly gain insight into the prediction our model made. Thus reducing the black box image from which machine learning is suffering from in the field.

In conclusion, we show with this thesis that deep learning techniques have reached a sufficient level technical maturity to provide interpretable predictions that can be integrated into research activities and soon into clinician practices.

6.2 Future Works

Despite the already interesting results and helpful insight on dementia gained by the visualization, this project has been conducted using only a small number of labeled data due to the difficulties in accessing a custom dataset. To have even more interesting results, the entire pipeline would have to be applied to a much larger dataset for which the labels would be provided.

We would also like to try the pipeline with some other modalities instead of using the T1 only. For example, it has been shown that the iron density inside a brain is correlated with dementia[17]. In fact, we think that the cause of dementia might be invisible through a T1 weighted image and that we might currently only be looking at the consequences.

In section 3.2.2 we decided to label all the scans of a patient as having dementia if his latest check-up he was diagnosed with dementia. In a future work, we would like to analyze the output of the model on his early scans and check if the model is able to detect something.

During the period of the project we had the opportunity to show our results to one clinician. A next step would be to confront them to more clinician to hear from them and improve even more the pipeline according to their needs.

Section 5.3 highlights intriguing results about the right hippocampus, further analysis should be perform to better understand the reason behind this higher attention on the right compare to the left side.

[17] https://www.medicalnewstoday.com/articles/measuring-iron-in-the-brain-can-point-to-dementia

Bibliography

[1] J. Ashburner and K. J. Friston, "Voxel-based morphometry—the methods," *NeuroImage*, vol. 11, no. 6, pp. 805 – 821, 2000.

[2] J. Samper-González, N. Burgos, S. Bottani, S. Fontanella, P. Lu, A. Marcoux, A. Routier, J. Guillon, M. Bacci, J. Wen, A. Bertrand, H. Bertin, M.-O. Habert, S. Durrleman, T. Evgeniou, and O. Colliot, "Reproducible evaluation of classification methods in alzheimer's disease: Framework and application to mri and pet data," *NeuroImage*, vol. 183, pp. 504 – 521, 2018.

[3] C.-w. Hsu, C.-c. Chang, and C.-J. Lin, "A practical guide to support vector classification chih-wei hsu, chih-chung chang, and chih-jen lin," 11 2003.

[4] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, p. 5–32, Oct. 2001.

[5] S. D. Bondi MW1, Edmonds EC1, "Alzheimer's disease: Past, present, and future," *Journal of the International Neuropsychological Society*, 10 2017.

[6] K. Franke and C. Gaser, "Ten years of brainage as a neuroimaging biomarker of brain aging: What insights have we gained?," *Frontiers in Neurology*, vol. 10, p. 789, 2019.

[7] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, p. 359–366, July 1989.

[8] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object recognition with gradient-based learning," in *Shape, Contour and Grouping in Computer Vision*, (Berlin, Heidelberg), p. 319, Springer-Verlag, 1999.

[9] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," 2017.

[10] B. Zhou, A. Khosla, L. A., A. Oliva, and A. Torralba, "Learning Deep Features for Discriminative Localization.," *CVPR*, 2016.

[11] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, p. 336–359, Oct 2019.

[12] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," 2014.

[13] S. Srinivas and F. Fleuret, "Full-gradient representation for neural network visualization," 2019.

[14] P. J. LaMontagne, T. L. Benzinger, J. C. Morris, S. Keefe, R. Hornbeck, C. Xiong, E. Grant, J. Hassenstab, K. Moulder, A. Vlassenko, M. E. Raichle, C. Cruchaga, and D. Marcus, "Oasis-3: Longitudinal neuroimaging, clinical, and cognitive dataset for normal aging and alzheimer disease," *medRxiv*, 2019.

[15] A. Y. K. M. O. R. F. L. R. S. Beason-Held LL, Goh JO, "Changes in brain function occur years before the onset of cognitive impairment.," 2013.

[16] N. J. e. a. Tustison, "N4itk: improved n3 bias correction.," *IEEE transactions on medical imaging vol. 29,6 (2010): 1310-20.*

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[18] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, (Madison, WI, USA), p. 807–814, Omnipress, 2010.

[19] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, p. 1929–1958, Jan. 2014.

[20] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013.

[21] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[22] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, p. 3371–3408, Dec. 2010.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.

[24] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[25] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," 2015.

[26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," 2015.

A

Brain Viewer Tool

The final goal of explaining a model is to help a specialist to make a better diagnosis. Therefore it's important to provide him with a tool that is as easy as possible to use. For this we build a web app in react[18] where a clinician can upload the brain he is interested in checking, and the attention map we computed in order to see them and navigate into them exactly the same way he already knows how to do using other visualization software such as SPM[19]. In addition, he is also provided with information that tells him how confident the model is at predicting dementia and the ground truth label (see figure A.2a). The control panel A.2b enables a richer experience notably through raising the pixel intensity of the saliency map to the power of its slicer value. Therefore highlighting more the zone of the image where the model has more attention. Last but not least, the checkboxes allow the user to hide/display the brain or the saliency map for better visual control. An intuition of the capability of this tool can be sensed through the screenshot in figure A.1. Or it can be tried online by following the instruction on the GitHub repository https://github.com/cgallay/BrainViewer.

[18] https://reactjs.org/
[19] https://www.fil.ion.ucl.ac.uk/spm/software/

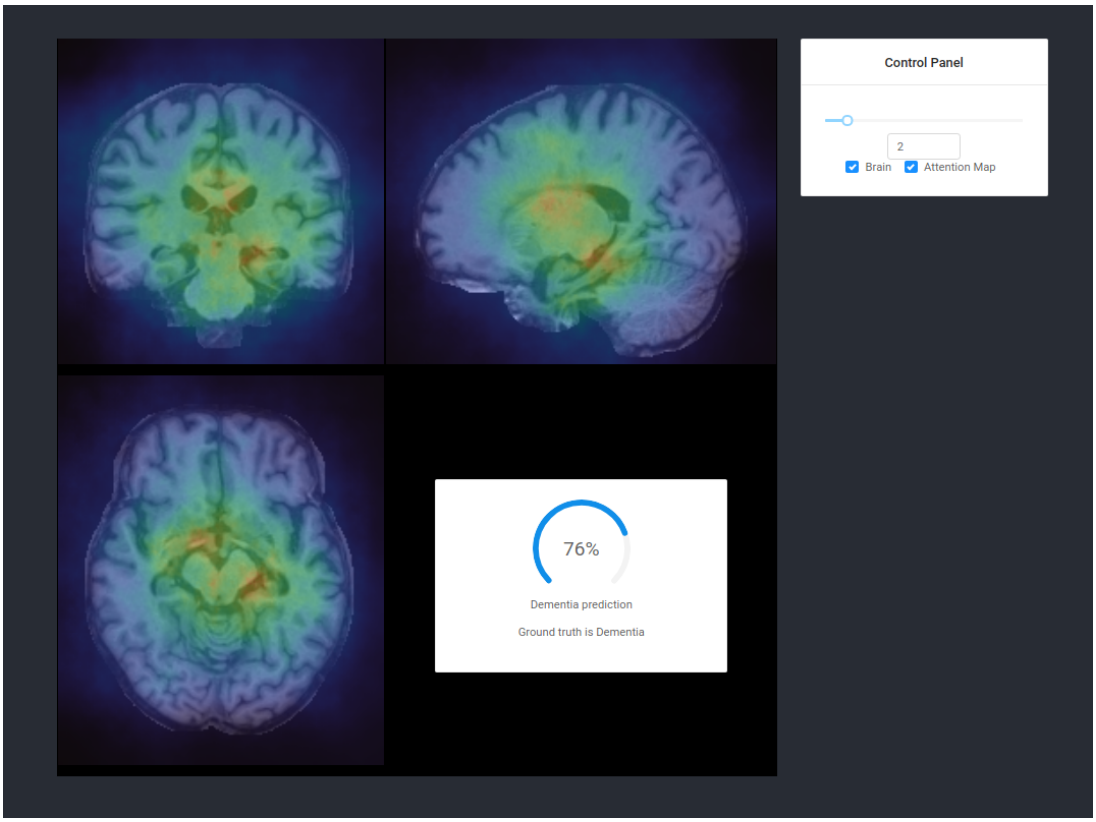
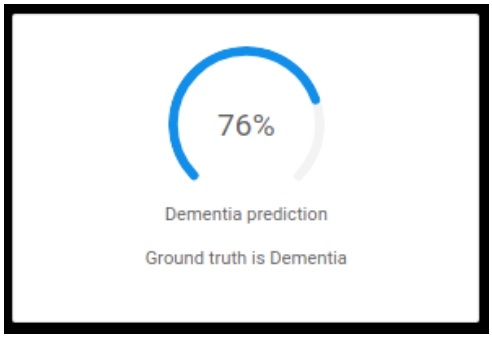
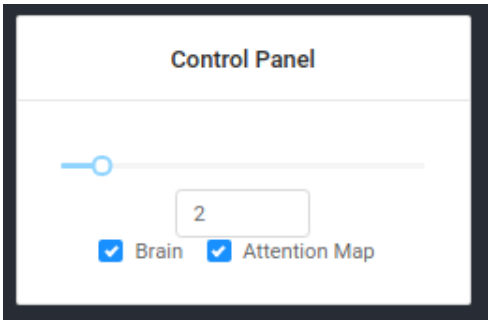


Figure A.1: Screenshot of the viewer.



(a) Prediction output.



(b) Control Panel.

Figure A.2: Extra element of the BrainViewer tool.

B

Age predictor

B.1 Training model

At the very beginning of the project, we had no access to MRI of people with dementia. In order to proceed anyway and gain some time, we choose to work on predicting the age of someone based only on an MRI scan of his brain. For this purpose, the public IXI dataset was used. From there we design our preprocessing pipeline, and model that we were able to reuse later for dementia detection simply by changing the training dataset. We present here the result we obtained for this experience.

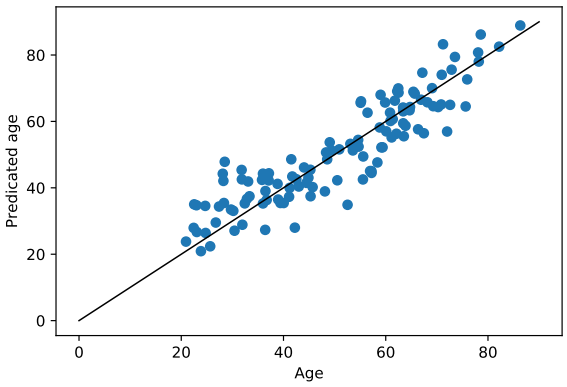


Figure B.1: Prediction of age versus real age on IXI test set.

The only variation compared to the dementia detection pipeline explained earlier is the last layer of the model we used and the loss to optimize. Instead of a *Sigmoid* function, the last layer for a regression simply does not contain any activation function. The loss we want to minimize is a mean square error, which is classic for regression as we want to penalize larger errors. To evaluate the performance, the Mean Absolute Error (MAE) was computed, as it is more meaningful and easier for a human to imagine.

We achieved an MSE of 45 which correspond to a MAE of roughly 5 years. Interestingly by looking at the correlation between our prediction error and the age of the patient in figure B.1, we found that the performance is uniform across ages which is a nice property of our predictor.

B.2 Dementia Seen as Over Ageing

In this section, we tried to check whether or not a patient with dementia can be seen as an overaged person. By looking at figure B.2 it is hard to say that indeed our age predictor trained on healthy brains is overageing dementia patients. Indeed we might see that the model has worst performances on damage brain but once again we do not have enough data to conclude anything with certainty.

Figure B.2 does indeed highlight age bias present in the dataset.

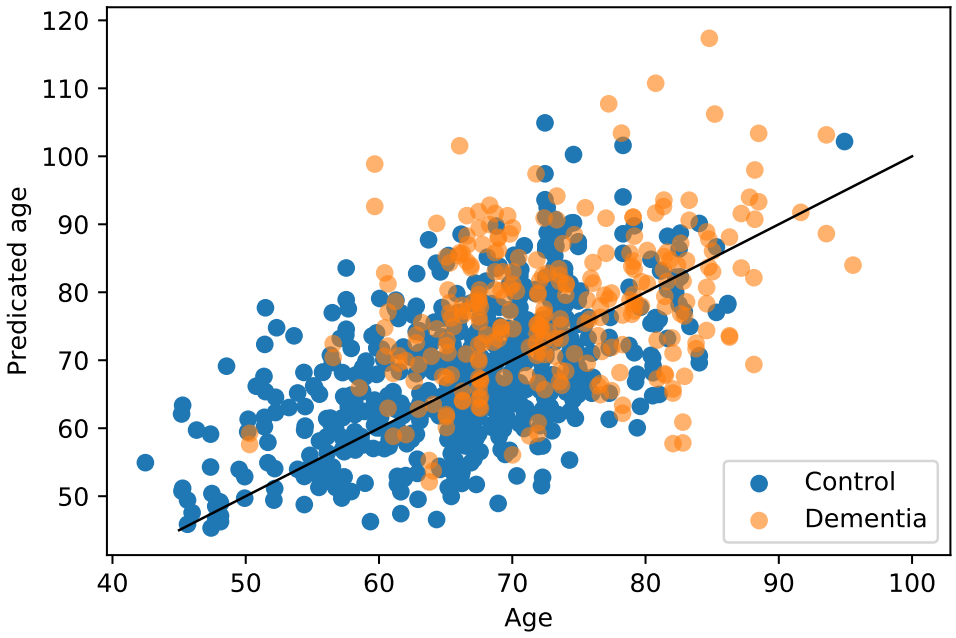


Figure B.2: Prediction of age for control and dementia patient on the OASIS dataset.

C

Morphometry Analysis

C.1 Basis Analysis

In this chapter, we performed a mass univariate Voxel based morphometry to compare demented patients versus healthy controls using the SPM Matlab toolbox. The data were all prepropoceed using SPM12 segmentation and aligned to the common MNI space using SPM12 Dartel. In SPM we first compared the patients and the controls according to the clinical diagnostic, the model include age and Total Intracranial Volume as confounds. The figure C.1 shows the T-statistical map corrected for multiple comparison (FWE $p < 0.05$). It highlights in yellow the brain regions where the grey matter volume of the patients is significantly lower than the grey matter volume of the controls. As expected, the results show that most of the differences are located in the Medial Temporal Lobe in the left and right hemispheres, the regions which include the hippocampus.

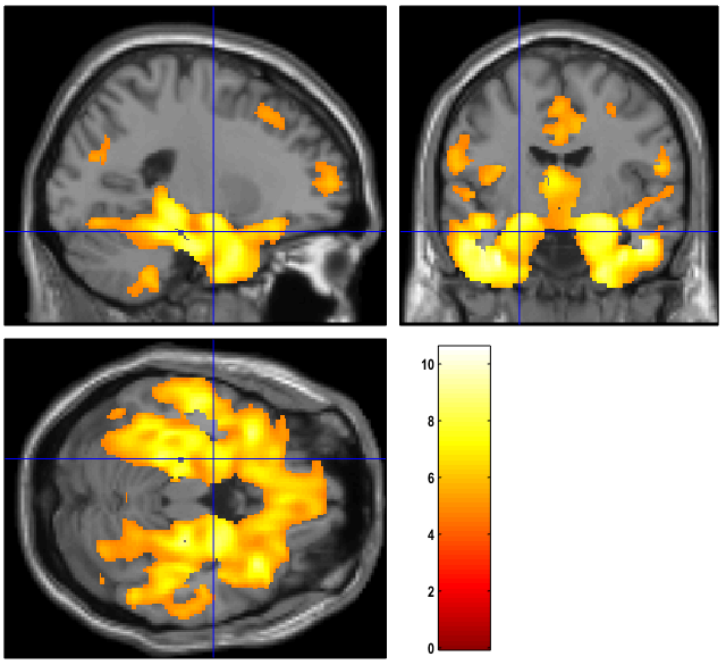


Figure C.1: A Tmap overlay that shows the brain atrophy pattern associated with the dementia disease at the population level.

C.2 Deep Learning Results Analysis

In this section, we wanted to further analyse the patients that our deep learning model detected as having dementia when the ground truth was indeed healthy. From figure C.2 shows the differences between healthy patients predicted as healthy and the healthy one predicted as AD. In general we would expect to see no difference, but there is a difference which is not detected by the clinicians. As we already highlighted in section 5.3, the right hippocampus seems to have a bigger importance than the left one

for dementia prediction.

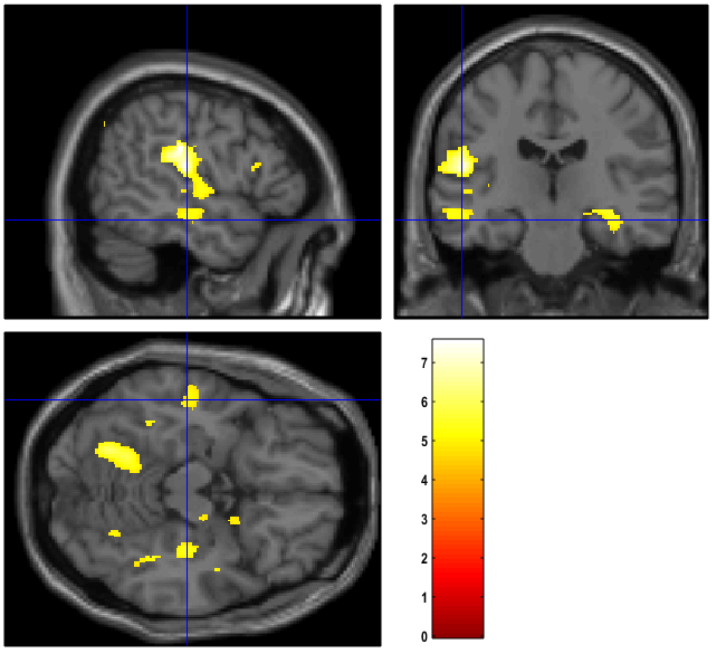


Figure C.2: Healthy patients predicted as AD by our model vs the healthy patients predicted as healthy.