Title
Convolutional Neural Networks based on Brain MRI for Alzheimer’s Disease Detection

Author
Civico Dorado, Noelia

Publication Date
2022

UNIVERSITY OF CALIFORNIA,
IRVINE

Convolutional Neural Networks based on Brain MRI for Alzheimer's Disease Detection

THESIS

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Computer Engineering

by

Noelia Civico Dorado

Thesis Committee:
Professor Nader Bagherzadeh, Chair
Professor Jean-Luc Gaudiot
Professor Chen-Yu (Philip) Sheu

2022

DEDICATION

To my parents and sister

in recognition of their love, help and support

TABLE OF CONTENTS

Page

LIST OF ABBREVIATIONS

AD	Alzheimer’s Disease
NC	Normal Cognitive
MCI	Mild Cognitive Impairment
CAD	Computer-Aided Diagnosis
MRI	Magnetic Resonance Imaging
PET	Position Emission Tomography
CT	Computed Tomography
SPECT	Single-Photon Emission Computed Tomography
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
NN	Neural Network
ANN	Artificial Neural Network
DNN	Deep Neural Network
CNN	Convolutional Neural Network
WHO	World Health Organization
FC	Fully-Connected
ReLU	Rectified Linear Unit
SVM	Support Vector Machines

LIST OF FIGURES

Page

LIST OF TABLES

Page

ACKNOWLEDGEMENTS

First of all, I would like to deeply thank my supervisor, Professor Nader Bagherzadeh, for giving me the opportunity of being part of his lab. Thank you for your valuable advice and guidance, as well as for the chance to broaden my knowledge.

Moreover, I would also like to thank Professor Roger Rangel for giving me the opportunity of coming to the University of California, Irvine with the Balsells Fellowship.

Last but not least, I would like to express my deepest gratitude to my family and friends, especially to my parents and sister. Thank you, mom and dad, for your unconditional support and for always encouraging me to pursue my dreams. And thank you Marina for our countless laughs and confidences.

ABSTRACT OF THE THESIS

Convolutional Neural Networks based on Brain MRI for Alzheimer's Disease Detection

by

Noelia Civico Dorado

Master of Science in Computer Engineering

University of California, Irvine, 2022

Professor Nader Bagherzadeh, Chair

In recent years, the problem of detecting Alzheimer's disease with computer-aided diagnosis systems has become a relevant research field for medical diagnosis efficiency and image classification innovation. Alzheimer's disease is an irreversible progressive neurogenerative disorder caused by damage to nerve cells, which leads to memory loss and other cognitive functioning skills deterioration. Detecting Alzheimer's in its preliminary stages is crucial to planning treatment and therefore delaying the progression of the disease. Magnetic Resonance Imaging scans can capture complex changes in the brain and assess the damage caused by the disease. The growing popularity, accuracy, and applicability of Convolutional Neural Networks make them an optimal solution to perform this medical task.

This study implements and compares several deep models and configurations, including both two-dimensional and three-dimensional convolutional neural networks. The results classifying normal cognitive subjects versus Alzheimer's patients show a good performance, especially for AlexNet, ResNet18, and ResNet34 architectures. In particular, 3D models are able to achieve better accuracy outcomes.

Chapter 1:

Introduction

1.1. Alzheimer's disease

Alzheimer's Disease (AD) is an irreversible progressive neurogenerative disorder caused by damage to nerve cells (neurons). AD is the most common cause of dementia among older adults. Most of the individuals diagnosed with AD are 65 or older. Studies indicate that people aged 65 and older survive an average of four to eight years after a diagnosis of AD, although some live as long as 20 years [1]. The World Health Organization (WHO) reported that over 55 million people worldwide live with AD in 2021 and was the 7th leading cause of death [2][3]. Statistics show that this number is expected to rise drastically in the following decades.

In AD, the deterioration of brain cells starts gradually affecting cognitive functioning -thinking, remembering, and reasoning- and therefore the brain parts responsible for memory, language, and learning. Common symptoms include memory loss, reduced ability to reason and think, difficulty with problem-solving skills, forgetfulness and deep confusion regarding time, events and places, doubts about family members and friends, and mood or behavioral changes. AD is a progressive disease, which implies that it gets worse with time. As the disease spreads to further parts of the brain and more neurons are damaged, symptoms also get worse: difficulty completing simple daily tasks, walking, swallowing, speaking, reading, or writing [4].

AD begins with mild deterioration and gets progressively worse. The disease has three major stages: mild, moderate, and severe. Nevertheless, there exists a previous phase called Mild Cognitive Impairment (MCI). MCI is a transitional state between healthy patients and AD patients. Individuals with MCI due to AD present memory problems that do not prevent them to continue with their daily routine. At this stage, they also have a biomarker evidence of Alzheimer's brain changes [1][5].

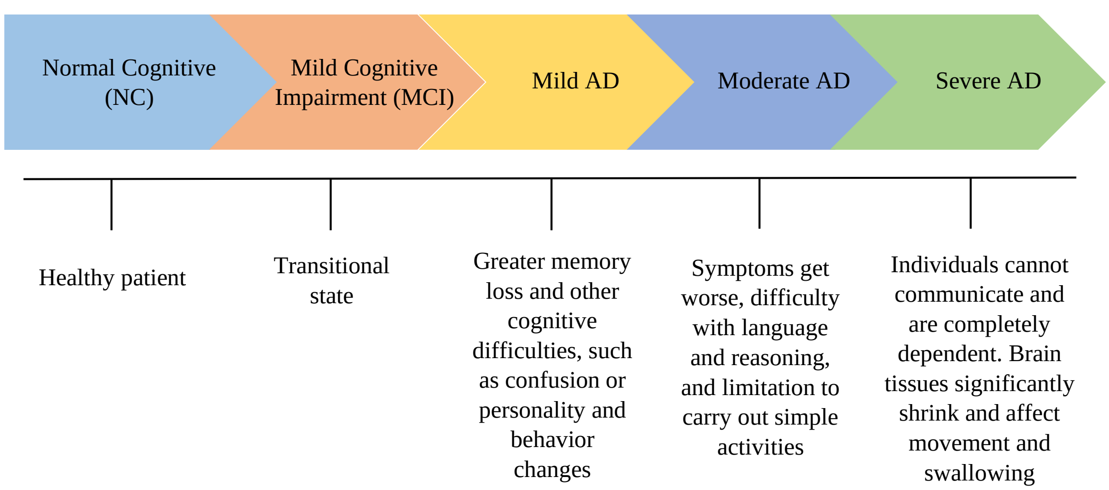

Figure 1: *AD stages*

Since AD is incurable, the early diagnosis is essential to postpone the progression of brain irreversible damage and extend the independence of patients for a longer period. In this case, direct observation of the signs is required to detect and evaluate AD through a comprehensive evaluation -memory, problem solving, attention, counting, and language tests- as well as brain scans [5].

1.2. Brain imaging in AD

Brain changes associated with AD may begin a decade or more before symptoms appear [5]. Beta-amyloid plaques and tau tangles appear in the brain and lead to the damage and destruction of neurons [1]. This damage is reflected in the hippocampus area and the cerebral cortex of the brain, which begin to shrink [6]. These affected regions are responsible of memory, thinking, planning and judgement [7]. As more neurons die, other parts of the brain continue shrinking until brain tissue is significantly affected.

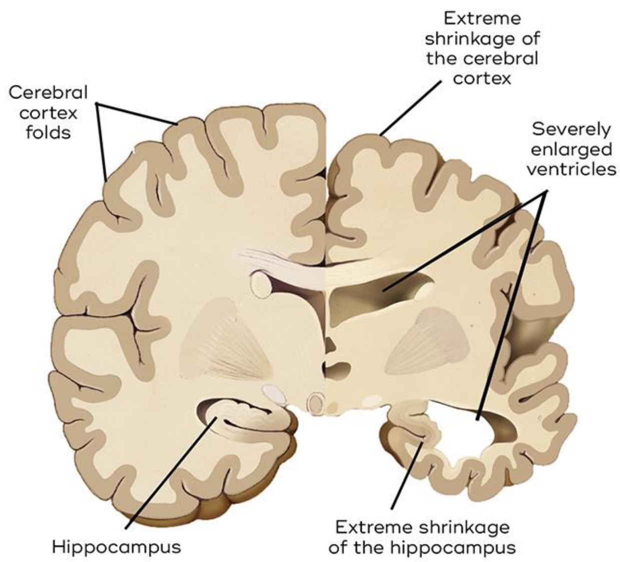

Figure 2: *Healthy brain vs. Severe AD brain* [8]

Medical imaging are techniques used to visualize representations of the inner parts of the human body for a medical purpose. In this case, neuroimaging tools together with Artificial Intelligence

(AI) and preprocessing methodologies can help doctors to detect whether a patient is developing AD and therefore help planning the treatment.

Common advanced brain imaging techniques in AD are Computed Tomography (CT), Magnetic Resonance Imaging (MRI), Single-Photon Emission Computed Tomography (SPECT), or Position Emission Tomography (PET).

In this thesis, MRI is the brain scan selected to identify and classify AD.

1.3. Motivation

As described before, AD is irreversible and has no cure. Nevertheless, the early diagnosis of AD plays a very significant role in planning a treatment and therefore delaying the progression of the disease.

In order to diagnose AD, medical experts manually inspect the brain imaging tools. This process can be time consuming, subjective, and also requires high concentration, effort, and money. Extracting by eye the first unnoticeable features of a brain scan to identify the disease in the early stages becomes a very hard task, which makes the diagnosis prone to human errors. For these reasons, Computer-Aided Diagnosis (CAD) systems are an optimal solution to achieve fast, efficient, robust, and accurate AD diagnosis.

Developing deep learning-based CAD systems to analyze medical images and detect and classify AD can help obtaining beneficial results by providing early detection and reducing medical efforts and errors. This gain in accuracy can lead to an improvement of efficiency and quality of the AD treatment.

1.4. Thesis scope

The scope of this thesis is to deeply investigate the performance of different approaches that are based on Convolutional Neural Networks (CNN) for AD classification.

This work aims to identify the presence of AD based on MRI scans of the brain by using the OASIS dataset and analyzing different image processing techniques as well as several deep learning strategies.

This study implements and evaluates eight different CNN architectures by comparing several performance metrics, such as training and validation loss, precision, recall, or F1 score (see Chapter 4). The learning rate of different optimizers, the training time, and the number of epochs are other examined parameters that can help to investigate the behavior of CNN models.

The evaluation of CNN models includes two-dimensional (2D) and three-dimensional (3D) structures. In order to use a 2D CNN on 3D MRI volumes, each MRI scan is split into 2D slices, neglecting the connection between 2D image slices in an MRI volume. The three 2D image views of the 3D MRI volume are considered: axial view, coronal view, and sagittal view (see Figure 4).

Distinguishing between different stages of AD is not an easy task. In this case, the experiments are performed for binary classification between AD and Normal Cognitive (NC).

The impact of different data preprocessing techniques on the end performance is also studied. The result of the different preprocessed images as well as the corresponding view in 2D slices can affect the model performance.

The project outline is featured below.

- **Chapter 2:** The theoretical background of MRI, Machine Learning (ML), and especially CNNs is described in detail.
- **Chapter 3:** This chapter defines the tools, design approaches and evaluation criteria used in the project.
- **Chapter 4:** The results of the thesis are stated by using multiple parameters and performing several comparisons at different stages.

Chapter 2:

Theoretical background

2.1. Magnetic Resonance Imaging (MRI)

Magnetic Resonance Imaging (MRI) is a medical diagnosis tool used in radiology to form images of the inside of an organ or body and its physiological processes without opening it surgically. Volumetric MRI uses radio frequency and a strong and uniform magnetic field projected through a scanner to generate a detailed brain image. The scans allow the detection of possible damage and other complex changes by measuring the energy released by protons within various tissue components [9].

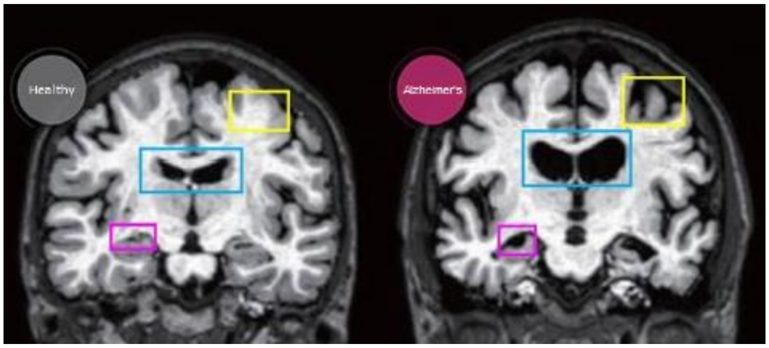

Figure 3: *MRI view of a healthy brain (left) and an Alzheimer's brain (right)*

(Yellow - Cortex, Blue - Ventricle, Purple - Hippocampus volume reduced) [10]

An MRI scan has three axis, which means that it has three different views: sagittal view, coronal view, and axial view (see Figure 4). Figure 5 shows an MRI scan obtained from the OASIS dataset, which is the one used to perform the experiments in this thesis.

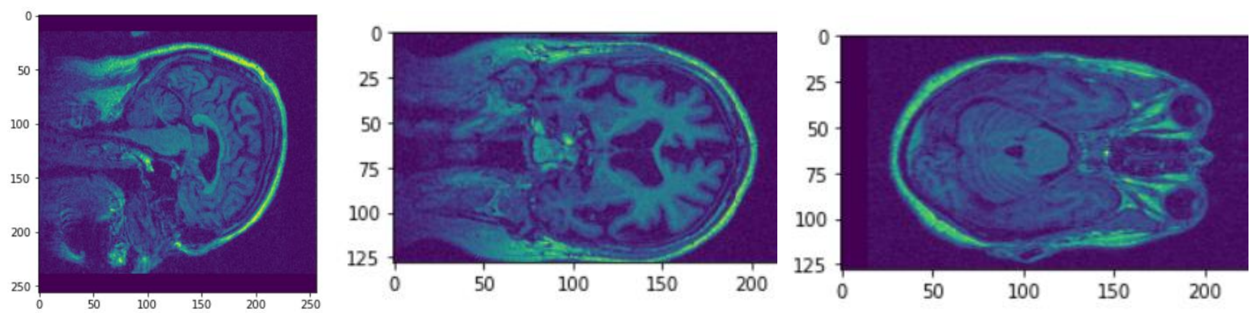

Figure 4: *MRI views: sagittal, coronal and axial (left to right)*

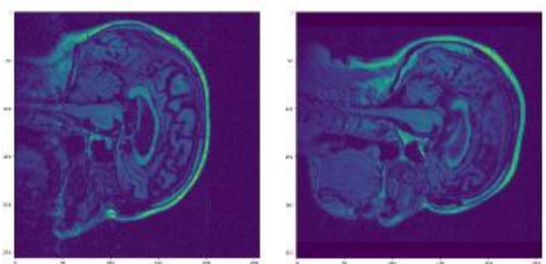

Figure 5: *OASIS dataset MRI example: NC subject vs. AD patient*

2.2. Artificial intelligence and medical imaging classification

Artificial Intelligence (AI) is the science of training machines to perform human tasks. Some decades ago, this term first appeared when scientists looked for a way of making computers capable of solving problems on their own. Intelligent machines that can emulate human behavior became a powerful and promising tool. The ability of independent reasoning and self-decision making led to the exponential growth of this new concept.

As this new scientific term started to play an important role in the engineering field, the idea of learning from actual actions to improve future results came to light, bringing up the concept of Machine Learning (ML). ML is a specific subset of AI that trains a machine on how to learn from data and make predictions. Despite its initial implication with pattern recognition, ML is nowadays employed in a wide range of applications related to computer systems, such as image classification, computer vision, object detection, language processing, speech recognition, and medicine. This groundbreaking method consists of looking for patterns and drawing conclusions without being explicitly programmed, only based on previous examples (datasets). Learning algorithms and complex models have been developed through learning from historical relationships and trends in data, generating in consequence reliable decisions and high accuracy results.

Inspired by the workings of the human brain, neural networks (also known as deep learning) are a computing system approach inside the ML field that tries to mimic how the human brain learns. It incorporates interconnected units (like neurons) that process information by responding to external inputs, transferring it between each unit multiple times to set optimal connections and parameters that can later extract conclusions from undefined data.

In general terms, image classification is to label an image or a part of an image with the corresponding classes based on its features. Artificial Neural Networks (ANN) are commonly applied in the analysis of visual data. They are an effective solution for image recognition and classification.

2.2.1. Convolutional Neural Networks

In Deep Learning (DL), a Convolutional Neural Network (CNN) is a type of neural network that consists of millions of neurons with learnable weights and biases, which are organized in several layers. CNNs are inspired by biological processes and its structure is equivalent to the connectivity pattern of neurons in the human brain. They differ from conventional neural networks because of performing convolution, which uses weight matrices (also known as filters or kernels) to produce feature maps from input data [11]. CNNs have suffered an exponential growth in terms of data computation and applicability during the last decade. Nowadays, they are used in a wide range of fields, such as image analysis, facial and speech recognition, medical diagnosis, and computer vision, in which object detection, classification, and segmentation have the strongest impact.

The preprocessing stage required in a CNN is minimum in comparison to other traditional image classification algorithms. With optimal training, this type of network has the ability of learning the values of the filters and other parameters on its own through backpropagation (see Section 3.3.3).

The architecture of a CNN is composed of an initial convolution layer, multiple hidden layers, several fully-connected layers, and a final fully-connected layer, which is called the classifier (see Figure 6). The main purpose of the first convolution layer is to extract features from the input image and drive them into the hidden layers, which are pooling layers and convolution layers partially connected. Between hidden layers, there are normally activation functions that help to keep important information for the next layers. The last fully-connected layer has a loss function, such as softmax or SVM, that allows the classification at the end. A more detailed explanation of each CNN layer is given later on in this chapter.

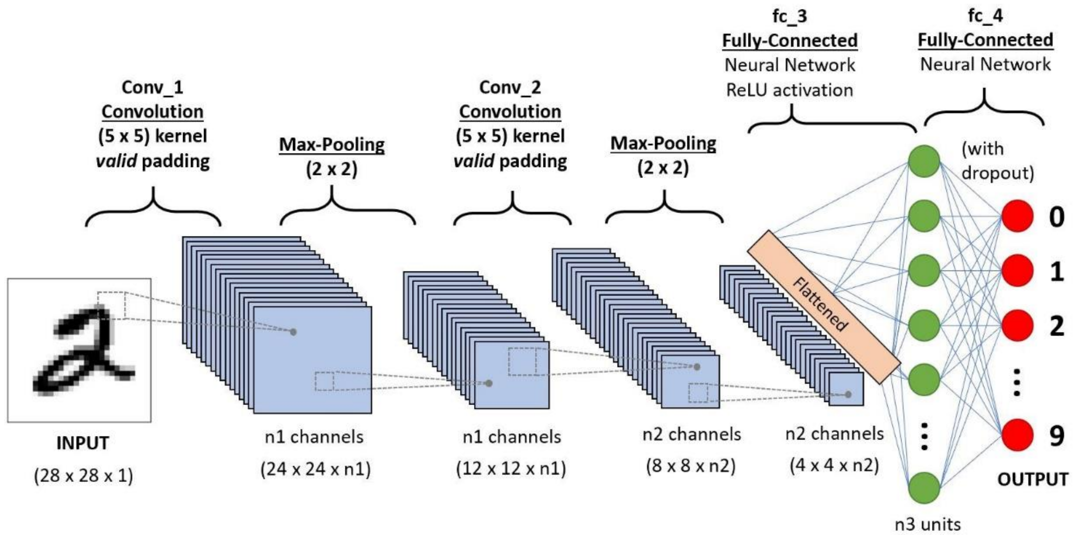

Figure 6: *Structure of a CNN with MNIST dataset* [12]

This kind of neural network is popular for two distinct attributes: sparse interactions and parameter sharing [11]. Making the kernel filter size smaller than the size of the input image allows capturing only important features and turning them into different feature maps that are driven through the different layers of the CNN. With fewer pixels of the image in consideration, sparse interactions or connectivity, as well as a reduction in parameters, is achieved. In consequence, this results in the reduction of memory footprint and computational overfitting, leading to model simplicity [4]. By capturing spatial and temporal dependencies, the model can be trained to interpret the image by extracting from low-level to high-level abstractions [13].

The structure of a CNN consists of different layers that are described in detail below.

Convolution layer

Convolutional layers are the core block of CNNs. This kind of layers apply a convolution operation across the width and height of the input image, generating feature maps as an output. For each input position, a dot product between the learnable filter and the corresponding pixels is computed.

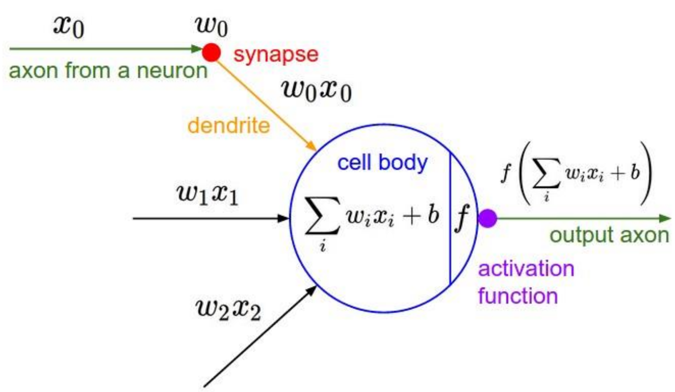

Figure 7: *Neuron model of a convolutional layer* [14]

In order to perform this computation, different hyperparameters should be considered.

- First, the **number of filters** or **depth** (F) corresponds to several sets of learnable weights, known as kernels, that look for different features or patterns of the input image.

- Second, the **filter size** or **kernel size** (K) describes the width and the height of the filters that are used in the convolution operation. Normally, the kernel is a two-dimensional matrix of size K×K. However, another possible approach is considering a three-dimensional matrix, in which the third dimension describes the number of multiple color channels (e.g., RGB). The number of color channels of the kernel should match with the number of color channels of the input image.

- Third, the **zero-padding** (P) defines how the convolution operation is performed, based on which the generated output size changes. Depending on this hyperparameter, the input is padded or not with zeros around the border.

- Finally, the **stride** (S) corresponds to the number of positions that the filter is slid each time. This value can also produce a variation on the output size.

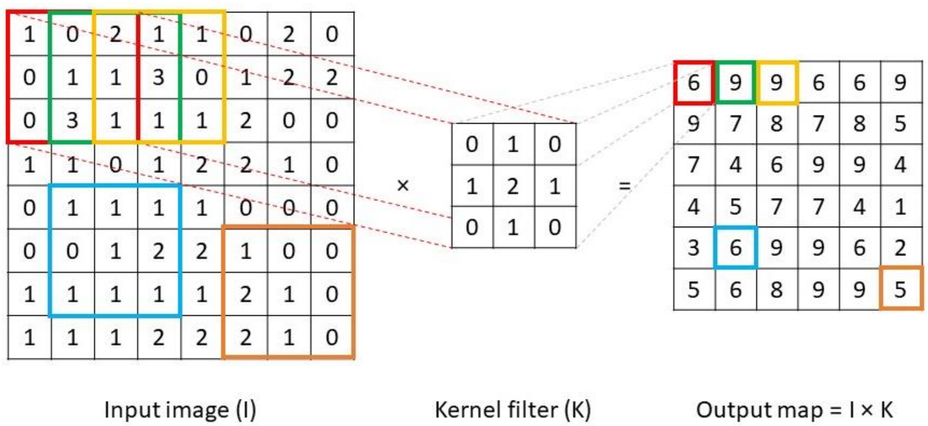

Figure 8: *Convolution operation with F=1, K=3, P=0 and S=1*

Taking all these definitions into account, the number of trainable parameters (TP) is defined as:

$$TP = weights + biases = K^2 \times F + F \qquad \textbf{\textit{(Eq. 1)}}$$

The number of connections (NC) is given by:

$$NC = output\_size^2 \times TP \qquad \textbf{\textit{(Eq. 2)}}$$

in which the output size corresponds to:

$$output\_size = input\_size - K + 1 \quad \textbf{\textit{(Eq. 3)}}$$

In the case of 3D images, the kernel has the same depth as the input image. When performing 3D convolutions, the 3D filter moves in three directions: width, height, and channels; and generates a volumetric output (see Figure 9).

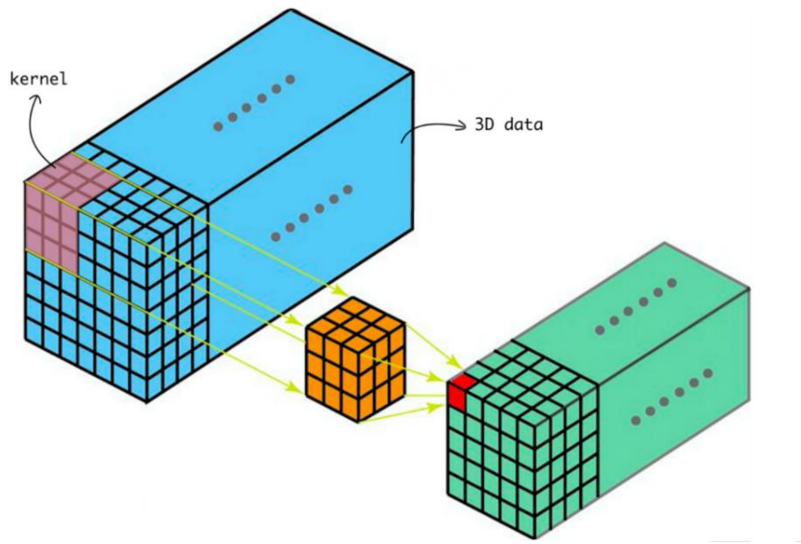

Figure 9: *3D convolution* [15]

Activation function

In neural networks, the activation function of a node (also known as non-linearity layer) defines the output of that node given an input or set of inputs (see Figure 7). More precisely, it produces a mapping from an input real number to a real number within a specific range in order to determine whether or not the information within the node is useful [11].

In consequence, given a combination of inputs and weights from the previous layer, the activation function controls how the information is processed and passed to the next layer. These mathematical equations are crucial when talking about the accuracy, computational efficiency, convergence, and convergence speed of a model.

An ideal activation function is both nonlinear and differentiable. Nonlinear behavior of an activation function allows our neural network to learn nonlinear relationships in the data. Differentiability is important because it allows to backpropagate the error in the neural network model when training to optimize the weights.

Apart from the *softmax* output activation function, the ReLU (rectified linear unit) is one of the most popular activation functions [16], especially in CNNs. Non-linear activation functions help the network learn complex data and provide accurate predictions. Mathematically, ReLU is defined as:

$$y = \max(0, x) \qquad \textbf{\textit{(Eq. 4)}}$$

This function is cheap to compute, trains rapidly, converges fast and is sparsely activated. Neurons in a network have different roles and therefore should be activated by different signals. Being sparsely activated allows neurons to process meaningful aspects of the problem. Nowadays, there are several variants of the ReLU activation function, such as leaky ReLU that solves the issue of zero gradients [4].

Although there are other activation functions, such as *perceptron,* or *sigmoid*, *tanh* and *arctan*, those functions are not widely used nowadays because of their non-differentiability or their backpropagation limitations respectively.

Pooling layer

The pooling layer or sub-sampling layer is commonly placed between convolutional layers with the objective of reducing the number of trainable parameters and computation in the neural network. In order to achieve a better power performance, this kind of layer reduces the spatial size of the input feature maps. By performing this size reduction, it also helps avoiding overfitting in the CNN. Normally, it operates with filters of size 2×2 and a stride of 2.

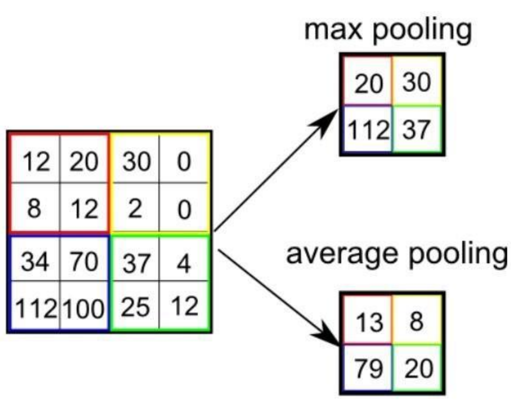

Figure 10: *Example of Max Pooling and Average Pooling* [12]

There are two possible approaches for implementing this behavior: Max Pooling and Average Pooling (see Figure 10). The Max Pooling returns the maximum value from the portion of the input map covered by the filter. On the other hand, the Average Pooling computes the average of all the values covered by the kernel. Max Pooling has been shown to have faster convergence and better classification performance [4].

3D pooling operation follows the same sliding approach as 3D convolutions (see Figure 9).

Fully-Connected layer

Fully-Connected (FC) layers are responsible of compiling the data extracted from previous convolutional layers and pooling layers to generate the final output classification of the neural network. In this kind of layers, all the inputs from the previous layer are connected to every neuron of the following layer. The main purpose is to flatten the feature maps into a single vector of values that represents the probability of each feature belonging to an output classification label [17].

The operation performed by a fully-connected layer consists basically of multiplying the weights per the input values and adding the bias terms. By executing this computation, non-linear combinations of the different features can be learned.

Classifier layer

After passing through the fully-connected layers, the final classification layer uses an output activation function, such as *softmax* or SVM, to get the probabilities of the input being one of the output classes or labels.

The Softmax classifier gives the normalized probabilities of a list of potential outcomes, which basically means that it takes an arbitrary input vector and converts it into a vector of values between zero and one that sum to one. This function is commonly used in multi-class classifications problems using deep learning techniques.

The SVM (Support Vector Machines) classifier is applied for finding a hyperplane in a N-dimensional space (where N is equal to the number of features) that distinctly classifies the different data points of each class or label [18].

The most common types of CNN architectures are featured below.

LeNet [19] [20]

LeNet is one of the first CNNs, developed by Yann LeCun in the 1990s. In the beginning, it was mainly used for character recognition applications, like reading zip codes or digits. The LeNet architecture is considered simple and small regarding memory footprint, although it is efficient enough to provide good results in many fields. The latest approach is called LeNet-5, which is a 5-layer CNN that reaches 99.2% accuracy on isolated character recognition.

AlexNet [21]

AlexNet is a deep CNN, designed by Alex Krizhevsky in 2012. It consists of 650,000 neurons and 60 million parameters, which makes it deeper and wider than LeNet-5. The uniqueness of this architecture resides in the kernel size of 11×11 and the stride of 4 in the first convolutional layer. In total, this architecture consists of 8 layers: 5 convolutional layers followed by Max Pooling layers, and 3 FC layers.

GoogleNet [22]

GoogleNet is a CNN architecture proposed by Szegedy in 2014. This network uses an inception module and consists of several dense components that create an optimal deep sparse structure with a lower number of parameters. The advantage of this method is multi-level feature extraction from each input at the same time.

VGGNet [23]

VGGNet is a CNN model developed by Karen Simonyan and Andrew Zisserman in 2014. In this case, the depth of the architecture plays a significant role in its performance, since it only uses small 3×3 kernel filters. This network achieves high performance, although it needs more memory and more parameters.

ResNet [24]

ResNet is a CNN architecture designed by Kaiming He *et al.* in 2015. ResNet (also known as residual network) consist of residual blocks, which introduce the *identity shortcut connection* concept to skip one or more layers (see Figure 11).

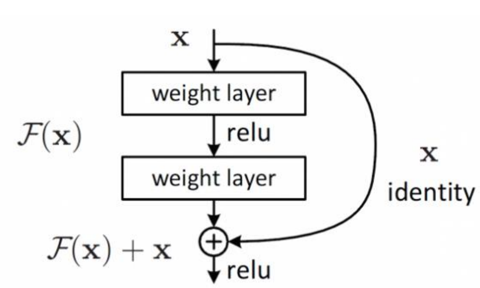

Figure 11: *Identity shortcut connection in ResNet* [24]

Chapter 3:

Design and implementation

The block diagram of an AD detection system is presented in Figure 12. In this chapter, each block of the system is analyzed in detail.

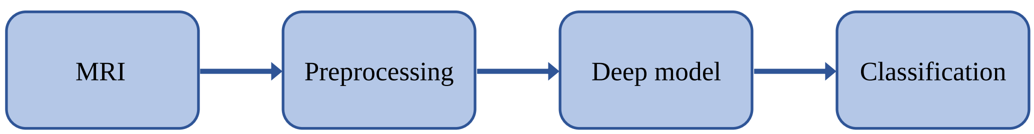

Figure 12: *Block diagram of an AD detection system*

3.1. Dataset description

The dataset used to train and validate the model is OASIS neuroimaging dataset of the brain [25]. The OASIS-2, which includes longitudinal MRI data in nondemented and demented older adults, is the release chosen to implement the deep AD classification model [26].

Table 1: *OASIS-2 brain dataset*

Classes	Number of subjects	Training dataset	Testing dataset	Total
Nondemented	72	156	40	196
Demented	64	111	29	140
Total	136	267	69	336

The OASIS-2 set consists of an MRI collection of 150 subjects (88 women) aged 60 to 96. The subjects are all right-handed and include both men and women. Each subject was scanned on two or more visits separated by at least one year, for a total of 373 imaging sessions. On the one hand, 72 of the subjects were characterized as nondemented throughout the study. The mean age of nondemented subjects was 75.82, and the number of female was higher than the number of male. On the other hand, 64 of the included subjects were characterized as demented at the time of their initial visits and remained so for subsequent scans. . The mean age of demented subjects was 74.95, and in contrast the number of women was lower than the number of men. Another 14 subjects were characterized as nondemented at the time of their initial visit and were subsequently characterized as demented at a later visit, which are labeled as converted subjects [25] [26].

The experiments that are carried out in this thesis perform binary classification between NC patient and AD patient. For this reason, the scans of the converted subjects or the MCI subjects are not considered at this stage of the project.

The classification task consists of two main phases: training and testing. In the training step, the input image is labeled to extract the features and train the algorithm. In the testing step, the classifier is validated by predicting the output and comparing it to the labeled data. In consequence, the dataset is split in two groups: 80% of the scans for training stage, and 20% of the scans for testing phase.

3.2. Data preprocessing

Data preprocessing is a crucial step when training an artificial neural network. In literature, there are several preprocessing methods depending on the type of data, the algorithm and the application that is performed. Preprocessing the data allows to obtain more meaningful and accurate results in the output of the model. An overview of the preprocessing steps is shown in Figure 13.

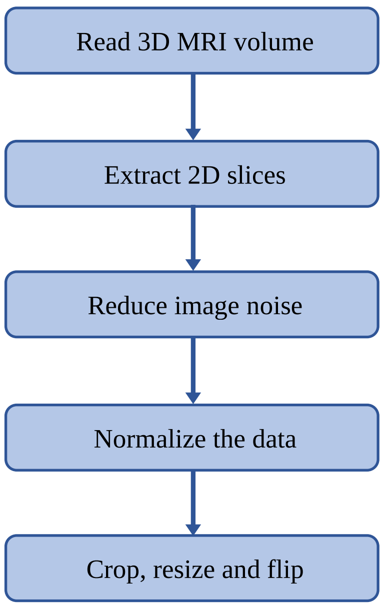

Figure 13: *Overview of the preprocessing steps*

The 3D MRI volumes are read by using a library called *NiBabel*, a neuroimaging tool in Python [27]. In the case of 2D CNN, the extraction of the 2D slices is the following step, whose approach is described in detail in Section 3.3.2.

The main step of the preprocessing process is normalization. This widely used technique aims to scale the data between a maximum and a minimum, making it to have zero mean and unit variance. Standardizing the data allows to speed up the training process by using a higher learning rate and a lower number of epochs. The learning rate is a configurable hyperparameter used in the training phase of a neural network to control how quickly the model is adapted to the problem [28], and an epoch is a pass through the entire dataset. In other words, making a similar data distribution of the input images leads to a faster convergence of the neural network.

The normalization formula is defined as:

$$z = \frac{x_i - \mu}{\sigma} \qquad \textbf{\textit{(Eq. 5)}}$$

where μ is the mean and σ is the standard deviation from the mean.

Reducing image noise is another preprocessing step. In this case, an image filter is added to the original image to eliminate some of the noise and smooth the edges. Cropping, resizing, and flipping (vertically or horizontally) the images can also help to improve the robustness of the model.

Image preprocessing is a key stage of an AD detection system. Other techniques, such as skull removal or data augmentation can be helpful to increase the accuracy, convergence, and robustness of the model. Segmentation is another possible preprocessing method, although it can make high-level information hard to convey since the region of interest in this case is wide. Future work can be developed in this data preprocessing step to improve the model behavior.

3.3. Model definition

Different 2D and 3D CNN architectures are implemented in this thesis. The schematics of the two approaches are illustrated in Figure 14.

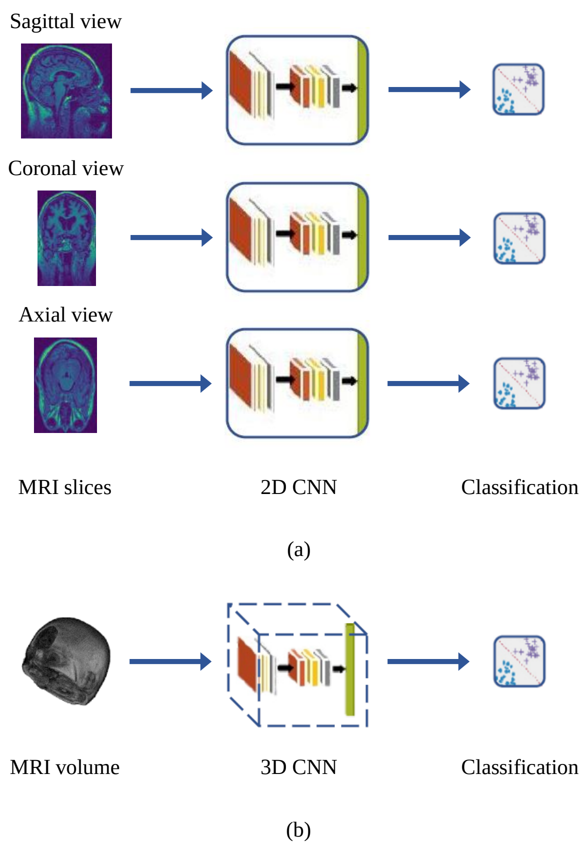

Figure 14: *Schemes of the applied approaches* [4]

3.3.1. CNN architectures

In this thesis, 4 different CNN architectures – AlexNet, VGG, GoogleNet, and ResNet - are implemented to train the brain image classifier. The description of these architectures is presented in Section 2.2.1. Several models with different depth and dimensions are tested. The list of the applied CNN structures is included in Table 2. The framework used to implement the models is Pytorch.

Table 2: *Implemented CNN architectures*

2D CNN	3D CNN
AlexNet	AlexNet
VGG-16	ResNet18
ResNet18	ResNet34
VGG-19	
GoogleNet	
ResNet34	
ResNet50	
ResNet101	

3.3.2. 2D vs. 3D

CNNs are a widely used approach to detect AD in deep intelligent CAD systems. Initially, CNNs were proposed to recognize patterns from 2D images. Nevertheless, nowadays the use of 3D CNNs is becoming more and more popular and effective.

In this thesis, both 2D and 3D structures are implemented and compared in order to identify which is the best choice in the case of detecting AD based on MRI scans.

2D CNNs use a small number of parameters during the training process and are great at capturing spatial features. In contrast, 3D CNNs need many parameters to be trained, which could lead to overfitting. However, they have the potential to capture the temporal information present in MRI volumes.

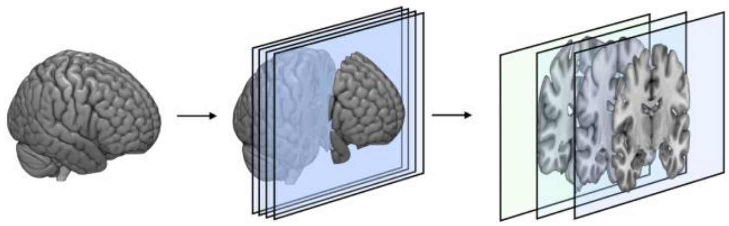

Figure 15: *Example of slicing an MRI volume* [29]

When implementing 2D CNN models based on 3D MRI data, the input volumes need to be sliced before fitting them into the network under the assumption that certain features of interest in 3D MRIs are preserved. Generally, 2D CNNs capture the middle part of brain scans as the input data and ignore the reminder [4]. In this thesis, the applied slicing approach consist of taking the middle 30 frames of the MRI scans as independent labeled input images. Figure 16 illustrates the importance of each MRI view by highlighting the regions of the brain from where relevant information can be extracted as AD progresses.

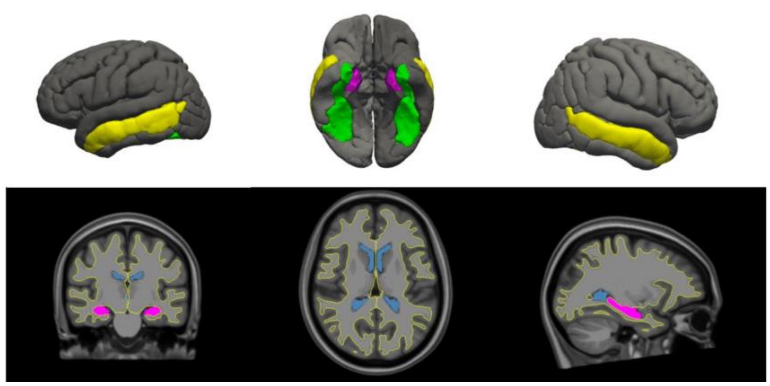

Figure 16: *Brain regions affected by AD in each MRI view* [30]

3.3.3. Backpropagation

Optimization is a crucial part of a neural network, where the weights and biases of the neurons are updated. In a neural network, backpropagation is used as a learning algorithm to compute a gradient descent with respect to weights and biases. The main purpose is to train the network efficiently by exploiting the chain rule. After each feed-forward pass through the network, the backpropagation algorithm does the backward-pass to adjust the model's parameters, boost the prediction and therefore minimize the loss function, which calculates how inaccurate the network is performing. The selection of this loss function depends on the performed task. In the case of image classification applications such as AD detection systems, cross entropy loss is the most common cost function.

Another relevant function during the backpropagation and training phase is the optimizer, which modifies the attributes of the network (e.g., learning rate and momentum). In image classification tasks, Adam is the most efficient optimizer. The optimal learning rate used to train all the models is 0.01.

Chapter 4:

Results

In order to evaluate the performance of the models, several metrics are considered in Table 3: training and testing accuracies, precision, recall and F1 score. The accuracy of a network describes how the model performs across all classes. The precision measures the model's accuracy in classifying a sample as positive. The recall measures the model's ability to detect positive samples. When a model has high recall but low precision, then the model classifies most of the positive samples correctly, but it has many false positives. When a model has high precision but low recall, then the model is accurate when it classifies a sample as positive, but it can only classify a few positive samples [31]. The F1 score is a statistical measure defined as the harmonic mean between precision and recall.

In this case, since the goal is to detect all the positive samples and misclassifying a negative sample would not lead to a significant error, recall can be considered as a more relevant performance metrics. Nevertheless, comparing the training and testing accuracies of the different CNN architectures is the main evaluation criteria.

Table 3: *Model evaluation*

Model architecture	CNN dimension	View of MRI slices	Training accuracy [%]	Testing accuracy [%]	Precision	Recall	F1 score
AlexNet	2D	Sagittal	78.43	78.07	0.72	0.73	0.72
	2D	Coronal	74.68	75.48	0.69	0.70	0.69
	2D	Axial	77.68	76.55	0.69	0.71	0.69
	3D		81.54	77.21	0.71	0.78	0.74
VGG-16	2D	Sagittal	75.30	76.19	0.65	0.66	0.66
	2D	Coronal	75.19	74.81	0.65	0.66	0.65
	2D	Axial	73.56	72.43	0.63	0.65	0.65
ResNet18	2D	Sagittal	76.55	75.07	0.76	0.83	0.79
	2D	Coronal	76.93	77.97	0.82	0.88	0.86
	2D	Axial	78.94	78.52	0.86	0.93	0.89
	3D		80.11	75.07	0.62	0.66	0.63
VGG-19	2D	Sagittal	76.30	76.19	0.66	0.66	0.66
	2D	Coronal	75.52	75.55	0.65	0.66	0.66
	2D	Axial	73.56	72.43	0.64	0.65	0.64
GoogleNet	2D	Sagittal	74.68	72.93	0.53	0.49	0.51
	2D	Coronal	73.44	72.06	0.53	0.50	0.51
	2D	Axial	71.69	71.57	0.53	0.50	0.52
ResNet34	2D	Sagittal	75.43	76.93	0.72	0.82	0.74
	2D	Coronal	78.05	78.31	0.86	0.93	0.89
	2D	Axial	76.68	76.38	0.74	0.83	0.79
	3D		77.50	74.21	0.55	0.61	0.58
ResNet50	2D	Sagittal	75.69	75.57	0.57	0.53	0.55
	2D	Coronal	75.30	74.81	0.57	0.53	0.55
	2D	Axial	73.77	74.69	0.56	0.55	0.55
ResNet101	2D	Sagittal	75.19	74.30	0.57	0.52	0.55
	2D	Coronal	74.55	74.52	0.57	0.52	0.55
	2D	Axial	73.77	74.6	0.53	0.52	0.53

As highlighted in Table 3, the CNN architectures that better perform when detecting AD based on MRI scans are AlexNet, ResNet18 and ResNet34. The plots that are shown below correspond to the best model behavior for each 2D view and 3D structure.

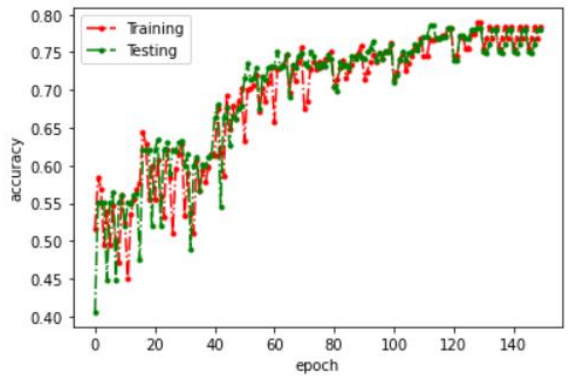

Figure 17: *2D Sagittal AlexNet performance*

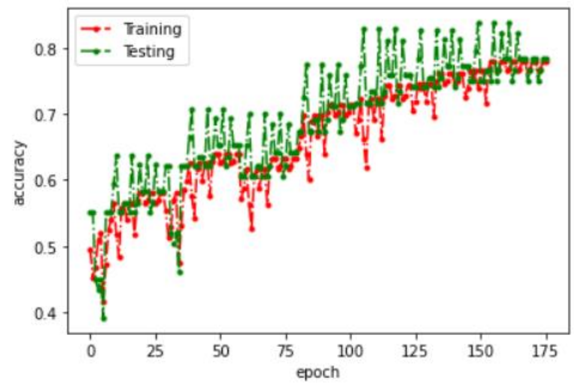

Figure 18: *2D Coronal ResNet34 performance*

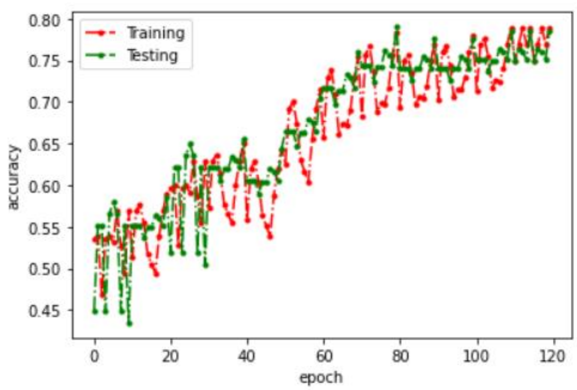

Figure 19: *2D Axial ResNet18 performance*

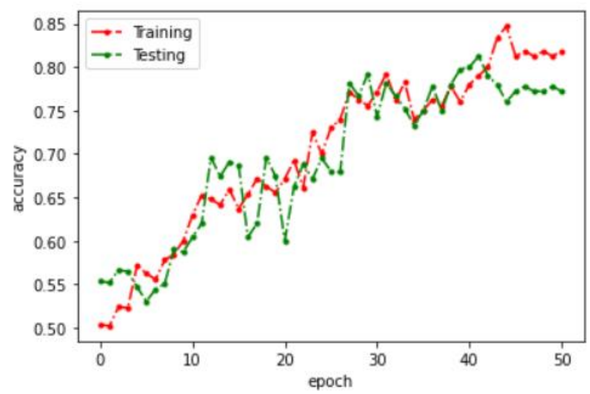

Figure 20: *3D AlexNet performance*

The lowest accuracy result is 71.57% for 2D Axial GoogleNet, and the highest accuracy result is 81.54% for 3D AlexNet. By comparing these numbers to similar approaches in current literature (e.g. [4]), the obtained results are remarkable and considered to be within the established performance accuracy range, although they are not able to beat the state-of-the-art by reaching close to 90% of accuracy. Two main reasons can justify this behavior: the amount of input data, and the implemented MRI preprocessing techniques, which act as a bottleneck to the performance of the system. In Section 5.1, possible future improvements in each of these directions are described in detail.

Table 4: *Performance evaluation*

Model architecture	CNN dimension	View of MRI slices	Average training time per epoch [s]	Number of epochs
AlexNet	2D	Sagittal	25.23	150
	2D	Coronal	20.53	120
	2D	Axial	21.36	130
	3D		160.73	45
VGG-16	2D	Sagittal	26.04	165
	2D	Coronal	25.79	160
	2D	Axial	28.66	150
ResNet18	2D	Sagittal	25.08	100
	2D	Coronal	21.56	110
	2D	Axial	20.91	115
	3D		231.32	50
VGG-19	2D	Sagittal	26.33	175
	2D	Coronal	25.11	165
	2D	Axial	29.07	160
GoogleNet	2D	Sagittal	25.36	170
	2D	Coronal	24.06	165
	2D	Axial	23.51	160
ResNet34	2D	Sagittal	20.54	160
	2D	Coronal	19.95	170
	2D	Axial	21.33	165
	3D		361.80	60
ResNet50	2D	Sagittal	26.88	150
	2D	Coronal	26.33	155
	2D	Axial	29.43	145
ResNet101	2D	Sagittal	26.09	130
	2D	Coronal	28.28	135
	2D	Axial	30.21	125

To further evaluate the experimental results of each architecture, two other evaluation criteria are considered in Table 4: average training time per epoch, and total number of required epochs to reach optimal performance. By multiplying both measures, the total training time for each model can be obtained.

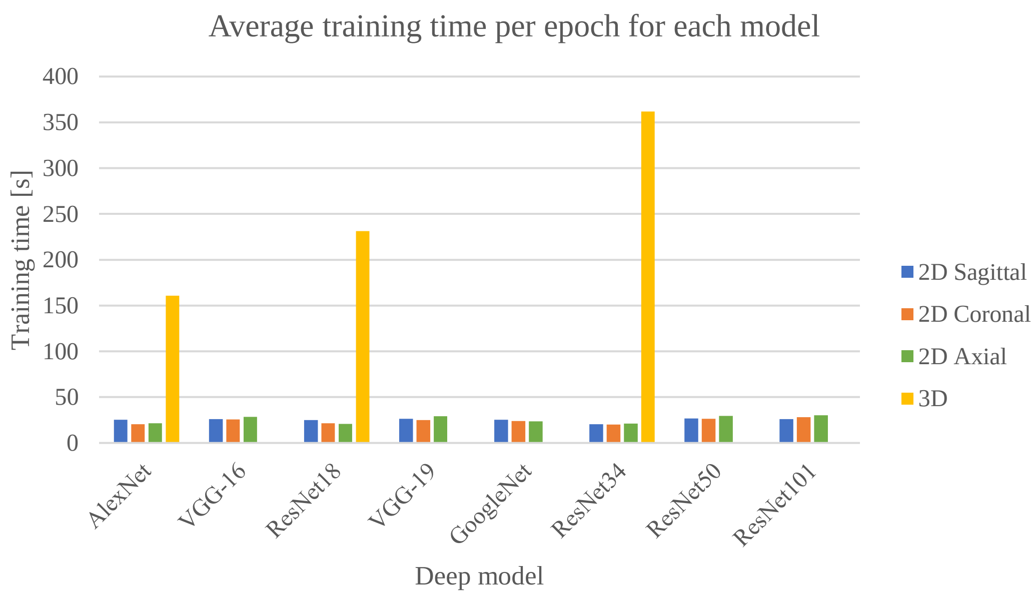

Figure 21: *Average training time per epoch for each deep model*

As illustrated in Figure 20, 3D models have a larger training time than 2D models. Nevertheless, they need a lower number of epochs to be optimally trained (see Figure 22). By comparing the total training time of 2D and 3D CNN architectures, the results show that in average 3D deep models take longer to train, but their performance is slightly better. Figure 21 illustrates the behavior of 2D CNN architectures in terms of timing. As shown, ResNet34 is the fastest model, followed by ResNet18 and AlexNet. This fact is expected, because normally deeper models or models with more learnable parameters take longer to train. In general, deeper models need fewer number of epochs in the training phase, however each epoch is more time consuming.

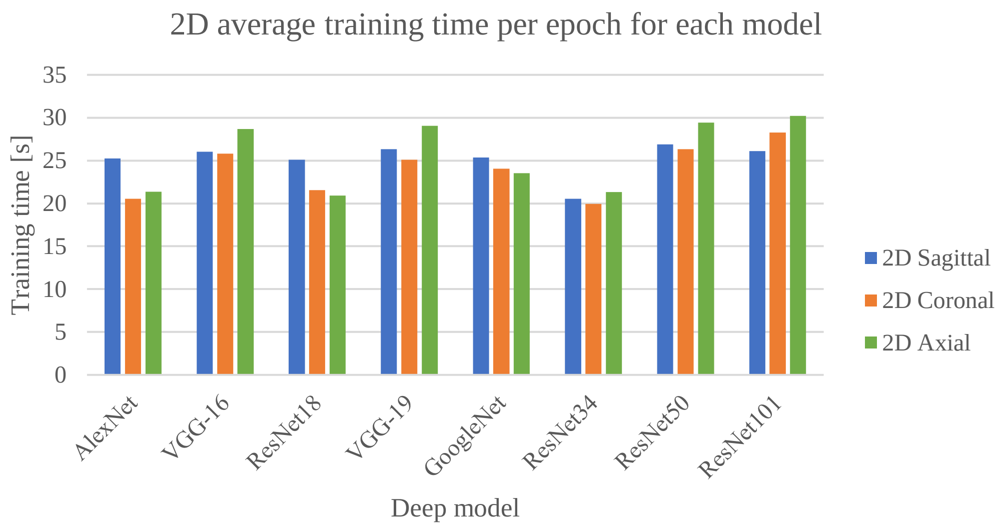

Figure 22: *2D average training time per epoch for each deep model*

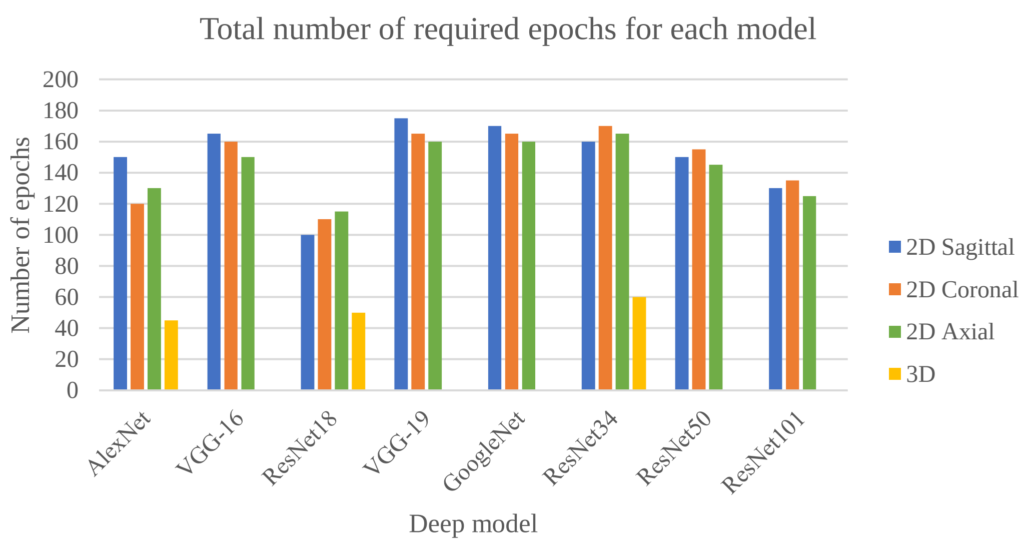

Figure 23: *Total number of required epochs for each deep model*

Chapter 5:

Conclusions

This thesis presents the design, implementation and experiments of several MRI-based AD detection approaches using CNNs. As seen throughout the project's results, building deep CNN models to perform binary detection of AD in early stages is possible and can achieve competitive outcomes, although it requires to carefully consider different design stages, preprocessing techniques, and network structures. In general, AlexNet, ResNet18 and ResNet34 have the best performance. The main weakness of these approaches is the necessity of large datasets.

Comparing 2D and 3D architectures leads to remarkable results, showing that 3D CNN structures can successfully exploit the temporal as well as the spatial information present in MRI volumes. In this case, overfitting is not a major inconvenient, as seen in the training and testing accuracy results. Nevertheless, if this problem appears at some point when performing future improvements, solving it by increasing the number of input scans or applying data augmentation should be effective. When evaluating 2D MRI slices, the views that are capable of achieving the highest accuracy results are sagittal and coronal. Implementing multi-view models can also help obtaining more robust and accurate 2D classifiers.

5.1. Future work

Promising future work towards the improvement of the proposed models to better detect, diagnose, and classify AD is possible in terms of data, preprocessing and architecture.

In terms of the amount of available data, OASIS dataset has a third release of their brain scans, which consists of three times more MRI volumes than the current used package. If getting access to this wider amount of data, the lack of input images and the possibility of overfitting could be easily overcome.

In terms of preprocessing, new and more effective techniques can be considered. For example, skull removal or image segmentation are two popular approaches. In the case of skull removal, recent software programs that can perform this task directly are used. When talking about segmentation, the benefits of this method in AD are unknown, because the region of interest of the brain is wider than in other diseases, and the fact that it does not convey high-level information could lead to worse accuracy results.

In terms of CNN models, possible improvements in the structure of the network such as multi-path or cascade approaches in the case of 2D CNN can help to achieve more competitive performance.

In the future, other possibilities are the implementation of multi-class classification between three stages (NC, AD and MCI), and the comparison of the behavior of the different architectures when using other types of brain images, such as PET.

REFERENCES

[1] Alzheimer Association, "2022 Alzheimer's disease facts and figures", *Alzheimer's Dement.*, 2022.

[2] "Dementia", https://www.who.int/news-room/fact-sheets/detail/dementia

[3] "The top 10 causes of death", https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death

[4] A. Ebrahimi, and S. Luo, "Convolutional neural networks for Alzheimer's disease detection on MRI images", *J. Med. Imaging*, 2021.

[5] "Alzheimer's Disease Fact Sheet | National Institute on Aging", https://www.nia.nih.gov/health/alzheimers-disease-fact-sheet#symptoms

[6] A. Nawaz, S. M. Anwar, R. Liaqat, J. Iqbal, U. Bagci, and M. Majid, "Deep Convolutional Neural Network based Classification of Alzheimer's Disease using MRI Data", *Proc. - 2020 23rd IEEE Int. Multi-Topic Conf. INMIC 2020*, 2020.

[7] J. Islam and Y. Zhang, "Brain MRI analysis for Alzheimer's disease diagnosis using an ensemble system of deep convolutional neural networks", *Brain Informatics*, 2018.

[8] "What causes dementia? - Queensland Brain Institute - University of Queensland", https://qbi.uq.edu.au/dementia/dementia-causes-and-treatment

[9] I. O. Korolev, "Alzheimer's Disease : A Clinical and Basic Science Review", *Med. Student Res. J.*, 2014.

[10] K.A.N.N.P. Gunawardena, R. N. Rajapakse, and N. D. Kodikara, "Applying convolutional neural networks for pre-detection of Alzheimer's disease from structural MRI data", *2017 24th Int. Conf. Mechatronics Mach. Vis. Pract. M2VIP 2017*, 2017.

[11] M. P. Hosseini, S. Lu, K. Kamaraj, A. Slowikowski, and H. C. Venkatesh, *Deep Learning Architectures*, 2020.

[12] "A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way | by Sumit Saha | Towards Data Science", https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[13] J. Bernal *et al.*, "Deep convolutional neural networks for brain image analysis on magnetic resonance imaging: a review", *Artif. Intell. Med.*, 2019.

[14] "CS231n Convolutional Neural Networks for Visual Recognition", https://cs231n.github.io/convolutional-networks/

[15] "Understanding 1D and 3D Convolution Neural Network | Keras | by Shiva Verma | Towards Data Science", https://towardsdatascience.com/understanding-1d-and-3d-convolution-neural-network-keras-9d8f76e29610

[16] "A Practical Guide to ReLU. Start using and understanding ReLU… | by Danqing Liu | Medium", https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7

[17] "An Intuitive Explanation of Convolutional Neural Networks – the data science blog",

https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

[18] "Support Vector Machine — Introduction to Machine Learning Algorithms | by Rohith Gandhi | Towards Data Science", https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

[19] S. Wu, W. Wei, and L. Zhang, "Comparison of machine learning algorithms for handwritten digit recognition", *Commun. Comput. Inf. Sci.*, 2018.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition", *Proc. IEEE,* 1998.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", 2012.

[22] C. Szegedy *et al.*, "Going deeper with convolutions", *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2015.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, 2015.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2016.

[25] "OASIS Brains - Open Access Series of Imaging Studies", https://www.oasis-brains.org/

[26] D. S. Marcus, A. F. Fotenos, J. G. Csernansky, J. C. Morris, and R. L. Buckner, "Open access series of imaging studies: Longitudinal MRI data in nondemented and demented older adults", *J. Cogn. Neurosci.*, 2010.

[27] "Neuroimaging in Python — NiBabel 4.0.0 documentation", https://nipy.org/nibabel/

[28] "Understand the Impact of Learning Rate on Neural Network Performance", https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/

[29] J. Bin Bae *et al.*, "Identification of Alzheimer's disease using a convolutional neural network model based on T1-weighted magnetic resonance imaging", *Sci. Rep.*, 2020.

[30] S. Tabarestani *et al.*, "A distributed multitask multimodal approach for the prediction of Alzheimer's disease in a longitudinal study", *Neuroimage*, 2020.

[31] "Accuracy, Precision, and Recall in Deep Learning | Paperspace Blog", https://blog.paperspace.com/deep-learning-metrics-precision-recall-accuracy/