Reproducible and interpretable deep learning for the diagnosis, prognosis and subtyping of Alzheimer's disease from neuroimaging data

Elina Thibeau-Sutre

SORBONNE UNIVERSITÉ

DOCTORAL THESIS

Reproducible and interpretable deep learning for the diagnosis, prognosis and subtyping of Alzheimer's disease from neuroimaging data

Author:
Elina THIBEAU-SUTRE

Referees:
Pierrick COUPÉ
Ivana ISGUM

Supervisors:
Ninon BURGOS
Olivier COLLIOT
Didier DORMONT

A thesis submitted in fulfillment of the requirements for the degree of PhD

in the

ARAMIS Lab
Institut du Cerveau - Paris Brain Institute (ICM), Inserm U 1127, CNRS UMR 7225, AP-HP Hôpital de la Pitié Salpêtrière and Inria

December 14, 2021

SORBONNE UNIVERSITÉ

Abstract

ARAMIS Lab
Institut du Cerveau - Paris Brain Institute (ICM), Inserm U 1127, CNRS UMR 7225, AP-HP
Hôpital de la Pitié Salpêtrière and Inria

Reproducible and interpretable deep learning for the diagnosis, prognosis and subtyping of Alzheimer's disease from neuroimaging data

by Elina THIBEAU-SUTRE

The goal of this PhD was the validation of the existence and the discovery of new subtypes of Alzheimer's disease, the first cause of dementia worldwide. Indeed, despite its discovery more than a century ago, this disease is still not well defined and existing treatments are only weakly effective, possibly because several phenotypes exist within the disease. In order to explore its heterogeneity, we employed deep learning methods applied to a neuroimaging modality, structural magnetic resonance imaging.

However, the discovery of important methodological biases in many studies in our field, as well as the lack of consensus regarding deep learning interpretability, partly changed the main objective of the PhD to focus more on issues of validation, robustness and interpretability of deep learning. Then, to correctly assess the ability of deep learning to detect Alzheimer's disease, three experimental studies were conducted. The first one is a study of deep learning methods for Alzheimer's classification and allowed a fair comparison of the methods. The second study found a lack of robustness of classification with deep learning in terms of atrophy patterns discovered using interpretability methods. Finally, the last study proposed a subtype discovery method aided by data augmentation. Although it works on synthetic data, it does not generalize to real data.

Experimental results of this PhD were obtained thanks to ClinicaDL, one major contribution of this PhD. It is an open source Python library that was used to improve the reproducibility of deep learning experiments.

SORBONNE UNIVERSITÉ

Résumé

ARAMIS Lab
Institut du Cerveau - Paris Brain Institute (ICM), Inserm U 1127, CNRS UMR 7225, AP-HP Hôpital de la Pitié Salpêtrière and Inria

Méthodes d'apprentissage profond reproductibles et interprétables pour le diagnostic, le pronostic et l'identification de sous-groupes de la maladie d'Alzheimer à partir de données de neuroimagerie

par Elina THIBEAU-SUTRE

L'objectif de cette thèse était la validation de l'existence ainsi que la découverte de nouveaux sous-types au sein de la maladie d'Alzheimer, première cause de démence au monde. En effet, malgré sa découverte il y a plus d'un siècle, celle-ci n'est toujours pas bien définie et les traitements existants ne montrent qu'une faible efficacité, ce qui pourrait être dû à l'existence de phénotypes différents au sein de la maladie. Afin d'explorer son hétérogénéité, nous avons employé des méthodes d'apprentissage profond appliquées à une modalité de neuroimagerie, l'imagerie par résonance magnétique structurelle.

Cependant, la découverte de biais méthodologiques importants dans de nombreuses études de notre domaine, ainsi que l'absence de consensus de la communauté sur la manière d'interpréter les résultats des méthodes d'apprentissage profond a fait en partie dévier la thèse de son objectif principal pour s'orienter d'avantage vers des problématiques de validation, de robustesse et d'interprétabilité de l'apprentissage profond. Ainsi, trois études expérimentales ont été menées pour s'assurer de la capacité des réseaux profonds de correctement détecter la maladie. La première est une étude expérimentale de méthodes d'apprentissage profond pour la classification de la maladie d'Alzheimer et a permis d'établir une juste comparaison des méthodes. La seconde étude a permis de constater un manque de robustesse de la classification avec l'apprentissage profond en termes de motifs d'atrophie découverts à l'aide de méthodes d'interprétabilité. Enfin, la dernière étude propose une méthode de découverte de sous-types aidée par l'augmentation de données. Bien que fonctionnant sur des données synthétiques, celle-ci ne généralise pas aux données réelles.

Une contribution majeure de la thèse est la librairie ClinicaDL, grâce à laquelle les résultats expérimentaux de la thèse ont été produits de manière à être reproductibles.

Remerciements

I first want to thank the referees, Pierrick Coupé and Ivana Isgum as well as the other members of the jury for taking time to read my thesis and give me constructive comments, both on their reports and their questions after the PhD defense.

Je souhaite également remercier mes encadrants, Ninon Burgos, Olivier Colliot et Didier Dormont sans qui cette thèse n'aurait pas pu avoir lieu. Leurs conseils réguliers m'ont permis de me donner la motivation de continuer à avancer (même quand ça ne marchait pas, c'est à dire une part non négligeable du temps tout de même), tout en me laissant également le champ libre pour explorer ce que je voulais et me laisser participer à d'autres projets de l'équipe.

À ce propos, je remercie Baptiste Couvy-Duchesne pour avoir lancé et organisé le projet PAC2019 qui m'a permis de travailler avec Benoît Martin de manière étroite (et très productive !) pendant plusieurs semaines.

Je souhaite aussi remercier l'équipe Clinica. Assister à ces réunions et participer au développement du logiciel dès le début de mon stage m'a permis de me rendre compte qu'un code lisible et réutilisable n'était pas en option. Parmi eux, je ne remercierai jamais assez Alexandre Routier, qui m'a grandement aidée dès le début de mon stage à comprendre toutes les subtilités des structures des bases de données de neuroimagerie, et qui a également prodigué d'excellents conseils pour ClinicaDL. Je souhaite également remercier Simona Bottani et Jorge Samper-Gonzalez qui ont tous deux participé à ma compréhension globale de Clinica, la neuroimagerie et son prétraitement à mon arrivée. Merci encore d'avoir fait tout le travail préliminaire de traitement de données qui m'a permis de commencer facilement mon propre travail !

Le projet ClinicaDL n'aurait d'ailleurs pas été possible sans Mauricio Díaz, qui a lancé l'idée initiale du projet en créant un package à partir du code source de notre premier repo de code, et qui a patiemment relu et commenté toutes mes propositions de code depuis sa création. Depuis celui-ci a bien évolué, et je suis heureuse de voir que Ravi Hassanaly s'y implique beaucoup en y apportant plein de nouvelles améliorations : je n'aurai jamais eu la force de faire la dernière mise à jour majeure sans lui.

J'ai également eu le plaisir de pouvoir donner cours lors de la thèse. à ce sujet je souhaite remercier Olivier Colliot qui m'a emmené donner cours au MVA, Ninon Burgos qui m'a proposé d'intervenir dans des workshops, et surtout Johann Faouzi, avec qui j'ai co-écrit une grande partie des contenus de ces cours. Je remercie également tous les chargés de TD et TP avec qui j'ai collaboré dans le cadre de Sorbonne Université, ainsi que les étudiants, qui m'ont permis de redécouvrir des facettes de Python que j'ai ensuite incorporées dans ma thèse.

Je remercie également Clément Chadebec et Stéphanie Allassonnière avec qui j'ai réalisé la dernière collaboration de ma thèse. Cette collaboration fut particulièrement enrichissante et sympathique, et je souhaite le meilleur à Clément pour le reste de sa thèse qui je n'en doute pas sera excellent!

Je souhaite également remercier tous mes collègues d'ARAMIS pour leur soutien, leur bonne humeur, et leurs apports divers et variés à mon travail. Je suis très heureuse d'avoir pu faire ma thèse parmi vous !

Enfin, je remercie ma famille et mes amis d'avoir supporté les aléas de ma thèse et de m'avoir toujours soutenue. Ce dernier remerciement s'adresse tout particulièrement à Amaury Legrand, qui a supporté la cohabitation avec une thésarde pendant plus de trois ans avec calme et sérénité.

Scientific Production

JOURNAL PAPERS

1. Wen*, J., **Thibeau-Sutre*, E.**, Díaz-Melo, M., Samper-González, J., Routier, A., Bottani, S., Dormont, D., Durrleman, S., Burgos, N. and Colliot, O., "Convolutional Neural Networks for Classification of Alzheimer's Disease: Overview and Reproducible Evaluation", *Medical Image Analysis*, 63, 101694 (2020) doi:10.1016/j.media.2020.101694 – hal-02562504 (*: joint first authorship)

2. Couvy-Duchesne*, B., Faouzi*, J., Martin*, B., **Thibeau-Sutre*, E.**, Wild*, A., Ansart, M., Durrleman, S., Dormont, D., Burgos, N. and Colliot, O., "Ensemble Learning of Convolutional Neural Network, Support Vector Machine, and Best Linear Unbiased Predictor for Brain Age Prediction: ARAMIS Contribution to the Predictive Analytics Competition 2019 Challenge", *Frontiers in Psychiatry*, 11 (2020) doi:10.3389/fpsyt.2020.593336 – hal-03136463 (*: joint first authorship)

3. Burgos*, N., Bottani*, S., Faouzi*, J., **Thibeau-Sutre*, E.** and Colliot, O., "Deep learning for brain disorders: from data processing to disease treatment", *Briefings in Bioinformatics*, 22(2), 1560–1576 (2021) doi:10.1093/bib/bbaa310 – hal-03070554 (*: joint first authorship)

4. Ansart, M., Epelbaum, S., Bassignana, G., Bône, A., Bottani, S., Cattai, T., Couronné, R., Faouzi, J., Koval, I., Louis, M., **Thibeau-Sutre, E.**, Wen, J., Wild, A., Burgos, N., Dormont, D., Colliot, O. and Durrleman, S., "Predicting the progression of mild cognitive impairment using machine learning: A systematic, quantitative and critical review", *Medical Image Analysis*, 67, 101848 (2021) doi:10.1016/j.media.2020.101848 – hal-02337815

5. Routier, A., Burgos, N., Guillon, J., Samper-González, J., Wen, J. and Bottani, S., Marcoux, A., Bacci, M., Fontanella, S., Jacquemont, T., Wild, A., Gori, P., Guyot, A., Lu, P., Díaz, M., **Thibeau-Sutre, E.**, Moreau, T., Teichmann, M., Habert, M.-O., Durrleman, S. and Colliot, O., "Clinica: an open source software platform for reproducible clinical neuroscience studies", *Frontiers in Neuroinformatics*, 15 (2021) doi:10.3389/fninf.2021.689675 – hal-02308126

SUBMITTED JOURNAL PAPERS

1. **Thibeau-Sutre, E.**, Díaz, M., Hassanaly, R., Routier, A., Didier, D., Colliot, O., Burgos, N., "ClinicaDL: an open-source deep learning software for reproducible neuroimaging processing". Submitted to *Computer Methods and Programs in Biomedicine*. hal-03351976

2. Chadebec, C., **Thibeau-Sutre, E.**, Burgos, N. and Allassonnière, S., "Data augmentation on neuroimaging data with variational autoencoders". Submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence* (under major revision). arXiv: 2105.00026

3. Berenbaum, A.,Burgos, N., **Thibeau-Sutre, E.**, Bottani, S., Habert, M.-O., Colliot, O., Kas, A., "Classification automatisée des TEP-TDM cérébrales au 18F-FDG par intelligence artificielle : preuve de concept". Submitted to *Médecine Nucléaire*.

PEER-REVIEWED CONFERENCE PROCEEDINGS

1. **Thibeau-Sutre, E.**, Colliot, O., Dormont, D. and Burgos, N., "Visualization approach to assess the robustness of neural networks for medical image classification", *SPIE Medical Imaging*, 11313, 113131J, 2020 doi:10.1117/12.2548952 – hal-02370532 — Oral presentation

CONFERENCE ABSTRACTS

1. Wen*, J., **Thibeau-Sutre*, E.**, Samper-González, J., Routier, A., Bottani, S., Dormont, D., Durrleman, S., Colliot, O. and Burgos, N., "How serious is data leakage in deep learning studies on Alzheimer's disease classification?", presented at 2019 *OHBM Annual meeting - Organization for Human Brain Mapping*, 2019. hal-02105133 (*: joint first authorship).

2. Routier, A., Marcoux, A., Melo, M. D., Guillon, J., Samper-González, J., Wen, J., Bottani, S., Guyot, A., **Thibeau-Sutre, E.**, Teichmann, M., Habert, M.-O., Durrleman, S., Burgos, N. and Colliot, O., "New advances in the Clinica software platform for clinical neuroimaging studies", presented at *OHBM 2019*. hal-02549242

IN PREPARATION

1. **Thibeau-Sutre, E.**, Collin, S., Dormont, D., Burgos, N. and Colliot, O., "Interpretability of Machine Learning applied to Brain Disorders".

2. Cacciamani, F., Houot, M., **Thibeau-Sutre, E.**, Migliaccio, R., Epelbaum, S., "Neural Correlates of Awareness of Cognitive Decline, Memory and Executive Functions in Pre-Dementia Alzheimer's Disease: a Multimodal Study".

3. **Thibeau-Sutre, E.**, Couvy-Duchesne, B., Dormont, D., Colliot, O. and Burgos, N., "MRI with a higher field strength predicts Alzheimer's disease: A case example of bias in the ADNI dataset"

TALKS AND POSTERS

1. Poster – Annual meeting of the Organization for Human Brain Mapping (OHBM), Rome, Italy, June 2019. hal-03365742
2. Poster – ICM Days, Louan, France, January 2020. hal-03365775
3. Oral presentation – SPIE Medical Imaging: Image Processing conference, Houston, United States, February 2020. Video available on SPIE Digital Library.
4. Poster – ICM Welcome Days, online poster session, October 2020. hal-03365788

SCIENTIFIC POPULARIZATION

1. Atlas workshop – Paris Brain Institute, March 2020. website.
2. Salon Culture & Jeux Mathématiques Inria - online, Mai 2020. website.
3. AI4Health Winter School - online, January 2021. website.

Contents

List of Figures

List of Tables

List of Abbreviations

AD	Alzheimer's Disease
ADNI	Alzheimer's Disease Neuroimaging Initiative
AIBL	Australian Imaging, Biomarker & Lifestyle Flagship Study of Ageing
AE	AutoEncoder
BA	Balanced Accuracy
BIDS	Brain Imaging Data Structure
bvFTD	behavioural variant of Fronto-Temporal Dementia
CAM	Class Activation Maps
CAPS	ClinicA Processed Structure
CDR	Clinical Dementia Rating
CN	Cognitively Normal
CNN	Convolutional Neural Network
CT	Computed Tomography
FC	Fully Connected [layer]
FTD	Fronto-Temporal Dementia
GAN	Generative Adversarial Network
Grad-CAM	Gradient-weighted Class Activation Mapping
LIME	Local interpretable model-agnostic explanations
LRP	Layer-Wise Relevance
MAPS	Model Analysis and Processing Structure
MCI	Mild Cognitive Impairment
MMSE	Mini-Mental State Examination
MRI	Magnetic Resonance Imaging
NIFD	Frontotemporal lobar Degeneration Neuroimaging Initiative
OASIS	Open Access Series of Imaging Studies
PET	Positron Emission Tomography
pMCI	progressive Mild Cognitive Impairment
ROI	Region Of Interest
SHAP	SHapley Additive exPlanations
SMC	Subjective Memory Complaint
sMCI	stable Mild Cognitive Impairment
SVM	Support Vector Machines
T1w [MRI]	T1-weighted [Magnetic Resonance Imaging]
VAE	Variational AutoEncoder

Mathematical Notations

- X_0 is an input tensor given to a neural network, and X refers to any input, sampled from the set $\mathscr{X}$.
- y is a vector of target classes corresponding to the input.
- f is a network of L layers. The first layer is the closest to the input, the last layer is the closest to the output. A layer is a function.
- g is a transparent function which aims at reproducing the behaviour of f.
- w and b are the weights and the bias associated to a linear function (for example in a fully-connected layer).
- u and v are locations (set of coordinates) corresponding to a node in a feature map. They belong respectively to the set $\mathscr{U}$ and $\mathscr{V}$ which lists all locations possible of their feature map.
- $A_k^l(u)$ is the value of the feature map computed by layer l, of K channels at channel k, at position u.
- $R_k^l(u)$ is the value of a property back-propagated through the $l+1$, of K channels at channel k, at position u. R^l and A^l have the same number of channels.
- o_c is the output node of interest (in a classification framework, it corresponds to the node of the class c).
- S_c is an attribution map corresponding to the output node o_c.
- m is a mask of perturbations. It can be applied to X to compute its perturbed version X^m.
- Φ is a function producing a perturbed version of an input X.
- Γ_c is the function computing the attribution map S_c from the black-box function f and an input X_0.

Introduction

With population aging, neurodegenerative diseases (including Alzheimer's) are becoming a major public health issue. A better characterization of this disease, especially of its heterogeneity, would allow a better management of patients.

This heterogeneity, both at the clinical and morphological levels, has already been observed by several studies, but these did not include the normal variability of the population. Indeed, the variability between brains of patients is not necessarily due to the disease, but may come from the natural diversity of the human brains (that we will call in the following the normal variability). Even if our brains all have the same global structure, they differ in many ways between individuals. Some of this variability can be explained by observable characteristics such as the age, education level or sex, but most of it cannot be modelled in such a simple way. In particular, normal aging itself causes brain changes that resemble alterations due to neurodegenerative diseases, though to a lesser extent. If this normal process is mistaken for a pathological process, then there is a risk in creating subtypes greatly biased towards the age of the patients. In order to discover more consistent subtypes associated with complex biomarkers, we chose to use deep learning methods.

In the following, the reader will be introduced to the three main notions necessary to understand the thesis: Alzheimer's disease, neuroimaging data and classification with deep learning. As this PhD is based on data-driven methods, a focus will be made on the data sets collected on dementia.

Alzheimer's disease

Alzheimer's disease (AD) is the most common form of dementia worldwide: in 2016, it affected 43.8 millions people (Nichols et al., 2019). It causes a diversity of symptoms in patients which substantially deteriorate their living conditions. Though it was discovered more than one century ago, the mechanisms underlying the course of the disease remain hypothetical (Reitz, 2012). Thus, even if symptomatic treatments exist, their effect remains weak. There is therefore a real need to better model the disease.

Diagnosis of AD

Alzheimer's disease is a neurodegenerative disease of the elderly, then it affects subjects who may already suffer from other symptoms and diseases related to old age. Thus, it can be difficult to disentangle this disease from other problems encountered by the patients. In brief, AD diagnosis is made based on a battery of cognitive tests after the exclusion of

another major cause for cognitive loss. In the following, two of the cognitive tests used to diagnose dementia will be considered to define our populations:

- **Mini-Mental State Evaluation** *(MMSE)* Its score ranges from 0 to 30 (perfect performance). AD diagnosis is considered when MMSE < 27. It aims at evaluating orientation to place, orientation to time, memory, attention and concentration, language, and visual construction.
- **Clinical Dementia Rating** *(CDR)* Its score ranges from 0 (not demented) to 3. AD diagnosis is considered when CDR > 0. It describes five degrees of impairment in performance on each of six categories of cognitive functioning including memory, orientation, judgment and problem solving, community affairs, home and hobbies, and personal care.

The threshold values correspond to the ones used in ADNI and AIBL (see Section Data sets). Other criteria, such as age and education level, may modulate them: a young participant with a high education level may be considered as demented with a higher MMSE than an old participant with a low education level. However, this definition of AD is not unique and may differ in different cohorts (see Open Access Series of Imaging Studies (OASIS) description).

Biomarkers

Though they were not used to diagnose participants in the cohorts used, some biomarkers are characteristic of the disease (Jack et al., 2016):

- the presence of senile plaques, caused by the accumulation in the brain of **amyloid-β** protein,
- the presence of neurofibrillary tangles, formed by hyperphosphorylation of **tau** protein,
- neurodegeneration, particularly of the hippocampi, a region known to be linked with memory processing.

Typical AD patients have a regional atrophy in the medial temporal lobe and the temporal and parietal neocortex (see Figure 1). But even though most AD patients present an atrophied hippocampus, some AD cases have a spared hippocampus compared to the atrophy of the rest of their cortex. Then, clinicians concluded that the disease may include different subtypes.

Phenotypic heterogeneity

Au et al., 2015 assume that one possible reason for the failure of clinical trials could actually be the lack of consideration for the heterogeneity of the disease, though some clinical studies already proposed to break it into several subtypes.

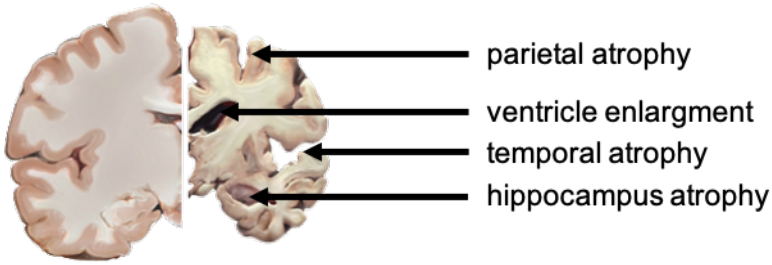


FIGURE 1: Comparison of a typical AD brain (right side) to a heathy brain (left side). Courtesy: https://www.nia.nih.gov/health/alzheimers-disease-fact-sheet.

Murray et al., 2011 used the location of neurofibrillary tangles in the hippocampus and three other cortical regions (mid-frontal, inferior temporal and superior temporal) in postmortem material to describe three main subtypes:

- hippocampal sparing (11%), corresponding to a significantly higher density of neurofibrillary tangles in cortical regions and significantly lower in the hippocampus,
- limbic (14%), corresponding to a significantly lower density of neurofibrillary tangles in cortical regions and significantly higher in the hippocampus,
- typical (75%) otherwise.

These subtypes also correspond to different demographic profiles. This way the hippocampal sparing subtype affected younger people with a higher male:female ratio, while the limbic subtype affected older people with a lower male:female ratio.

A larger diversity of subtypes is described by the review of Lam et al., 2013, which links some subtypes (illustrated in Figure 2) to related disorders:

- temporal (pure amnestic) is a late-onset subtype. Though memory may be significantly impaired other abilities remain borderline to normal. Biomarkers are found only in limbic regions. This subtype can be compared to the limbic subtype of Murray et al., 2011.
- left (language) is an early-onset subtype. In this case, the most affected cognitive function is language, and the biomarkers pattern is asymmetrical, with a greater involvement of the left temporal and parietal lobes. This subtype can be compared with logopenic progressive aphasia.
- right (visuoperceptive) is an early-onset subtype. Patients encounter visuospatial dysfunction and their right hemisphere is more affected than the left one. This subtype can be compared with posterior cortical atrophy.
- frontal (executive) is a rare early-onset subtype. It is associated with behavioural symptoms and a greater involvement of the frontal lobe. This subtype can be compared with fronto-temporal dementia.

- typical is a late-onset subtype. Patients are mainly amnestic, but other cognitive functions may also decline (contrary to the case of temporal subtype). The most affected regions are the temporal lobes and the hippocampi.

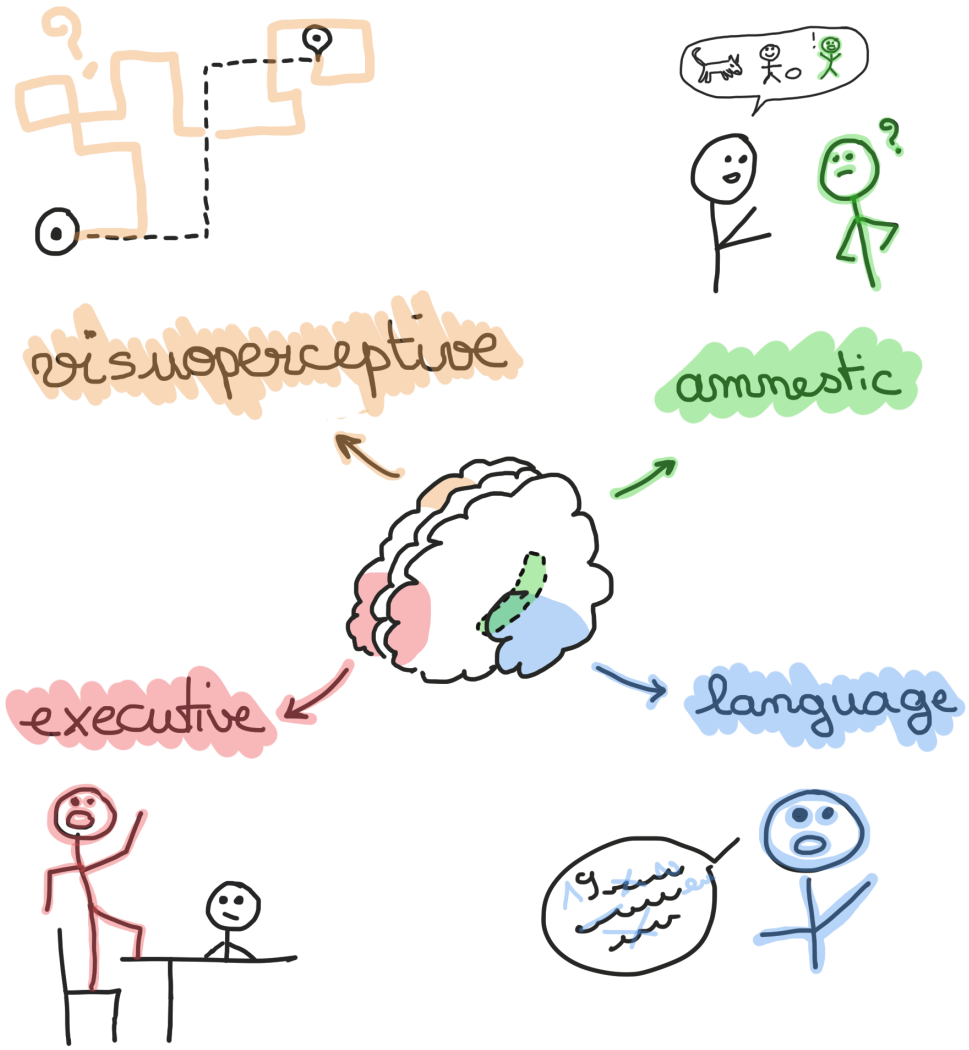


FIGURE 2: Illustration of the main regions affected and clinical symptoms associated to the subtypes identified by Lam et al., 2013.

Data sets

Three publicly available data sets have been mainly used for the study of AD: the Alzheimer's Disease Neuroimaging Initiative (ADNI), the Australian Imaging, Biomarkers and Lifestyle (AIBL) and the Open Access Series of Imaging Studies (OASIS). In the following, we briefly describe these data sets and provide explanations on the diagnosis labels provided. Indeed, the diagnostic criteria of these studies differ, hence there is no strict equivalence between the labels of ADNI and AIBL, and those of OASIS.

In addition to these three data sets, the project Neuroimaging in Frontotemporal Dementia (NIFD) was used in chapter 5.

Alzheimer's Disease Neuroimaging Initiative (ADNI)

Part of the data used in this PhD were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database. The ADNI was launched in 2003 as a public-private partnership, led by Principal Investigator Michael W. Weiner, MD. The primary goal of ADNI has been to test whether serial magnetic resonance imaging (MRI), positron emission tomography (PET), other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimer's disease (AD). For up-to-date information, see `http://adni.loni.usc.edu`. The ADNI study is composed of 4 cohorts: ADNI-1, ADNI-GO, ADNI-2 and ADNI-3. These cohorts are dependant and longitudinal, meaning that one cohort may include the same patient more than once and that different cohorts may include the same patients. Diagnosis labels are given by a physician after a series of tests (Petersen et al., 2010). The existing labels are:

- AD (Alzheimer's disease): mildly demented patients,
- MCI (mild cognitive impairment): patients in the prodromal phase of AD,
- NC (normal controls): elderly control participants,
- SMC (significant memory concern): participants with cognitive complaints and no pathological neuropsychological findings. The designations SMC and subjective cognitive decline (SCD) are equivalently found in the literature.

The screening protocol of ADNI excludes participants with "any significant neurologic disease other than suspected incipient Alzheimer's disease", as well as those affected by psychiatric disorders (see ADNI clinical protocols for more information[1]). Then the study focuses on AD in its purest form, which may not be the most common cases in real life.

Since the ADNI-GO and ADNI-2 cohorts, new patients at the very beginning of the prodromal stage have been recruited (Aisen et al., 2010), hence the MCI label has been split into two labels:

- EMCI (early MCI): patients at the beginning of the prodromal phase,
- LMCI (late MCI): patients at the end of the prodromal phase (similar to the previous label MCI of ADNI-1).

This division is made on the basis of the score obtained on memory tasks corrected by the education level. However, both classes remain very similar, and they are fused in many studies under the MCI label.

Australian Imaging, Biomarkers and Lifestyle (AIBL)

Similarly to ADNI, the Australian Imaging, Biomarker & Lifestyle Flagship Study of Ageing seeks to discover which biomarkers, cognitive characteristics, and health and lifestyle

[1] `https://adni.loni.usc.edu/methods/documents/`

factors determine the development of AD. The AIBL project includes a longitudinal cohort of patients. Several modalities are present in the data set, such as clinical and imaging (MRI and PET) data, as well as the analysis of blood and CSF samples. As in ADNI, the diagnosis is given according to a series of clinical tests (Ellis et al., 2009, 2010) and the existing labels are AD, MCI and NC.

Open Access Series of Imaging Studies (OASIS)

This project includes three cohorts, OASIS-1, OASIS-2 and OASIS-3. The first cohort is only cross-sectional, whereas the other two are longitudinal. Available data is far more limited than in ADNI with only few clinical tests and imaging data (both MRI and PET only in OASIS-3). Diagnosis labels are given only based on the clinical dementia rating (CDR) scale (Marcus et al., 2007). Two labels can be found in the OASIS-1 data set:

- AD, which corresponds to patients with a non-null CDR score. This class gathers patients who would be spread between the MCI and AD classes in ADNI. A subdivision of this class is done based on the CDR, the scores of 0.5, 1, 2 and 3 representing very mild, mild, moderate and severe dementia, respectively.
- Control, which corresponds to patients with a CDR of zero. Unlike ADNI, some of the controls are younger than 55.

More information may be found at `http://www.oasis-brains.org`.

Neuroimaging in Frontotemporal Dementia (NIFD)

NIFD is the nickname for the frontotemporal lobar degeneration neuroimaging initiative (more information on LONI website https://ida.loni.usc.edu). It is a longitudinal study of variants of fronto-temporal dementia (FTD). It provides mainly MRI (FLAIR and T1w), along with some PET images. The main existing labels are:

- CN (control): elderly control participants,
- bvFTD (behavioural variant): variant of FTD with early behavioural and executive deficits,
- PNFA (progressive non fluent aphasia): variant of FTD with progressive deficits in speech, grammar, and word output,
- svFTD (semantic variant): variant of FTD with semantic knowledge and naming impairments.

Neuroimaging data

The literature on the phenotypic heterogeneity of AD describes subtypes according to cognitive abilities and patterns of biomarkers in the brain of the patients. The goal of the

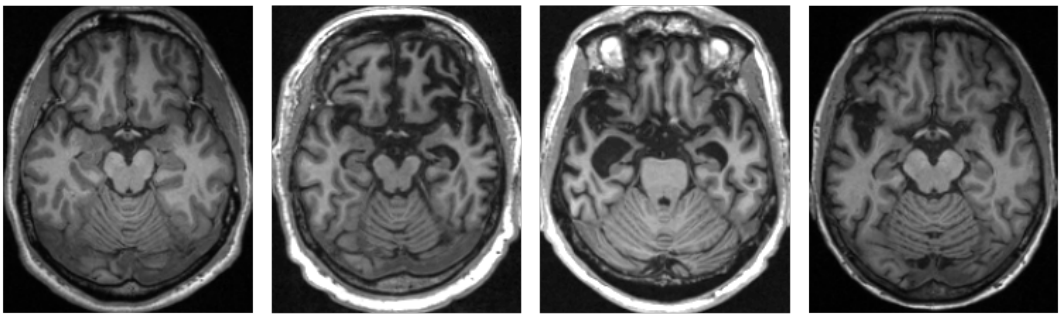


FIGURE 3: From left to right, T1w MRI of a control, bvFTD patient, svFTD patient and PNFA patient of NIFD.

PhD was to find subtypes according to neuroimaging data and validate the findings based on neuropsychological assessments. One reason not to use directly neuropsychological assessments was that they are less robust than neuroimaging as they show more variability. Indeed, as explained in ADNI guidelines[2]: "neuropsychological testing is not a mechanical process. The examiner encounters a wide range of emotional and physical problems that can interfere with testing, and the skill and judgment of the examiner often affect the subject's willingness to be tested and the effort he/she invests".

Several imaging modalities (illustrated in Figure 4) corresponding to AD biomarkers are available in the studied data base:

- amyloid-PET allows finding the location of senile plaques in the brain,
- FDG-PET allows finding hypometabolic regions (stage before neurodegeneration),
- structural MRI allows characterizing the neurodegeneration patterns.

We finally exclusively used T1w MRI, a type of structural MRI, as it is more commonly found in neuroimaging data sets.

MRI acquisition

Though structural MRI is more robust than neuropsychological tests, one should not forget that inhomogeneities between images exist, and that they may bias the results obtained. The possible sources of inhomogeneity are the following:

- the resolution of the image, which largely depends on the strength of the magnet (see appendix A),
- the machine itself,
- the sequence performed to acquire the image,
- correction steps that might be performed before the preprocessing pipeline.

The preprocessing steps performed on images tend to reduce these biases, but at the cost of a possible information loss.

[2] http://adni.loni.usc.edu/wp-content/uploads/2010/09/ADNI_GeneralProceduresManual.pdf

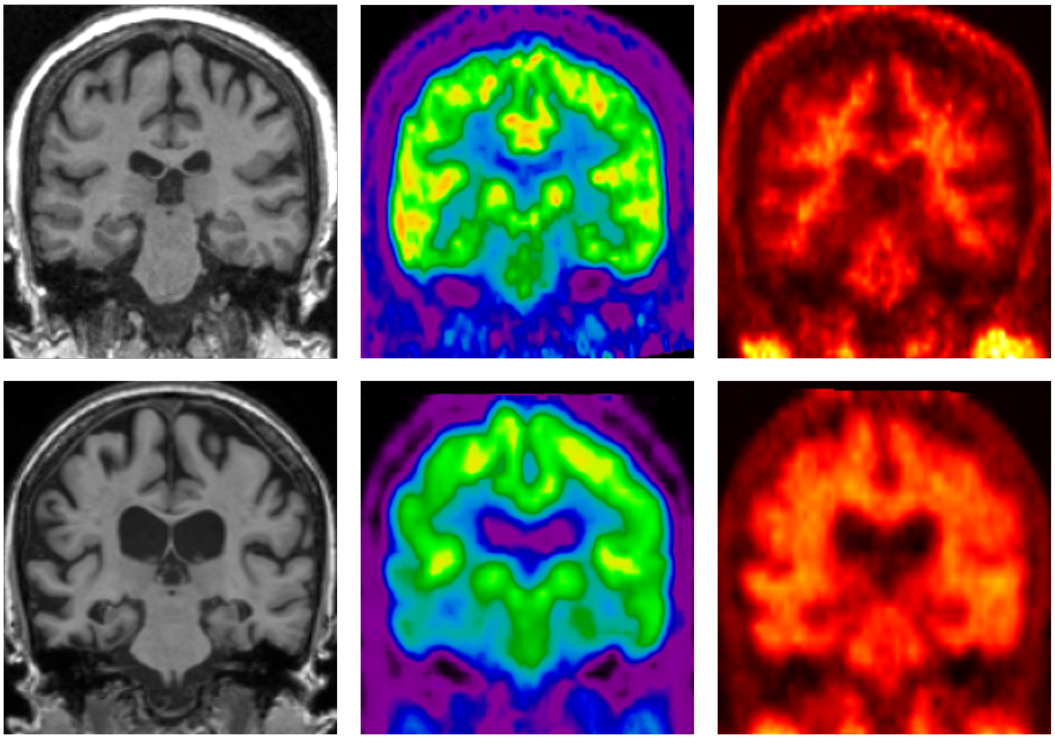


FIGURE 4: The first row corresponds to images of a CN participant and the second row to a demented patient of ADNI. From left to right, modalities are T1w MRI, FDG-PET and amyloid-PET.

MRI preprocessing

A proper image preprocessing procedure is a fundamental step to ensure a good classification performance, especially in the domain of MRI (Cuingnet et al., 2011; Lu and Weng, 2007; Uchida, 2013). Although CNNs have the potential to extract low-to-high level features from the raw images, the influence of image preprocessing remains to be clarified. We present here the essential steps for MR image processing in the context of AD classification.

Bias field correction

MR images can be corrupted by a low frequency and smooth signal caused by magnetic field inhomogeneities. This bias field induces variations in the intensity of the same tissue in different locations of the image, which deteriorates the performance of image analysis algorithms such as registration (Vovk et al., 2007). Several methods exist to correct these intensity inhomogeneities, two popular ones being the nonparametric nonuniformity intensity normalization (N3) algorithm (Sled et al., 1998), available for example in the Freesurfer software package[3], and the N4 algorithm (Tustison et al., 2010) implemented in ITK[4].

[3] http://surfer.nmr.mgh.harvard.edu/fswiki/recon-all
[4] http://hdl.handle.net/10380/3053

Intensity rescaling and standardization

MR images usually have different intensity ranges and the intensity distribution of the same tissue type may be different between two images, which might affect the subsequent image preprocessing steps. The first point can be dealt with by globally rescaling the image, for example between 0 and 1 using the minimum and maximum intensity values. Intensity standardization can also be achieved using techniques such as histogram matching (Madabhushi and Udupa, 2005).

Skull stripping

Extracranial tissues can be an obstacle for image analysis algorithms (Kalavathi and Prasath, 2016). A large number of methods have been developed for brain extraction, also called skull stripping, and many are implemented in software tools, such as the Brain Extraction Tool (BET) (Smith, 2002) available in FSL[5], or the Brain Surface Extractor (BSE) (Shattuck et al., 2001) available in BrainSuite[6]. These methods are often sensitive to the presence of noise and artifacts, which can result in over or under segmentation of the brain.

Image registration

Medical image registration consists of spatially aligning two or more images, either globally (rigid and affine registration) or locally (non-rigid registration), so that voxels in corresponding positions contain comparable information. A large number of software tools have been developed for MRI-based registration (Oliveira and Tavares, 2014). FLIRT[7] (Greve and Fischl, 2009; Jenkinson and Smith, 2001; Jenkinson et al., 2002) and FNIRT[8] (Andersson et al., 2007) are FSL tools dedicated to linear and non-linear registration, respectively. The Statistical Parametric Mapping (SPM) software package[9] and Advanced Normalization Tools[10] (ANTs) also offer solutions for both linear and non-linear registration (Ashburner and Friston, 2000; Avants et al., 2014; Friston et al., 1995).

Deep learning classification

Deep learning techniques allow learning a large variety of tasks by optimizing the weights of a neural network according to a loss which depends on the output of the network and possibly on other components (input, labels, weights...). This optimization is often a variant of the stochastic gradient descent algorithm, and is performed thanks to the successive applications of the network to inputs of a data set as large as possible. Contrary to other machine learning methods which may need some feature extraction and selection, deep learning methods take as input high-dimensional and minimally processed data.

[5] https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/BET/UserGuide
[6] http://brainsuite.org/processing/surfaceextraction/bse
[7] https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FLIRT
[8] https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FNIRT
[9] https://www.fil.ion.ucl.ac.uk/spm
[10] http://stnava.github.io/ANTs

Deep neural networks are composed of a succession of layers (i.e. functions) which take as input and compute feature maps (i.e. intermediate computation values). A single value of a feature map is a node.

During the training phase, the network updates its weights to make a series of inputs match with their corresponding target labels:

1. *Forward pass*: the network processes the input image to compute the output value.
2. *Loss computation*: the difference between the true labels and the output values is computed according to a criterion (cross-entropy, mean squared error...). This difference is called the loss, and should be as low as possible
3. *Backward pass*: for each learnable parameter of the network, the gradients with respect to the loss are computed.
4. *Weight update*: weights are updated according to the gradients and an optimizer rule (stochastic gradient descent, Adam, Adadelta...).

As a network is a composition of functions, the gradients of the weights of a layer l with respect to the loss can be easily obtained according to the values of the gradients according to the criterion in the following layers. This way of computing gradients layer per layer is called back-propagation.

In this PhD, deep neural networks were trained to differentiate CN participants from AD patients based on their 3D T1w MRI. This classification was performed with a particular type of networks: convolutional neural networks (CNN). These networks include mainly four types of layers:

- convolutional layers are well suited to find patterns in images. They are composed of a series of filters which are represented by a 3D matrix of weights. Their output feature map is the concatenation of the outputs of these filters. The first convolutional layers of the network learn local patterns in the input by browsing the image with their filters. Then these patterns are again assembled by the successive layers to obtain wider and wider patterns, with the benefit of much less parameters than fully-connected layers,
- pooling layers decrease the size of the image by computing one scalar output per pool of voxels according to a fixed rule. This dimension decrease is necessary as we want to extract a series of scalar values (for example probabilities of diagnosis statuses) from an image of about 1 million voxels,
- fully-connected layers link every node of the input feature maps to every node of the output feature map. This layer is very costly and is often put at the end of the network when the dimension is small enough to allow it,
- activation functions allow introducing non linearity in the network, as convolutions and fully-connected layers are actually linear functions.

For a more didactic presentation of the layers in a CNN network, please refer to the ClinicaDL tutorial[11].

Deep learning use has exploded with the launch of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC): a benchmark in object category classification and detection on hundreds of object categories and millions of images (Deng et al., 2009). CNNs significantly improved the performance obtained by non-deep learning methods, thus gaining popularity beyond the field of computer vision and into the medical fields (Zhang et al., 2020). However, deep learning success is highly correlated with the size of the data sets used, and medical data is difficult and expensive to gather. This is why the most crucial issue encountered in the field is overfitting, as the number of parameters of the networks is too large for the amount of training data.

This PhD also tackled this fundamental issue by studying methods to alleviate overfitting such as transfer learning and extensive MRI preprocessing (Chapter 3) and data augmentation (Chapter 5).

Contributions

The goal of this PhD was to provide reproducible and interpretable methods for Alzheimer's disease subtyping.

Indeed, the characterization of the heterogeneity of this disease is fundamental, as current treatments are not efficient enough to prevent dementia (they can only slightly delay it), and that this failure could actually be caused by a lack of consideration for the different phenotypes of the disease.

As explained in this introduction, clinical works already described this heterogeneity based on post-mortem material and the review of the literature. As these studies led to the definition of different subtypes (though some may overlap) and that no consensus was reached in the field, other studies are needed to explore the diversity of subtypes that may exist. Moreover, these studies rely on pre-existing clinical hypotheses, then agnostic approaches may bring additional information. This is why studies from the machine learning field tried to solve this issue (state-of-the art presented in Chapter 5). But these studies do not take enough into account the normal variability and may mix it with the heterogeneity of the disease. This is why I developed during my research internship at ARAMIS a first machine learning method for Alzheimer's disease subtyping that takes into account the normal variability. However, this method took as input highly-engineered features: gray matter probabilities in regions of a neuroanatomical atlas. Then results were biased towards the atlas chosen, and patterns across several neuroanatomical regions could not be correctly characterized. This is why I chose to develop a similar framework using deep learning methods.

Though the initial focus of the PhD was the discovery of AD subtypes, many contributions are rather in the field of the validation and interpretability of deep learning

[11] https://aramislab.paris.inria.fr/clinicadl/tuto/Notebooks-AD-DL/deep_learning.html#common-network-layers

methods. Indeed, I discovered that many results of the field could not be exploited because of methodological issues, then I decided to spend time on the development of open-source and reliable frameworks.

I first tried to find which deep learning method best suited our problem, and began a literature review. Unfortunately, the results of our field were contaminated by data leakage, and we could not draw any conclusion from them. This is why a benchmark was conducted to establish which components found in the literature (preprocessing techniques, MRI patching/slicing, transfer learning...) were actually the best in our context. These two findings led to the publication of a first article as joint first author (Wen et al., 2020). The knowledge acquired from this first study also allowed me to contribute to three other publications:

- a participation to the PAC 2019 challenge[12], which consisted in predicting brain age with different machine learning systems (Couvy-Duchesne et al., 2020). In this study I trained and inferred the age with convolutional neural networks closely collaborating with Benoît Martin,
- a literature review on the prediction of mild cognitive impairment using machine learning (minor contribution) (Ansart et al., 2021),
- a literature review of deep learning methods for brain disorders (Burgos et al., 2020). More precisely in this review I contributed in the sections on brain age, identification of biomarkers in EEG signals, cancer detection and diagnosis, disease prediction, and interpretability.

The second component of this PhD is on deep learning interpretability, as it should help in identifying the patterns of AD subtypes. A first analysis of the literature highlighted that a large amount of methods exist, and that their robustness has not been assessed. This is why I carefully checked the robustness of the method I used to interpret our CNN. But after assessing its robustness, I observed that problems came from the CNN training that led to the identification of different patterns between two different trainings of the same network. This observation led to a conference article (Thibeau-Sutre et al., 2020) that raises awareness on this lack of robustness, which is not easily noticeable. These considerations also led to the writing of a literature review on deep learning interpretability in the context of brain disorders (this work will be submitted as a chapter of the book "Machine Learning for Brain Disorders" edited by Olivier Colliot).

Thirdly, the first goal of the PhD, characterizing Alzheimer's disease subtypes with neuroimaging data, was addressed on a work on synthetic data. In this study, a CNN was trained to differentiate a normal distribution from a heterogeneous abnormal one. The different abnormality patterns could then be retrieved thanks to a basic interpretability method. However, when using a small amount of data to mimic the size of real neuroimaging data sets, the patterns could not be easily recovered. A successful solution was to use data augmentation to artificially increase the size of the data set. As this finding

[12] https://web.archive.org/web/20200214101600/https://www.photon-ai.com/pac2019

did not translate to real data with simple data augmentation methods, I worked with another PhD student which developed a data augmentation method with variational autoencoders by validating his method on neuroimaging data. This work was submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence.

Finally one major contribution of this PhD is the contribution to open-source reproducible frameworks for the processing of neuroimaging data, Clinica and ClinicaDL. Clinica is a software for neuroimaging formatting and preprocessing. My contributions to Clinica (Routier et al., 2021) concerned the iotools, which ease the use of the clinical and imaging data of data sets following the Brain Image Data Structure (BIDS) standard, and also the BIDS converter of ADNI. In particular, I adapted these tools for a study led by Federica Cacciamani of preclinical AD which will be submitted by the end of the PhD. I was lead programmer of ClinicaDL, the deep learning extension of Clinica that provides many tools such as network training, inference on new data, interpretability or hyperparameters search. The tools of ClinicaDL were built following (Wen et al., 2020), which explained that most deep learning studies of our field were contaminated with data leakage. Then its main goal is to provide a safe environment for deep learning users to avoid common pitfalls found in the literature and enhance reproducibility. This work led to the submission of an article to Computer Methods and Programs in Biomedicine.

Outline of the manuscript

The methodology developed during the PhD consists in clustering subtypes thanks to the extraction of the patterns learnt by a network to differentiate AD patients from CN participants. ClinicaDL, an open-source Python library was developed to allow the reproducibility of the results found during this PhD.

The next two chapters are states of the art on the application of deep learning methods to neuroimaging data (Chapter 1) and the interpretability of deep learning methods (Chapter 2). As chapter 1 highlighted the presence of biases in the literature, chapter 3 intends to give a fair benchmark of deep learning techniques classifying AD from CN participants with MRI. Then chapter 4 assesses the robustness of CNNs in discovering patterns of atrophy, and concluded that the training of CNNs is not robust, which is a major issue in the application of the method. Next, chapter 5 tries to implement the method to detect Alzheimer's disease subtypes. Despite a success on synthetic data thanks to data augmentation, the findings cannot be translated easily to real cases. Finally, chapter 6 presents ClinicaDL, a major contribution of the PhD.

Chapter 1

State of the art – Application of deep learning methods to Alzheimer's disease classification using T1-weighted magnetic resonance imaging

---

This chapter is a part of an article published in *Medical Image Analysis*:

- **Title:** Convolutional neural networks for classification of Alzheimer's disease: Overview and reproducible evaluation
- **Authors:** Junhao Wen[†], Elina Thibeau-Sutre[†], Mauricio Diaz-Melo, Jorge Samper-González, Alexandre Routier, Simona Bottani, Didier Dormont, Stanley Durrleman, Ninon Burgos, Olivier Colliot, for the Alzheimer's Disease Neuroimaging Initiative and the Australian Imaging Biomarkers and Lifestyle flagship study of ageing.
- **DOI:** doi:10.1016/j.media.2020.101694
- **Contributions:** Literature review, 3D subject-level experiments, code programming and refactoring, manuscript redaction

[†] denotes shared first authorship

In this section we review and summarize the different studies using convolutional neural networks (CNNs) and anatomical magnetic resonance imaging (MRI) for Alzheimer's disease (AD) classification. In particular, we review their validation procedures and the possible presence of data leakage. Our methodology for the literature review is described in Section 1.1.

Before describing the studies we selected, we briefly describe other studies that were kept in our bibliography but that are out of our scope (Section 1.2) to give the reader a

better idea of methods developped in our field. Depending on the disease stage that is studied, different classification experiments can be performed. We present the main tasks considered in the literature in Section 1.3. We found that a substantial proportion of the studies performed a biased evaluation of results due to the presence of data leakage. These issues are discussed in Section 1.4. Finally, we review the 32 studies that used end-to-end CNNs on image data, the main focus of this work (Section 1.5).

1.1 Literature review methodology

We searched PubMed and Scopus for articles published up to the time of the search (15th of January 2019). Our query contains words linked to four different concepts: Alzheimer's disease, classification, deep learning and neuroimaging. The words matching these concepts were identified in the abstracts and titles of the articles of a first bibliography done on Google Scholar. In Scopus a restriction was added to remove the articles linked to electroencephalography that appeared with our query and were out of our scope. This restriction was not applied in PubMed as it concerns only a few articles (less than 10). The line of the query linked to the neuroimaging concept was extended to all fields, as some authors do not mention at all in the title, abstract or keywords the modalities that they employed.

Scopus query:
TITLE-ABS-KEY (alzheimer's OR alzheimer OR "Mild Cognitive Impairment")
AND
TITLE-ABS-KEY (classification OR diagnosis OR identification OR detection OR recognition)
AND
TITLE-ABS-KEY (cnn OR "Convolutional Network" OR "Deep Learning" OR "Neural Network" OR autoencoder OR gan)
AND
ALL (mri OR "Magnetic Resonance Imaging" OR "Structural Magnetic Resonance Imaging" OR neuroimaging OR brain-imaging)
AND NOT
TITLE-ABS-KEY (eeg OR eegs OR electroencephalogram OR electroencephalographic)

PubMed query:
(alzheimer's [Title/Abstract] OR alzheimer [Title/Abstract] OR "Mild Cognitive Impairment" [Title/Abstract])
AND
(cnn OR "Convolutional Network" [Title/Abstract] OR "Deep Learning" [Title/Abstract] OR "Neural Network" [Title/Abstract] OR autoencoder [Title/Abstract] OR gan [Title/Abstract])
AND
(classification [Title/Abstract] OR diagnosis [Title/Abstract] OR identification [Title/Abstract] OR detection [Title/Abstract] OR recognition [Title/Abstract])

AND
(mri OR "Magnetic Resonance Imaging" OR "Structural Magnetic Resonance Imaging" OR neuroimaging OR brain-imaging)

391 records were found with Scopus and 80 records were found with PubMed. After merging the two sets and removing duplicates, 406 records were identified. Before filtering the result, we removed from this list 10 conference proceedings books and 1 non-english article. We finally ended with 395 records to filter. Once identified, all records were filtered in a 3-step process. We selected the records based on the abstract, the publication type and the content. The full process is displayed in Figure 1.1.

1.1.1 Record screening based on the abstract

During this step, the abstracts of the articles were read to keep only the methods corresponding to the following criteria:

- use of anatomical MRI (when the modality was specified),
- classification of AD stages, then we excluded papers using deep learning to preprocess, segment or complete data, as well as the classification of different diseases or classification of different symptoms in AD population (depression, impulsive control disorders...),
- exclusion of animal models,
- exclusion of reviews.

We chose to exclude the 31 reviews of our set as none of them focused on our topic. We did not detail the reasons of the exclusion of the papers in the diagram as many papers cumulate several criteria of exclusion. After this screening phase, we were left with 124 records.

1.1.2 Record screening based on the type of publication

Our search on PubMed and Scopus comprises only peer-reviewed items. However, there is a different level of peer-review between conference papers and journal articles, hence we kept all journal articles and recent conference papers (published since 2017). We decided to not only restrict to journal articles because it would have reduced the number of items to 48. We decided to keep recent conference papers because we considered that if the older ones were not transformed into journal articles it may mean that their contributions were not sufficient. After this step, the set contained 93 items.

1.1.3 Record screening based on the article content

This step was mainly used to sort the papers between the different sections of our state-of-the art. We detected in this way papers that were out of the scope of our review (longitudinal

and multimodal studies, deep learning techniques other than CNN). We excluded only 22 papers because of i) use of another modality (1 paper); ii) duplicate content (2 papers); iii) lack of explanation on the method employed (7 papers); iv) no access to the content (12 papers). This step was reviewed by another member of the team to confirm the exclusions. In the end, our search resulted in 71 conference and journal articles, including 32 that are centered on our topic.

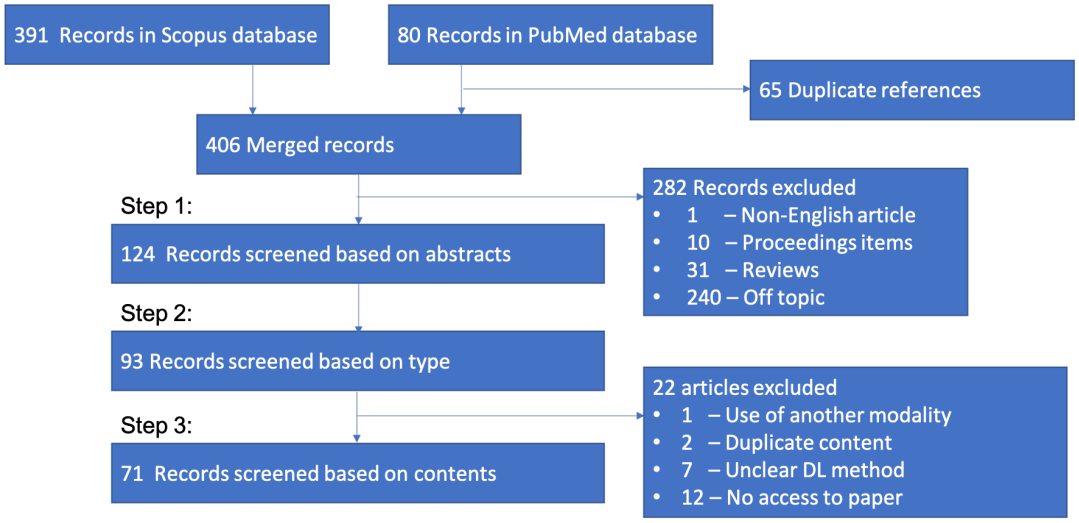


FIGURE 1.1: Diagram summarizing the bibliographic methodology

1.2 Other deep learning approaches for AD classification

Several studies found during our literature search are out of our scope: either CNNs were not used in an end-to-end manner or not applied to images, other network architectures were implemented, or the approach required longitudinal or multimodal data.

In several studies, the CNN is used as a feature extractor only and the classification is performed using either a random forest (Chaddad et al., 2018), SVM with linear or polynomial kernels and logistic regression (Çitak-ER et al., 2017), extreme ML (Lin et al., 2018), SVM with different kernels (Shen et al., 2018), or logistic regression and XGBoost (decision trees) (Shmulev et al., 2018). Only Shmulev et al., 2018 compared the results obtained with the CNN classification with those obtained with other classifiers based on features extracted by the CNN, and concluded that the latter is more efficient. Instead of being directly applied to the image, CNNs can be applied to pre-extracted features. This is the case of (Suk et al., 2017) where the CNN is applied to the outputs of several regression models performed between MRI-based features and clinical scores with different hyperparameters. CNNs can also be applied to non-Euclidean spaces, such as graphs of patients (Parisot et al., 2018) or the cortical surface (Mostapha et al., 2018).

Other architectures have been applied to anatomical MRI. Many studies used a variant of the multilayer perceptron composed of stacked FC layers (Amoroso et al., 2018; Baskar et al., 2018; Cárdenas-Peña et al., 2016, 2017; Dolph et al., 2017; Gorji and Haddadnia, 2015; Gutiérrez-Becker and Wachinger, 2018; Jha et al., 2017; Lu et al., 2018; Mahanand et al., 2012;

Maitra and Chatterjee, 2006; Ning et al., 2018; Raut and Dalal, 2017; Shams-Baboli and Ezoji, 2017; Zhang et al., 2018; Zhou et al., 2019) or of a probabilistic neural network (Duraisamy et al., 2019; Mathew et al., 2018). In other studies, high-level representations of the features are extracted using both unsupervised structures, such as deep Boltzmann machine (Suk et al., 2014) or AE (Suk et al., 2015), and supervised deep polynomial networks (Shi et al., 2018); then an SVM is used for classification. Non-CNN architectures require extensive preprocessing as they have to be applied to imaging features such as cortical thickness, shapes, or texture, and regional features. Moreover, feature selection or embedding is also often required (Amoroso et al., 2018; Dolph et al., 2017; Jha et al., 2017; Lu et al., 2018; Mahanand et al., 2012; Mathew et al., 2018; Suk et al., 2014, 2015) to further reduce dimensionality.

Deep learning-based classification approaches are not limited to cross-sectional anatomical MRI. Longitudinal studies exploit information extracted from several time points of the same subject. A specific structure, the recurrent neural network, has been used to study the temporal correlation between images (Bhagwat et al., 2018; Cui et al., 2018; Wang et al., 2018b). Several studies exploit multi-modal data (Aderghal et al., 2018; Cheng and Liu, 2017; Esmaeilzadeh et al., 2018; Li et al., 2015; Liu et al., 2016; Liu et al., 2018b; Liu et al., 2018d; Liu et al., 2015; Lu et al., 2018; Ning et al., 2018; Ortiz et al., 2016; Qiu et al., 2018; Raut and Dalal, 2017; Senanayake et al., 2018; Shi et al., 2018; Shmulev et al., 2018; Spasov et al., 2018; Suk et al., 2014; Thung et al., 2017; Vu et al., 2017, 2018; Zhou et al., 2017, 2019), such as multiple imaging modalities (PET and diffusion tensor imaging), demographic data, genetics, clinical scores, or cerebrospinal fluid biomarkers. Note that multimodal studies that also reported results with MRI only (Aderghal et al., 2018; Cheng and Liu, 2017; Liu et al., 2018b; Qiu et al., 2018; Senanayake et al., 2018; Shmulev et al., 2018; Vu et al., 2017, 2018) are displayed in Tables 1.1 and 1.2. Exploiting multiple time-points and/or modalities is expected to improve the classification performance. However, these studies can be limited by the small number of subjects having all the required time points and modalities.

1.3 Main classification tasks

Even though its clinical relevance is limited, differentiating patients with AD from cognitively normal subjects (CN), i.e. AD vs CN, is the most widely addressed task: 25 of the 32 studies presenting an end-to-end CNN framework report results with this task (Tables 1.1 and 1.2). Before the development of dementia, patients go through a phase called mild cognitive impairment (MCI) during which they have objective deficits, but not severe enough to result in dementia. Identifying this early stage of AD by differentiating MCI patients from CN subjects (MCI vs CN) is another task of interest, reported in nine studies. Patients with MCI may remain stable or subsequently progress to AD dementia or to another type of dementia. Distinguishing MCI subjects that will progress to AD (denoted as pMCI) from those who will remain stable (denoted as sMCI) would allow predicting the group of subjects that will likely develop the disease. This task (sMCI vs pMCI) has been

performed in seven studies. Other experiments performed in the 32 studies on which we focus include differentiating AD from MCI patients (AD vs MCI) and multi-class tasks.

1.4 Main causes of data leakage

Unbiased evaluation of classification algorithms is critical to assess their potential clinical value. A major source of bias is data leakage in training examples, which refers to the use of test data in any part of the training process (Kaufman et al., 2012). Data leakage can be difficult to detect for deep learning approaches as they can be complex and very flexible. We assessed the prevalence of data leakage among the papers described in Section 1.5 and analyzed its causes. The articles were labeled into three categories: i) Clear when data leakage was explicitly witnessed; ii) Unclear when no sufficient explanation was offered and iii) None detected. The labels are given in the last column of Tables 1.1 and 1.2. They were further categorized according to the cause of data leakage. Four main causes were identified:

1. Wrong data split. Not splitting the data set at the subject-level when defining the training, validation and test sets can result in data from the same subject to appear in several sets. This problem can occur when patches or slices are extracted from a 3D image, or when images of the same subject are available at multiple time points. (Bäckström et al., 2018) showed that, using a longitudinal data set, a biased data set split (at the image level) can result in an accuracy increase of 8 percent points compared to an unbiased split (at the subject-level).

2. Late split. Procedures such as data augmentation, feature selection or autoencoder (AE) pre-training must never use the test set and thus be performed after the training/validation/test split to avoid biasing the results. For example, if data augmentation is performed before isolating the test data from the training/validation data, then images generated from the same original image may be found in both sets, leading to a problem similar to the wrong data split.

3. Biased transfer learning. Transfer learning can result in data leakage when the source and target domains overlap, for example when a network pre-trained on the AD vs CN task is used to initialize a network for the MCI vs CN task and that the CN subjects in the training or validation sets of the source task (AD vs CN) are also in the test set of the target task (MCI vs CN).

4. Absence of an independent test set. The test set should only be used to evaluate the final performance of the classifier, not to choose the training hyperparameters (e.g. learning rate) of the model. A separate validation set must be used beforehand for hyperparameter optimization.

Note that we did not consider data leakage occurring when designing the network architecture, possibly chosen thanks to successive evaluations on the test set, as the large majority of the studies does not explicit this step. All these data leakage causes may not

have the same impact on data performance. For instance, it is likely that a wrong data split in a longitudinal data set or at the slice-level is more damaging than a late split for AE pre-training.

1.5 Classification of AD with end-to-end CNNs

This section focuses on CNNs applied to an Euclidean space (2D or 3D images) being the only machine learning component of an end-to-end framework (from the input to the final label). A summary of these studies can be found in Tables 1.1 and 1.2. The tables indicate whether data leakage was potentially present, which could have biased the performance upwards. We categorized studies according to the type of input of the network: i) 2D slice-level, ii) 3D patch-level, iii) ROI-based and iv) 3D subject-level.

1.5.1 2D slice-level CNN

Several studies used 2D CNNs with input composed of the set of 2D slices extracted from the 3D MRI volume (Farooq et al., 2017; Gunawardena et al., 2017; Hon and Khan, 2017; Islam and Zhang, 2017, 2018; Qiu et al., 2018; Taqi et al., 2018; Valliani and Soni, 2017; Wang et al., 2018a; Wang et al., 2017b; Wu et al., 2018). An advantage of this approach is that existing CNNs which had huge success for natural image classification, e.g. ResNet (He et al., 2016) and VGGNet (Simonyan and Zisserman, 2015), can be easily borrowed and used in a transfer learning fashion. Another advantage is the increased number of training samples as many slices can be extracted from a single 3D image.

In this subsection of the bibliography, we found only one study in which neither data leakage was detected nor biased metrics were used (Valliani and Soni, 2017). They used a single axial slice per subject (taken in the middle of the 3D volume) to compare the ResNet (He et al., 2016) to an original CNN with only one convolutional layer and two fully connected (FC) layers. They studied the impact of both transfer learning, by initializing their networks with models trained on ImageNet, and data augmentation with affine transformations. They conclude that the ResNet architecture is more efficient than their baseline CNN and that pre-training and data augmentation improve the accuracy of the ResNet architecture.

In all other studies, we detected a problem in the evaluation: either data leakage was present (or at least suspected) (Farooq et al., 2017; Gunawardena et al., 2017; Hon and Khan, 2017; Islam and Zhang, 2017; Taqi et al., 2018; Wang et al., 2018a; Wang et al., 2017b; Wu et al., 2018) or an imbalanced metric was computed on a severely imbalanced data set (one class is less than half of the other) (Islam and Zhang, 2018; Qiu et al., 2018). These studies differ in terms of slice selection: i) one study used all slices of a given plane (except the very first and last ones that are not informative) (Farooq et al., 2017); ii) other studies selected several slices using an automatic (Hon and Khan, 2017; Wu et al., 2018) or manual criterion (Qiu et al., 2018); iii) one study used only one slice (Wang et al., 2018a). Working with several slices implies to fuse the classifications obtained at the slice-level to obtain a classification at the subject-level. Only one study (Qiu et al., 2018) explained how they

Study	Performance					Approach	Data leakage
	AD vs CN	sMCI vs pMCI	MCI vs CN	AD vs MCI	Multi-class		
Aderghal et al., 2017a	ACC=0.84	–	ACC=0.65	ACC=0.67†	–	ROI-based	None detected
Aderghal et al., 2018	BA=0.90	–	BA=0.73	BA=0.83	–	ROI-based	None detected
Bäckström et al., 2018*	ACC=0.90	–	–	–	–	3D subject-level	None detected
Cheng et al., 2017	ACC=0.87	–	–	–	–	3D patch-level	None detected
Cheng and Liu, 2017	ACC=0.85	–	–	–	–	3D subject-level	None detected
Islam and Zhang, 2018	–	–	–	–	ACC=0.93[1]†	2D slice-level	None detected
Korolev et al., 2017	ACC=0.80	–	–	–	–	3D subject-level	None detected
Li et al., 2017	ACC=0.88	–	–	–	–	3D subject-level	None detected
Li et al., 2018	ACC=0.90	–	ACC=0.74†	–	–	3D patch-level	None detected
Lian et al., 2018	ACC=0.90	ACC=0.80†	–	–	–	3D patch-level	None detected
Liu et al., 2018e	ACC=0.91	ACC=0.78†	–	–	–	3D patch-level	None detected
Liu et al., 2018c	ACC=0.91	–	–	–	–	3D patch-level	None detected
Qiu et al., 2018	–	–	ACC=0.83†	–	–	2D slice-level	None detected
Senanayake et al., 2018	ACC=0.76	–	ACC=0.75	ACC=0.76	–	3D subject-level	None detected
Shmulev et al., 2018	–	ACC=0.62	–	–	–	3D subject-level	None detected
Valliani and Soni, 2017	ACC=0.81	–	–	–	ACC=0.57[2]	2D slice-level	None detected

TABLE 1.1: Summary of the studies without data leakage performing classification of AD using CNNs on anatomical MRI.

* In (Bäckström et al., 2018), data leakage was introduced on purpose to study its influence, which explains its presence in both categories.
† Use of accuracy on a severely imbalanced dataset (one class is less than half of the other), leading to an over-optimistic estimation of performance.
[1]CN vs mild vs moderate vs severe
[2]AD vs MCI vs CN
[3]AD vs LMCI vs EMCI vs CN
[4]sMCI vs pMCI vs CN
ACC: accuracy; BA: balanced accuracy.

Study	Performance					Approach	Data leakage (type)
	AD vs CN	sMCI vs pMCI	MCI vs CN	AD vs MCI	Multi-class		
Aderghal et al., 2017b	ACC=0.91	–	ACC=0.66	ACC=0.70	–	ROI-based	Unclear (b,c)
Basaia et al., 2019	BA=0.99	BA=0.75	–	–	–	3D subject-level	Unclear (b)
Hon and Khan, 2017	ACC=0.96	–	–	–	–	2D slice-level	Unclear (a,c)
Hosseini Asl et al., 2018	ACC=0.99	–	ACC=0.94	ACC=1.00	ACC=0.95[2]	3D subject-level	Unclear (a)
Islam and Zhang, 2017	–	–	–	–	ACC=0.74[1]†	2D slice-level	Unclear (b,c)
Lin et al., 2018	ACC=0.89	ACC=0.73	–	–	–	ROI-based	Unclear (b)
Liu et al., 2018b	ACC=0.85	ACC=0.74	–	–	–	3D patch-level	Unclear (d)
Taqi et al., 2018	ACC=1.00	–	–	–	–	2D slice-level	Unclear (b)
Vu et al., 2017	ACC=0.85	–	–	–	–	3D subject-level	Unclear (a)
Wang et al., 2018a	ACC=0.98	–	–	–	–	2D slice-level	Unclear (b)
Bäckström et al., 2018*	ACC=0.99	–	–	–	–	3D subject-level	Clear (a)
Farooq et al., 2017	–	–	–	–	ACC=0.99[3]†	2D slice-level	Clear (a,c)
Gunawardena et al., 2017	–	–	–	–	ACC=0.96[2]	3D subject-level	Clear (a,b)
Vu et al., 2018	ACC=0.86	–	ACC=0.86	ACC=0.77	ACC=0.80[2]	3D subject-level	Clear (a,c)
Wang et al., 2017b	–	–	ACC=0.91	–	–	2D slice-level	Clear (a,c)
Wang et al., 2019	ACC=0.99	–	ACC=0.98	ACC=0.94	ACC=0.97[2]	3D subject-level	Clear (b)
Wu et al., 2018	–	–	–	–	ACC=0.95[4]†	2D slice-level	Clear (a,b)

TABLE 1.2: Summary of the studies with potential data leakage performing classification of AD using CNNs on anatomical MRI.

Types of data leakage: a: wrong dataset split; b: absence of independent test set; c: late split; d: biased transfer learning (see Section 1.4).
* In (Bäckström et al., 2018), data leakage was introduced on purpose to study its influence, which explains its presence in both categories.
† Use of accuracy on a severely imbalanced dataset (one class is less than half of the other), leading to an over-optimistic estimation of performance.
[1]CN vs mild vs moderate vs severe
[2]AD vs MCI vs CN
[3]AD vs LMCI vs EMCI vs CN
[4]sMCI vs pMCI vs CN
ACC: accuracy; BA: balanced accuracy.

performed this fusion. Other studies did not implement fusion and reported the slice-level accuracy (Farooq et al., 2017; Gunawardena et al., 2017; Hon and Khan, 2017; Wang et al., 2017b; Wu et al., 2018) or it is unclear if the accuracy was computed at the slice- or subject-level (Islam and Zhang, 2017, 2018; Taqi et al., 2018).

The main limitation of the 2D slice-level approach is that MRI is 3-dimensional, whereas the 2D convolutional filters analyze all slices of a subject independently. Moreover, there are many ways to select slices that are used as input (as all of them may not be informative), and slice-level accuracy and subject-level accuracy are often confused.

1.5.2 3D patch-level CNN

To compensate for the absence of 3D information in the 2D slice-level approach, some studies focused on the 3D patch-level classification (see Tables 1.1 and 1.2). In these frameworks, the input is composed of a set of 3D patches extracted from an image. In principle, this could result, as in the 2D slice-level approach, in a larger sample size, since the number of samples would be the number of patches (and not the number of subjects). Additional advantages of patches are the lower memory usage, which may be useful when one has limited resources, and the lower number of parameters to learn. However, these potential advantages are not fully exploited in the surveyed papers as they trained one network for each patch.

Two studies (Cheng et al., 2017; Liu et al., 2018b) used very large patches. Specifically, they extracted 27 overlapping 3D patches of size 50×41×40 voxels covering the whole volume of the MR image (100×81×80 voxels). They individually trained 27 convolutional networks (one per patch) comprising four convolutional layers and two FC layers. Then, an ensemble CNN was trained to provide a decision at the subject level. This ensemble CNN is partly initialized with the weights of the previously trained CNNs. Liu et al., 2018b used exactly the same architecture as Cheng et al., 2017 and enriched it with a fusion of PET and MRI inputs. They also gave the results obtained using the MRI modality only, which is the result reported in Tables 1.1 and 1.2.

Li et al., 2018 used smaller patches (32×32×32). By decreasing the size of the patches, they had to take into account a possible discrepancy between patches taken at the same coordinates for different subjects. To avoid this dissimilarity between subjects without performing a non-linear registration, they clustered their patches using k-means. Then they trained one CNN per cluster, and assembled the features obtained at the cluster-level in a similar way to Cheng et al., 2017; Liu et al., 2018b.

The following three studies (Lian et al., 2018; Liu et al., 2018c,e) used even smaller patches (19×19×19). Only a subset of patches, chosen based on anatomical landmarks, are used. These anatomical landmarks are found in a supervised manner via a group comparison between AD and CN subjects. This method requires a non-linear registration to build the correspondence between voxels of different subjects. Similarly to other studies, in (Liu et al., 2018c), one CNN is pre-trained for each patch and the outputs are fused to obtain the diagnosis of a subject. The approach of Liu et al., 2018e is slightly different as they consider that a patch cannot be labelled with a diagnosis, hence they do not train

one CNN per patch individually before ensemble learning, but train the ensemble network from scratch. Finally, Lian et al., 2018 proposed a weakly-supervised guidance: the loss of the network is based on the final classification scores at the subject level as well as the intermediate classification done on the patch and region level.

There are far less data leakage problems in this section, with only a doubt about the validity of the transfer learning between the AD vs CN and MCI vs CN tasks in (Liu et al., 2018b) because of a lack of explanations. Nevertheless this has no impact on the result of the AD vs CN task for which we did not detect any problem of data leakage. As for 2D-slice level in which a selection of slices must be made, one must choose the size and stride of patches. The choice of these hyperparameters will depend on the MRI preprocessing (e.g. a non-linear registration is likely needed for smaller patches). Nevertheless, note that the impact of these hyperparameters has been studied in the pre-cited studies (which has not been done for the 2D slice-level approaches). The main drawback of these approaches is the complexity of the framework: one network is trained for each patch position and these networks are successively fused and retrained at different levels of representation (region-level, subject-level).

1.5.3 ROI-based CNN

3D patch-level methods use the whole MRI by slicing it into smaller inputs. However, most of these patches are not informative as they contain parts of the brain that are not affected by the disease. Methods based on regions of interest (ROI) overcome this issue by focusing on regions which are known to be informative. In this way, the complexity of the framework can be decreased as fewer inputs are used to train the networks. In all the following studies, the ROI chosen was the hippocampus, which is well-known to be affected early in AD (Dickerson et al., 2001; Salvatore et al., 2015; Schuff et al., 2009). Studies differ by the definition of the hippocampal ROI.

Aderghal et al., 2018; Aderghal et al., 2017a,b performed a linear registration and defined a 3D bounding box comprising all the voxels of the hippocampus according to a segmentation with the AAL atlas. These three studies used a "2D+ϵ approach" with patches made of three neighbouring 2D slices in the hippocampus. As they use only one or three patches per patient, they do not cover the entire region. The first study (Aderghal et al., 2017a) only uses the sagittal view and classifies one patch per patient. The architecture of the CNN is made of two convolutional layers associated with max pooling, and one FC layer. In the second study (Aderghal et al., 2017b), all the views (sagittal, coronal and axial) are used to generate patches. Then, three patches are generated per subject, and three networks are trained for each view and then fused. The last study from the same author (Aderghal et al., 2018) focuses on the transfer learning from anatomical MRI to diffusion MRI, which is out of our scope. In (Lin et al., 2018) a non-linear registration was performed to obtain a voxel correspondence between the subjects, and the voxels belonging to the hippocampus[1] were identified after a segmentation implemented with MALP-EM (Ledig

[1] In their original paper, this anatomical structure was called the "hippopotamus" (sic).

et al., 2015). 151 patches were extracted per image with sampling positions fixed during the experiments. Each of them was made of the concatenation of three 2D slices along the three possible planes (sagittal, coronal and axial) originated at one voxel belonging to the hippocampus.

The main drawback of this methodology is that it studies only one (or a few) regions while AD alterations span over multiple brain areas. However, it may reduce the risk of overfitting because the inputs are smaller (~3000 voxels in our bibliography) and fewer than in methods allowing patch combinations.

1.5.4 3D subject-level CNN

Recently, with the boost of high-performance computing resources, more studies used a 3D subject-level approach (see Tables 1.1 and 1.2). In this approach, the whole MRI is used at once and the classification is performed at the subject level. The advantage is that the spatial information is fully integrated.

Some studies readapted two classical architectures, ResNet (He et al., 2016) and VGGNet (Simonyan and Zisserman, 2015), to fit the whole MRI (Korolev et al., 2017; Shmulev et al., 2018). In both cases, the classification accuracies obtained with VGGNet and ResNet are equivalent, and their best accuracies are lower than that of other 3D subject-level approaches. Another study (Senanayake et al., 2018) used a set of complex modules from classical architectures such as ResNet and DenseNet (dilated convolutions, dense blocks and residual blocks), also without success.

Other studies defined original architectures (Bäckström et al., 2018; Basaia et al., 2019; Cheng and Liu, 2017; Hosseini Asl et al., 2018; Li et al., 2017; Vu et al., 2017, 2018; Wang et al., 2019). We detected data leakage in all studies except (Bäckström et al., 2018; Cheng and Liu, 2017; Li et al., 2017). Bäckström et al., 2018; Cheng and Liu, 2017 had a similar approach by training one network from scratch on augmented data. One crucial difference between these two studies is the preprocessing step: Bäckström et al., 2018 used non-linear registration whereas Cheng and Liu, 2017 performed no registration. Li et al., 2017 proposed a more complex framework fusing the results of a CNN and three networks pre-trained with an AE.

For the other studies using original architectures, we suspect data leakage (Basaia et al., 2019; Hosseini Asl et al., 2018; Vu et al., 2017, 2018; Wang et al., 2019), hence their performance cannot be fairly compared to the previous ones. However we noted that Hosseini Asl et al., 2018; Vu et al., 2017, 2018 studied the impact of pre-training with an AE, and concluded that it improved their results (accuracy increased from 5 to 10 percent points).

In the 3D-subject level approach, the number of samples is small compared to the number of parameters to optimize. Indeed, there is one sample per subject, typically a few hundreds to thousands of subjects in a data set, thus increasing the risk of overfitting.

1.6 Conclusion

A high number of the 32 selected studies presented a biased performance because of data leakage: 10 were labeled as Unclear because of lack of explanations, and 6 as Clear (we do not count here the study of Bäckström et al., 2018 as data leakage was done deliberately to study its impact). This means that about 50% of the surveyed studies could report biased results.

In addition to that problem, most studies are not comparable because the data sets used, subjects selected among them and preprocessing performed are different. Furthermore, these studies often do not motivate the choice of their architecture or hyperparameters. It might be that many of them have been tried (but not reported) thereby resulting in a biased performance on the test set. Finally, the code and key implementation details (such as hyperparameters values) are often not available, making them difficult if not impossible to reproduce.

Chapter 2

State of the art – Application of interpretability methods to brain disorders

---

This chapter will be submitted as a chapter of the book named *Machine Learning for Brain Disorders* and edited by Olivier Colliot. It will be published in the Neuromethods series.

- **Title:** Interpretability of Machine Learning applied to Brain Disorders
- **Authors:** Elina Thibeau-Sutre, Sasha Collin, Didier Dormont, Ninon Burgos, Olivier Colliot

---

2.1 Introduction

2.1.1 Need for interpretability

Many metrics have been developed to evaluate the performance of machine learning systems. In the case of supervised systems, these metrics take as input the ground truth and the outputs of the algorithm to evaluate its ability to reproduce a label given by a physician. However, the users (patients and clinicians) may want more information before relying on such systems. On which features is the model relying to compute the results? Are these features close to the way a clinician thinks? If not, why? This questioning coming from the actors of the medical field is justified, as errors in real life may lead to dramatic consequences. Physicians may thus not want to take the responsibility of the machine learning system failures... unless they trust it. And this trust cannot be built only based on a set of metrics evaluating the performance of the system. Indeed, various examples of machine learning systems taking correction decisions for the wrong reasons exist, e.g. (DeGrave et al., 2021; Fong and Vedaldi, 2017; Ribeiro et al., 2016). Thus, even though their performance is high, they may be unreliable and, for instance, not generalize well to slightly different data sets. A way of preventing this issue is to interpret the model with an appropriate method whose output will highlight the reasons why a model took its decision.

In Ribeiro et al., 2016, the authors show a now classical case of a system that correctly classifies images for wrong reasons. They purposely designed a biased data set in which wolves always are in a snow environment whereas huskies are not. Then, they trained a classifier to differentiate wolves from huskies: this classifier had a good accuracy, but classified as wolves huskies with a snowy background, and as huskies wolves that were not in the snow. Using an interpretability method, they further highlighted that the classifier was looking at the background and not at the animal (see Figure 2.1).

(a) Husky classified as wolf (b) Explanation

FIGURE 2.1: Example of an interpretability method highlighting why a network took the wrong decision. The explained classifier was trained on the binary task "Husky" vs "Wolf". The pixels used by the model are actually in the background and highlight the snow. Reproduced from (Ribeiro et al., 2016).

Another study by Fong and Vedaldi, 2017 detected a bias in ImageNet (a widely used data set of natural images) as the interpretation of images with the label "chocolate sauce" highlighted the importance of the spoon. Indeed, ImageNet "chocolate sauce" images often contained spoons, leading to a spurious correlation. There are also examples of similar problems in medical applications. For instance, a recent paper by DeGrave et al., 2021 showed with interpretability methods that some deep learning systems detecting COVID-19 from chest radiographs actually relied on confounding factors rather than on the actual pathological features. Indeed, their model trained on public data sets widely used by many studies focused on other regions than the lungs to evaluate the COVID-19 status (edges, diaphragm and cardiac silhouette).

2.1.2 How to interpret models

According to Lipton, 2018, model interpretability can be broken down into two categories: transparency and post-hoc explanations.

A model can be considered as transparent when it (or all parts of it) can be fully understood as such, or when the learning process is understandable. A natural and common candidate that fits, at first sight, these criteria is the linear regression algorithm, where coefficients are usually seen as the individual contributions of the input features, but also decision trees, where model predictions can be broken down into a series of understandable operations. However, one may need to be cautious about the real interpretability allowed by these models. Indeed, in some cases a feature may have not been kept by the model, but this does not mean that it is not associated with the target. This

is the case for example for sparse models like LASSO, but also multiple non-regularized linear regressions. Moreover, features given as input to transparent models are often highly-engineered, and choices made before the training step (preprocessing, feature selection) may also hurt the transparency of the whole framework.

The second category of interpretability methods, post-hoc interpretations, allows dealing with non-transparent models. Xie et al., 2020 proposed a taxonomy in three categories: *visualization* methods consist in extracting an attribution map of the same size as the input whose intensities allow knowing where the algorithm focused its attention, *distillation* approaches consist in reproducing the behavior of a black-box model with a transparent one, and *intrinsic* strategies include interpretability components within the framework, which are trained along with the main task (for example, a classification). In this work we focus on this second category of methods (post-hoc), and proposed a new taxonomy including other methods of interpretation. Post-hoc interpretability is the category the most used nowadays, as it allows interpreting deep learning methods that were recently adapted to neuroimaging studies.

2.1.3 Chapter content and outline

This chapter focuses on methods developed to interpret non-transparent machine learning systems, mainly deep learning systems, computing high-level information from high-dimensional inputs (i.e., classification or regression). The interpretability of other frameworks (in particular generative models such as variational autoencoders or generative adversarial networks) is not covered as there are not enough studies addressing them. It may be because high-dimensional outputs (such as images) are easier to interpret "as such", whereas small dimensional outputs (such as scalars) are less transparent.

Most interpretability methods presented in this chapter produce an attribution map: an array with the same dimensions as that of the input (up to a resizing), that can be overlaid on top of the input in order to exhibit an explanation of the model prediction. In the literature, many different terms may coexist to name this output such as saliency map, interpretation map or heatmap. To avoid misunderstandings, in the following we will only use the term "attribution map".

The chapter is organized as follows. Section 2.2 presents the most commonly used interpretability methods proposed for computer vision, independently of medical applications. It also describes metrics developed to evaluate the reliability of interpretability methods. Then, section 2.3 details their application to the neuroimaging domain. Finally, section 2.4 discusses current limitations of the domain, presents benchmarks conducted in the neuroimaging field and gives some advice to the readers who would like to interpret their own models.

A brief description of the diseases mentioned in the present chapter are provided in Appendix C.

2.2 Theoretical framework of interpretability methods

This section presents the main interpretability methods proposed in the domain of computer vision. We restrict ourselves to the methods that have been applied to the neuroimaging domain (the applications themselves being presented in Section 2.3). The outline of this section is largely inspired from the one proposed by Xie et al., 2020:

1. **weight visualization** consists in directly visualizing weights learned by the model, which is natural for linear models but quite less informative for deep learning networks,

2. **feature map visualization** consists in displaying intermediate results produced by a deep learning network to better understand its operation principle,

3. **back-propagation methods** are back-propagating a signal through the machine learning system from the output node of interest o_c to the level of the input to produce an attribution map,

4. **perturbation methods** evaluate the difference in performance between an original input and its locally perturbed versions to infer which parts of the input is relevant for the machine learning system,

5. **distillation** approximates the behavior of a black-box model with a more transparent one, and then draw conclusions from this new model,

6. **intrinsic** includes the only methods of this chapter that are not post-hoc explanations: in this case, interpretability is obtained thanks to components of the framework that are trained at the same time as the model.

Finally, a section is dedicated to the metrics used to evaluate different properties (for example reliability or human-intelligibility) of the attribution maps produced.

We caution readers that this taxonomy is not perfect: some methods may belong to several categories (for example LIME and SHAP could be linked either to perturbation or distillation methods). Moreover, interpretability is still an active research field, then some categories may (dis)appear or be fused in the future.

The interpretability methods were (most of the time) originally proposed to interpret machine learning frameworks learning to perform a classification. In this case, the network outputs an array of size C, corresponding to the number of different labels existing in the data set, and the goal is to know how the output node corresponding to a particular class c interacts with the input or with other parts of the network. However, these techniques can be easily extended to other tasks: for example for a regression task, we will just have to consider the output node containing the continuous variable learned by the network. Moreover, some methods do not depend on the nature of the algorithm (e.g. standard-perturbation or LIME) and can be applied to any machine learning algorithm.

2.2.1 Weight visualization

One of the most intuitive way to understand the result of a training task is to directly visualize the weights learned by the algorithm. This method is really simple, as it does not require further processing. However, though it can make sense for linear models, it is not very informative for most networks unless they are specially designed for this interpretation.

This is the case for AlexNet (Krizhevsky et al., 2012), a convolutional neural network (CNN) trained on natural images (ImageNet). In this network the size of the kernels in the first layer is large enough (11×11) to distinguish patterns of interest. Moreover, as the three channels in the first layer correspond to the three color channels of the images (red, green and blue), the values of the kernels can also be represented in terms of colors (this is not the case for hidden layers, in which the meaning of the channels is lost). The 96 kernels of the network were illustrated in the original article as in Figure 2.2. However, for hidden layers, this kind of interpretation may be misleading as non-linearity activation layers are added between the convolutions or fully-connected layers, this is why they only visualized the weights of the first layer.

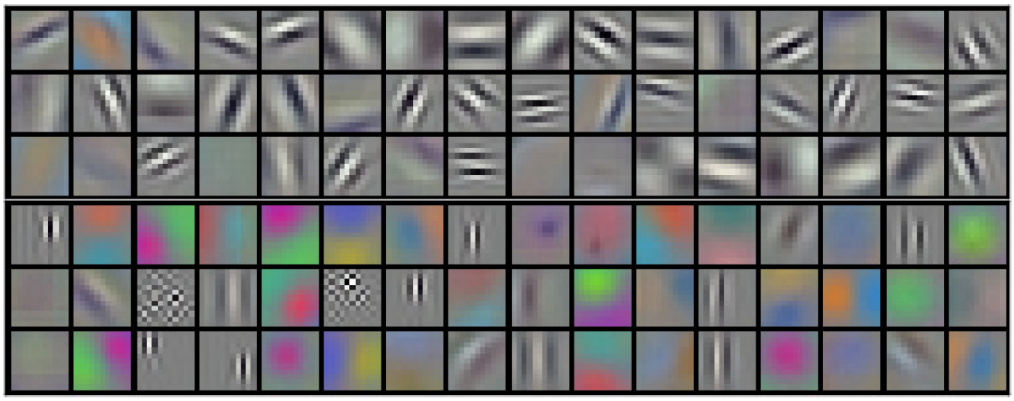


FIGURE 2.2: 96 convolutional kernels of size 3@11×11 learned by the first convolutional layer on the 3@224×224 input images by AlexNet.
Reproduced from (Krizhevsky et al., 2012).

To understand the weight visualization in hidden layers of a network, Voss et al., 2021 proposed to add some context to the input and the output channels. This way they enriched the weight visualization with feature visualization methods able to generate an image corresponding to the input node and the output node (see Figure 2.3). However, the feature visualization methods used to bring some context can also be difficult to interpret themselves, then it only moves the interpretability problem from weights to features.

2.2.2 Feature map visualization

Feature maps are the results of intermediate computations done from the input and resulting in the output value. Then, it seems natural to visualize them, or link them to concepts to understand how the input is successively transformed into the output.

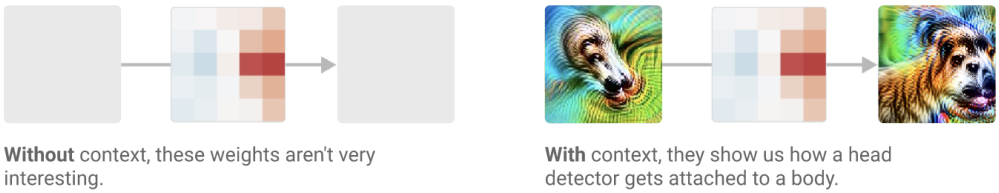


FIGURE 2.3: The weights of small kernels in hidden layers (here 5×5) can be really difficult to interpret alone. Here some context allow better understanding how it modulates the interaction between concepts conveyed by the input and the output.
Reproduced from (Voss et al., 2021).

Methods described in this section aim at highlighting which concepts a (part of a) feature map A conveys.

Direct interpretation

The output of a convolution has the same shape as its input: a 2D image processed by a convolution will become another 2D image (the size may vary). Then, it is possible to directly visualize these feature maps and compare them to the input to understand the operations performed by the network. However, the number of filters of convolutional layers (often a hundred) makes the interpretation difficult as a high number of images must be interpreted for a single input.

Instead of directly visualizing the feature map A, it is possible to study the latent space including all the values of the samples of a data set at the level of feature map A. Then it is possible to study the deformations of the input by drawing trajectories between samples in this latent space, or more simply to look at the distribution of some label in a manifold learned from the latent space. This way it is possible to better understand which patterns were detected, or at which layer in the network classes begin to be separated (in the classification case). These interpretations are often not conceptualized and may only appear in the context of an application (see Section 2.3.2 for examples).

Input optimization

Olah et al., 2017 proposed to compute an input that maximizes the value of a feature map A (see Figure 2.4). However this technique leads to unrealistic images that may be themselves difficult to interpret, particularly for neuroimaging data. To have a better insight of the behavior of layers or filters, another simple technique illustrated by the same authors consists in isolating the inputs that led to the highest activation of A. The combination of both methods, displayed in Figure 2.5, allows a better understanding of the concepts conveyed by the feature map A of a GoogleNet trained on natural images.

2.2.3 Back-propagation methods

The goal of these interpretability methods is to link the value of an output node of interest o_c to the image X_0 given as input to a network. The methods of this section bring an answer

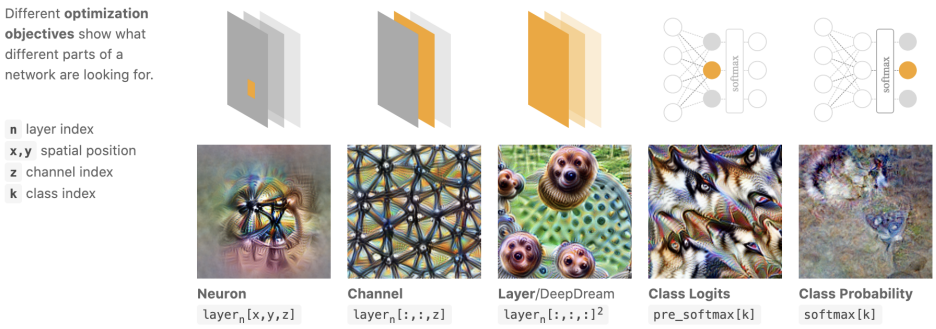


FIGURE 2.4: Optimization of the input for different levels of feature maps.
Reproduced from (Olah et al., 2017).

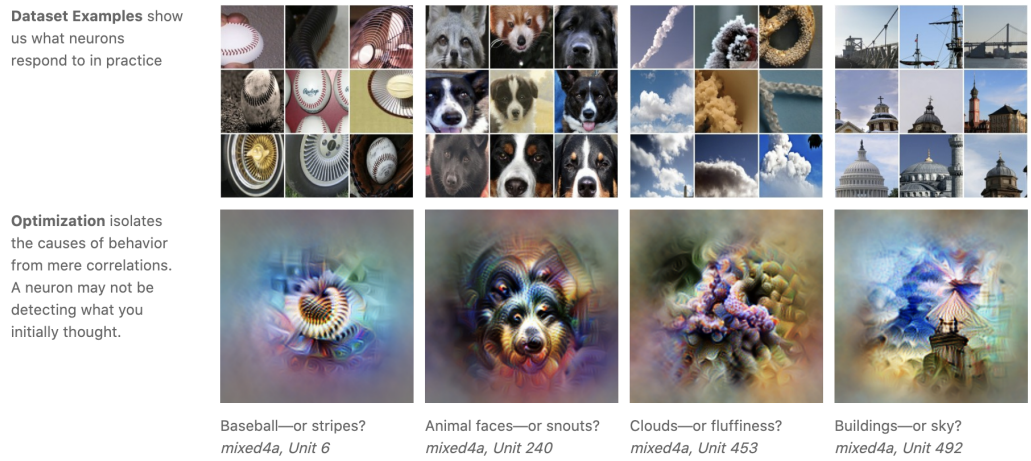


FIGURE 2.5: Interpretation of a neuron of a feature map by optimizing the input associated with a bunch of training examples maximizing this neuron.
Reproduced from (Olah et al., 2017).

by back-propagating a signal from o_c to X_0: this process (backward pass) can be seen as the opposite operation than the one done when computing the output value from the input (forward pass).

Any property can be back-propagated as soon as its value at the level of a feature map $l-1$ can be computed according to its value in the feature map l. In this section, the back-propagated properties are gradients or the relevance of a node o_c.

Gradient back-propagation

During network training, gradients corresponding to each layer are computed according to the loss to update the weights. Then, we can see these gradients as the difference needed at the layer level to improve the final result: by adding this difference to the weights, the probability of the true class y increases.

In the same way, the gradients can be computed at the image level to find how the input should vary to change the value of o_c. This gradient computation was proposed by Simonyan et al., 2014, in which the attribution map S_c corresponding to the input image

X_0 and the output node o_c is computed according to the following equation:

$$S_c = \frac{\partial o_c}{\partial X}\Big|_{X=X_0} \tag{2.1}$$

Due to its simplicity, this method is the most commonly used to interpret deep learning networks. Its attribution map is often called a "saliency map", however this term is also used in some articles to talk about any attribution map, and this is why we chose to avoid this term in this chapter.

This method was modified to derive many similar methods based on gradients computation described in the following paragraphs.

gradient⊙input This method is the point-wise product of the gradient map described at the beginning of the section and the input. Evaluated in Shrikumar et al., 2017, it was presented as an improvement of the gradients method, though the original paper does not give strong arguments on the nature of this improvement.

DeconvNet & guided back-propagation The key difference between this procedure and the standard back-propagation method is the way the gradients are back-propagated through the ReLU layer.

The ReLU layer is a commonly used activation function that sets to 0 the negative input values, and does not affect positive input values. The derivative of this function in layer l is the indicator function $\mathbb{1}_{A^{(l)}>0}$: it outputs 1 (resp. 0) where the feature maps computed during the forward pass were positive (resp. negative).

Springenberg et al., 2015 proposed to back propagate the signal differently. Instead of applying the indicator function of the feature map $A^{(l)}$ computed during the forward pass, they directly applied ReLU to the back-propagated values $R^{(l+1)} = \frac{\partial o_c}{\partial A^{(l+1)}}$, which corresponds to multiplying it by the indicator function $\mathbb{1}_{R^{(l+1)}>0}$. This "backward deconvnet" method allows back-propagating only the positive gradients, and, according to the authors, it results in a reconstructed image showing the part of the input image that is most strongly activating this neuron.

The guided back-propagation method (equation 2.4) combines the standard back-propagation (equation 2.2) with the backward deconvnet (equation 2.3): when back-propagating gradients through ReLU layers, a value is set to 0 if the corresponding top gradients or bottom data is negative. This adds an additional guidance to the standard back-propagation by preventing backward flow of negative gradients.

$$R^{(l)} = \mathbb{1}_{A^{(l)}>0} * R^{(l+1)} \tag{2.2}$$

$$R^{(l)} = \mathbb{1}_{R^{(l+1)}>0} * R^{(l+1)} \tag{2.3}$$

$$R^{(l)} = \mathbb{1}_{A^{(l)}>0} * \mathbb{1}_{R^{(l+1)}>0} * R^{(l+1)} \tag{2.4}$$

Any back-propagation procedure can be "guided", as it only concerns the way ReLU functions are managed during back-propagation (this is the case for example for guided Grad-CAM).

While it was initially adopted by the community, this method showed severe defects as discussed later in section 2.4.

CAM & Grad-CAM In this setting, attribution maps are computed at the level of a feature map produced by a convolutional layer, and then upsampled to be overlaid and compared with the input. The first method, class activation maps (CAM) was proposed by Zhou et al., 2016, and can be only applied to CNNs with the following specific architecture:

1. a series of convolutions associated with activation functions and possibly pooling layers. These convolutions output a feature map A with N channels,
2. a global average pooling that extracts the mean value of each channel of the feature map produced by the convolutions,
3. a single fully-connected layer.

The CAM corresponding to o_c will be the mean of the channels of the feature map produced by the convolutions, weighted by the weights w_{kc} learned in the fully-connected layer

$$S_c = \sum_{k=1}^{N} w_{kc} * A_k \quad . \tag{2.5}$$

This map has the same size as A_k, which might be smaller than the input if the convolutional part performs dimension reduction operations (which is very often the case). Then, the map is upsampled to the size of the input to be overlaid on the input.

Selvaraju et al., 2017 proposed an extension of CAM that can be applied to any architecture: Grad-CAM. Like CAM, the attribution map is a linear combination of the channels of a feature map computed by a convolutional layer. But, in this case, the weights of each channel are computed using gradient back-propagation

$$\alpha_{kc} = \frac{1}{|\mathscr{U}|} \sum_{u \in \mathscr{U}} \frac{\partial o_c}{\partial A_k(u)} \quad . \tag{2.6}$$

The final map is then the linear combination of the feature maps weighted by the coefficients. A ReLU activation is then applied to the result to only keep the features that have a positive influence on class c

$$S_c = ReLU(\sum_{k=1}^{N} \alpha_{kc} * A_k) \quad . \tag{2.7}$$

Similarly to CAM, this map is then upsampled to the input size.

Grad-CAM could be applied to any feature map produced by a convolution, but in practice the last convolutional layer is very often chosen. The authors argue that this layer

is "the best compromise between high-level semantics and detailed spatial information" (the latter is lost in fully-connected layers, as the feature maps are flattened).

Because of the upsampling step, CAM and Grad-CAM produce maps that are more human-friendly because they contain more connected zones, contrary to other attribution maps obtained with gradient back-propagation that can look very scattered. However, the smallest the feature maps A_k, the blurrier they are, leading to a possible loss of interpretability.

Relevance back-propagation

Instead of back-propagating gradients to the level of the input or of the last convolutional layer, Bach et al., 2015 proposed to back-propagate the score obtained by a class c, which is called the relevance. This score corresponds to o_c after some postprocessing (for example softmax), as its value must be positive if class c was identified in the input. At the end of the back-propagation process, the goal is to find the relevance R_u of each feature u of the input (for example, of each pixel of an image) such that $o_c = \sum_{u \in \mathscr{U}} R_u$ is true.

In their paper, Bach et al., 2015 takes the example of a fully-connected function defined by a matrix of weights w and a bias b at layer $l+1$. The value of a node v in feature map $A^{(l+1)}$ is computed during the forward pass by the given formula:

$$A^{(l+1)}(v) = b + \sum_{u \in \mathscr{U}} w_{uv} A^{(l)}(u) \tag{2.8}$$

During the back-propagation of the relevance, $R^{(l)}(u)$, the value of the relevance at the level of the layer $l+1$, is computed according to the values of the relevance $R^{(l+1)}(v)$ which are distributed according to the weights w learnt during the forward pass and the values of $A^{(l)}(v)$:

$$R^{(l)}(u) = \sum_{v \in \mathscr{V}} R^{(l+1)}(v) \frac{A^{(l)}(u) w_{uv}}{\sum_{u' \in \mathscr{U}} A^{(l)}(u') w_{u'v}} \ . \tag{2.9}$$

The main issue of the method comes from the fact that the denominator may become (close to) zero, leading to the explosion of the relevance back-propagated. Moreover, it was shown by Shrikumar et al., 2017 that when all activations are piece-wise linear (such as ReLU or leaky ReLU) the layer-wise relevance (LRP) method reproduces the output of gradient⊙input, questioning the usefulness of the method.

This is why Samek et al., 2017 proposed two variants of the standard-LRP method (Bach et al., 2015). Moreover they describe the behavior of the back-propagation in other layers than the linear ones (the convolutional one following the same formula as the linear). To simplify the equations in the following paragraphs, we now denote the weighted activations as $z_{uv} = A^{(l)}(u) w_{uv}$.

ϵ-rule The ϵ-rule integrates a parameter $\epsilon > 0$, used to avoid numerical instability. Though it avoids the case of a null denominator, this variant breaks the rule of relevance conservation across layers

$$R^{(l)}(u) = \sum_{v \in \mathcal{V}} R^{(l+1)}(v) \frac{z_{uv}}{\sum\limits_{u' \in \mathcal{U}} z_{u'v} + \epsilon \times sign\left(\sum\limits_{u' \in \mathcal{U}} z_{u'v}\right)} \quad . \tag{2.10}$$

β-rule The β-rule keeps the conservation of the relevance by treating separately the positive weighted activations z^+_{uv} from the negative ones z^-_{uv}

$$R^{(l)}(u) = \sum_{v \in \mathcal{V}} R^{(l+1)}(v) \left((1+\beta) \frac{z^+_{uv}}{\sum\limits_{u' \in \mathcal{U}} z^+_{u'v}} - \beta \frac{z^-_{uv}}{\sum\limits_{u' \in \mathcal{U}} z^-_{u'v}} \right) \quad . \tag{2.11}$$

Though these two LRP variants improve the numerical stability of the procedure, they imply to choose the values of parameters that may change the patterns in the obtained attribution map.

Deep Taylor decomposition Deep Taylor decomposition (Montavon et al., 2017) was proposed by the same team as the one which proposed the original LRP method and its variants. It is based on similar principles as LRP: the value of the score obtained by a class c is back-propagated, but the back-propagation rule is based on first-order Taylor expansions.

The back-propagation from node v in at the level of $R^{(l+1)}$ to u at the level of $R^{(l)}$ can be written

$$R^{(l)}(u) = \sum_{v \in \mathcal{V}} \frac{\partial R^{(l+1)}(v)}{\partial A^{(l)}(u)}\Big|_{\tilde{A}^{(l)}(u^{(v)})} \left(A^{(l)}(u) - \tilde{A}^{(l)}(u^{(v)}) \right) \quad . \tag{2.12}$$

This rule implies a root point $\tilde{A}^{(l)}(u^{(v)})$ which is close to $A^{(l)}(u)$ and meets a set of constraints depending on v.

2.2.4 Perturbation methods

Instead of relying on a backward pass (from the output to the input) as in the previous section, perturbation methods rely on the difference between the value of o_c computed with the original inputs and a locally perturbed input. This process is less abstract for humans than back-propagation methods as we can reproduce it ourselves: if the part of the image that is needed to find the good output is hidden, we are also not able to predict correctly, whereas we are able to correctly predict an image in which parts uncorrelated to the task are perturbed. Moreover, it is model-agnostic and can be applied to any algorithm or deep learning architecture.

The main drawback of these techniques is that the nature of the perturbation is crucial, leading to different attribution maps depending on the perturbation function used. Moreover, Montavon et al., 2018 suggest that the perturbation rule should keep the perturbed input in the training data distribution. Indeed, if it is not the case one cannot know if the network performance dropped because of the location or the nature of the perturbation. In the ideal case, one would like to only consider the location of the perturbation to draw conclusions on the features identified by the network in the input.

Standard perturbation

Zeiler and Fergus, 2014 proposed the most intuitive method relying on perturbations. This standard perturbation procedure consists in removing information locally in a specific zone of an input X_0 and evaluating if it modifies the output node o_c. The more the perturbation degrades the task performance, the more crucial this zone is for the network to correctly perform the task. To obtain the final attribution map, the input is perturbed according to all possible locations. Examples of attribution maps obtained with this method are displayed in Figure 2.6.

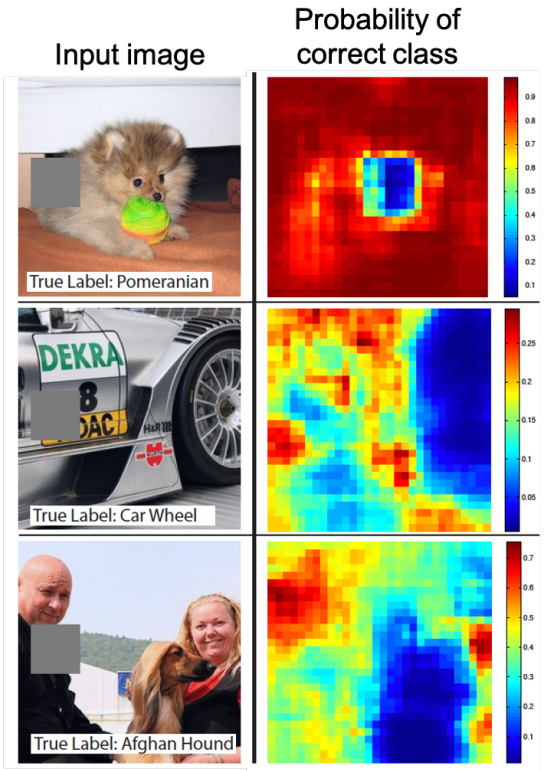


FIGURE 2.6: Attribution maps obtained with standard perturbation. Here the perturbation is a gray patch covering a specific zone of the input as shown in the left column. The attribution map s(right column) display the probability of the correct class: the lower the value, the most important it is for the network to correctly identify the label. This kind of perturbation takes the perturbed input out of the training distribution.
Reproduced and modified from (Zeiler and Fergus, 2014).

As evaluating the impact of the perturbation at each pixel location is computationally expensive, one can choose not to perturb the image at each pixel location, but to skip some of them (i.e. scan the image with a stride > 1). This will lead to a smaller attribution map, which needs to be upsampled to be compared to the original input (in the same way as CAM & Grad-CAM).

However, in addition to the problem of the nature of the perturbation already mentioned in the introduction of this section, this method presents two drawbacks:

- the attribution maps depend on the size of the perturbation: if the perturbation becomes too large, the perturbation is not local anymore, if it too small it is not

meaningful anymore (a pixel perturbation cannot cover a pattern),

- input features are considered independently from each other: if the result of a network relies on a combination of features that cannot all be covered at the same time by the perturbation, their influence may not be detected.

Optimized perturbation

To deal with these two issues, Fong and Vedaldi, 2017 proposed to optimize a perturbation mask covering the whole input. This perturbation mask m has the same size as the input X_0. Its application is associated with a perturbation function Φ and leads to the computation of the perturbed input X_0^m. Its value at a coordinate u reflects the quantity of information remaining in the perturbed image:

- if $m(u) = 1$, the pixel at location u is not perturbed and has the same value in the perturbed input as in the original input ($X_0^m(u) = X_0(u)$).
- if $m(u) = 0$ the pixel at location u is fully perturbed and the value in the perturbed image is the one given by the perturbation function only ($X_0^m(u) = \Phi(X_0)(u)$).

This principle can be extended to any value between 0 and 1 with the a linear interpolation

$$X_0^m(u) = m(u)X_0(u) + (1 - m(u))\Phi(X_0)(u) \ . \tag{2.13}$$

Then, the goal is to optimize this mask m according to three criteria:

1. the perturbed input X_0^m should lead to the lowest performance possible,
2. the mask m should perturb the minimum number of pixels possible, and
3. the mask m should produce connected zones (i.e. avoid the scattered aspect of gradient maps).

These three criteria are optimized using the following loss:

$$f(X_0^m) + \lambda_1 \|1 - m\|_{\beta_1}^{\beta_1} + \lambda_2 \|\nabla m\|_{\beta_2}^{\beta_2} \tag{2.14}$$

with f being a function that decreases as the performance of the network decreases.

Though theoretically it allows focusing on the minimum connected sets of pixels needed for the network to perform its task, the method also presents two drawbacks:

- The values of hyperparameters must be chosen (λ_1, λ_2, β_1, β_2) to find a balance between the three optimization criteria of the mask,
- The mask may not highlight the most important features of the input but instead create artifacts in the perturbed image to artificially degrade the performance of the network (see Figure 2.7).

FIGURE 2.7: In this example, the network learned to classify objects in natural images. Instead of masking the maypole at the center of the image, it creates artifacts in the sky to degrade the performance of the network.
Reproduced and modified from (Fong and Vedaldi, 2017).

2.2.5 Distillation

In this section, a transparent method is developed to reproduce the behavior of a black box one. Then it is possible to consider simple interpretability methods (such as weight visualization) on the transparent method instead of considering the black box (see the introduction for examples of interpretation of transparent methods).

LIME

Ribeiro et al., 2016 proposed Local Interpretable Model-agnostic Explanations (LIME). This approach is:

- **local**, as the explanation is valid in the vicinity of a specific input X_0,
- **interpretable**, as an interpretable model g (linear model, decision tree...) is computed to reproduce the behavior of f on X_0, and
- **model-agnostic**, as it does not depend on the algorithm trained.

This last property comes from the fact that the vicinity of X_0 is explored by sampling variations of X_0 that are perturbed versions of X_0. Then LIME shares the advantage (model agnostic) and drawback (perturbation function dependent) of perturbations methods presented in section 2.2.4. Moreover, the authors specify that, in the case of images, they group pixels of the input in d super-pixels (contiguous patches of similar pixels). The algorithm computing these super-pixels has to be chosen by the user and may also influence the results, in the same way as any standard feature selection procedure.

The loss to be minimized to find g specific to the input X_0 is the following:

$$\mathscr{L}(f, g, \pi_{X_0}) + \Omega(g) \ , \tag{2.15}$$

where π_{X_0} is a function that defines the locality of X_0 (i.e. $\pi_{X_0}(X)$ decreases as X becomes closer to X_0), $\mathscr{L}$ measures how unfaithful g is in approximating f according π_{X_0}, and Ω is a measure of the complexity of g.

Ribeiro et al., 2016 limited their search to sparse linear models (i.e. with a limited number K of non-null weights), however other assumptions could be made on g.

g is not applied to the input directly but to a binary mask $m \in \{0,1\}^d$ that transforms the input X in X^m and is applied according to a set of d super-pixels. For each super-pixel u:

1. if $m(u) = 1$ the super-pixel u is not perturbed,
2. if $m(u) = 0$ the super-pixel u is perturbed (i.e. it is grayed).

They used $\pi_{X_0}(X) = \exp\frac{(X-X_0)^2}{\sigma^2}$ and $\mathscr{L}(f,g,\pi_{X_0}) = \sum_m \pi_{X_0}(X_0^m) * (f(X_0^m) - g(m))^2$. Finally $\Omega(g)$ is the number of non-zero weights to g, and its value is limited to K. This way they select the K super-pixels in X_0 that best explain the algorithm result $f(X_0)$.

SHAP

Lundberg and Lee, 2017 proposed SHAP (SHapley Additive exPlanations), a theoretical framework that gathers several existing interpretability methods, including LIME. In this framework each of the N features (again, super-pixels for images) is associated with a coefficient ϕ that denotes its contribution to the result. The contribution of each feature is evaluated by perturbing the input X_0 with a binary mask m (see paragraph on LIME). Then the goal is to find an interpretable model g specific to X_0, such that

$$g(m) = \phi_0 + \sum_1^N \phi_i m_i \tag{2.16}$$

with ϕ_0 being the output when the input is fully perturbed.

The authors look for an expression of ϕ that respects three properties:

- **Local accuracy** g and f should match in the vincinity of X_0: $g(m) = f(X_0^m)$.
- **Missingness** Perturbed features should not contribute to the result: $m_i = 0 \rightarrow \phi_i = 0$.
- **Consistency** Let's denote as $m \setminus i$ the mask m in which $m_i = 0$. For any two models f^1 and f^2, if $f^1(X_0^m) - f^1(X_0^{m\setminus i}) \geq f^2(X_0^m) - f^2(X_0^{m\setminus i})$, then for all $m \in \{0,1\}^N$ $\phi_i^1 \geq \phi_i^2$ (ϕ^k are the coefficients associated with model f^k).

Lundberg and Lee, 2017 show that only one expression is possible for the coefficients ϕ, which can be approximated with different algorithms:

$$\phi_i = \sum_{m\in\{0,1\}^N} \frac{|m|!(N-|m|-1)!}{N!}\left[f(X_0^m) - f(X_0^{m\setminus i})\right] . \tag{2.17}$$

2.2.6 Intrinsic

Contrary to the previous sections in which interpretability methods could be applied to (almost) any network after the end of the training procedure, the following methods require to design the framework before the training phase, as the interpretability components and the network are trained simultaneously. In the papers presented in this section (Ba et al.,

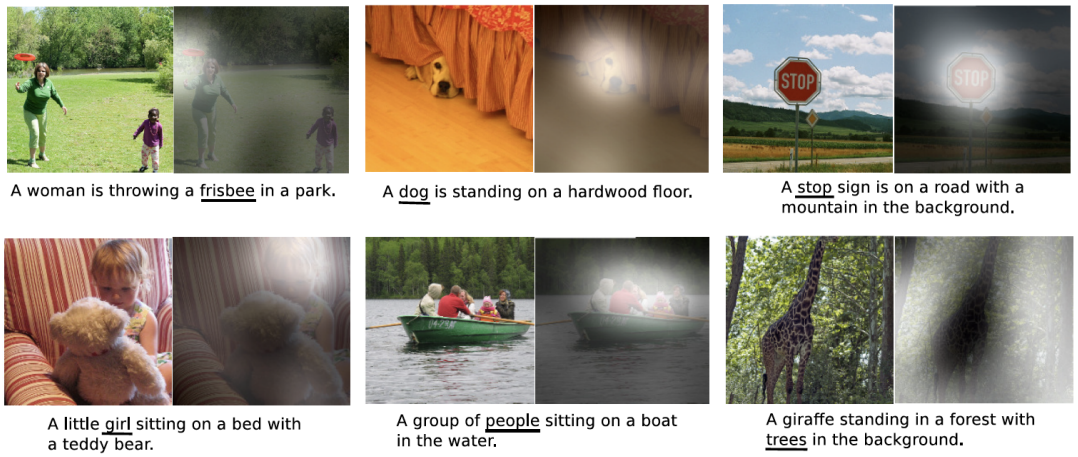


FIGURE 2.8: Examples of images correctly captioned by the network. The focus of the attribution map is highlighted in white and the associated word in the caption is underlined.
Reproduced from (Xu et al., 2015)

2015; Wang et al., 2017a; Xu et al., 2015), the advantages of these methods are dual: they improve both the interpretability and performance of the network. However, the drawback is that they have to be implemented before training the network, then they cannot be applied in all cases.

Attention modules

Attention is a concept in machine learning that consists in producing an attribution map from a feature map and using it to improve learning another task (such as classification, regression, reconstruction...) by making the algorithm focus on the part of the feature map highlighted by the attribution map.

In the deep learning domain, we take as reference the work of Xu et al., 2015, in which a network is trained to produce a descriptive caption of natural images. This network is composed of three parts:

1. a convolutional encoder that reduces the dimension of the input image to the size of the feature maps A,
2. an attention module that generates an attribution map S_t from A and the previous hidden state of the long short-term memory (LSTM) network,
3. an LSTM decoder that computes the caption from its previous hidden state, the previous word generated, A and S_t.

As S_t is of the same size as A (smaller than the input), the result is then upsampled to be overlaid on the input image. As one attribution map is generated per word generated by the LSTM, it is possible to know where the network focused when generating each word of the caption (see Figure 2.8). In this example, the attribution map is given to a LSTM, which uses it to generate a context vector z_t by applying a function ϕ to A and S_t.

More generally in CNNs, the point-wise product of the attribution map S and the feature map A is used to generate the refined feature map A' which is given to the next layers of the network. Adding an attention module implies to make new choices for the architecture of the model: its location (on lower or higher feature maps) may impact the performance of the network. Moreover, it is possible to stack several attention modules along the network, as it was done in (Wang et al., 2017a).

Modular Transparency

Contrary to the studies of the previous sections, the frameworks of these categories are composed of several networks (modules) that interact with each other. Each module is a black box, but the transparency of the function, or the nature of the interaction between them, allows understanding how the system works globally and extracting interpretability metrics from it.

A large variety of setups can be designed following this principle, and it is not possible to draw a more detailed general rule for this section. We will take the example described in (Ba et al., 2015), which was adapted to neuroimaging data (see Section 2.3.6), to illustrate this section, though it may not be representative of all the aspects of modular transparency.

Ba et al., 2015 proposed a framework (illustrated in Figure 2.9) to perform the analysis of an image in the same way as a human, by looking at successive relevant locations in the image. To perform this task, they assemble a set of networks that interact together:

- **Glimpse network** This network takes as input a patch of the input image and the location of its center to output a context vector that will be processed by the recurrent network. Then this vector conveys information on the main features in a patch and its location.
- **Recurrent network** This network takes as input the successive context vectors and update its hidden state that will be used to find the next location to look at and to perform the learned task at the global scale (in the original paper a classification of the whole input image).
- **Emission network** This network takes as input the current state of the recurrent network and outputs the next location to look at. This will allow computing the patch that will feed the glimpse network.
- **Context network** This network takes as input the whole input at the beginning of the task and outputs the first context vector to initialize the recurrent network.
- **Classification network** This network takes as input the current state of the recurrent network and outputs a prediction for the class label.

The global framework can be seen as interpretable as it is possible to review the successive processed locations.

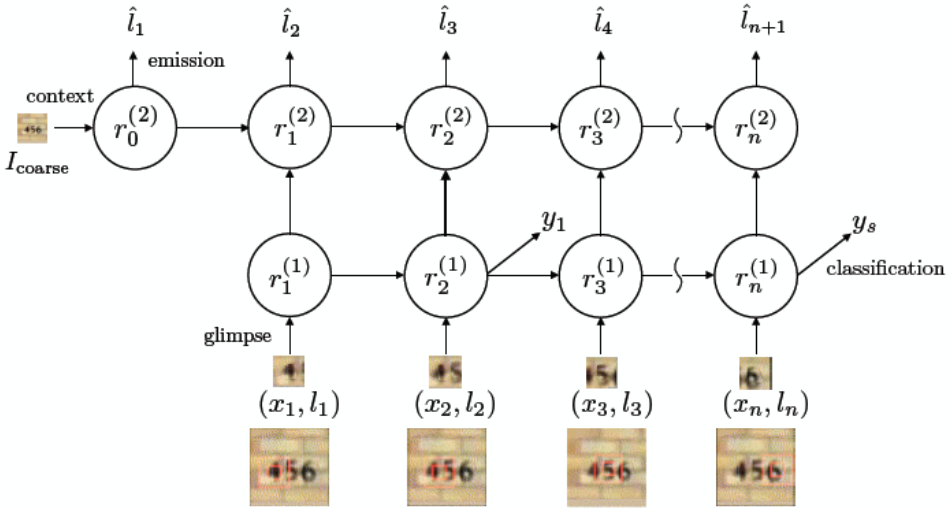


FIGURE 2.9: Framework with modular transparency browsing an image to compute the output at the global scale.
Reproduced from (Ba et al., 2015).

2.2.7 Interpretability metrics

To evaluate the reliability of the methods presented in the previous sections, one cannot only rely on qualitative evaluation. This is why interpretability metrics that evaluate attribution maps were proposed. These metrics may evaluate different properties of attribution maps.

- **Fidelity** evaluates if the zones highlighted by the map influence the decision of the network.
- **Sensitivity** evaluates how the attribution map changes according to small changes in the input X_0.
- **Continuity** evaluates if two close data points lead to similar attribution maps.

In the following, Γ is an interpretability method computing an attribution map S of the black-box network f and an input X_0.

(In)fidelity

Yeh et al., 2019 proposed a measure of infidelity of Γ based on perturbations applied according to a vector m of the same shape as the attribution map S. The explanation is infidel if perturbations applied in zones highlighted by S on X_0 leads to negligible changes in $f(X_0^m)$ or, on the contrary, if perturbations applied in zones not highlighted by S on X_0 lead to significant changes in $f(X_0^m)$. The associated formula is

$$\text{INFD}(\Gamma, f, X_0) = \mathbb{E}_m \left[\sum_i \sum_j m_{ij} \Gamma(f, X_0)_{ij} - (f(X_0) - f(X_0^m))^2 \right] . \tag{2.18}$$

Sensitivity

Yeh et al., 2019 also gave a measure of sensitivity. As suggested by the definition, it relies on the construction of attribution maps according to inputs similar to X_0: $\tilde{X_0}$. As changes are small, sensitivity depends on a scalar ϵ set by the user, which corresponds to the maximum difference allowed between X_0 and $\tilde{X_0}$. Then sensitivity corresponds to the following formula:

$$\text{SENS}_{\max}(\Gamma, f, X_0, \epsilon) = \max_{\|\tilde{X_0} - X_0\| \leq \epsilon} \|\Gamma(f, \tilde{X_0}) - \Gamma(f, X_0)\| \quad . \tag{2.19}$$

Continuity

Continuity is very similar to sensitivity, except that it compares different data points belonging to the input domain $\mathcal{X}$, whereas sensitivity may generate similar inputs with a perturbation method. This measure was introduced in Montavon et al., 2018 and can be computed using the following formula:

$$\text{CONT}(\Gamma, f, \mathcal{X}) = \max_{X_1, X_2 \in \mathcal{X} \ \& \ X_1 \neq X_2} \frac{\|\Gamma(f, X_1) - \Gamma(f, X_2)\|_1}{\|X_1 - X_2\|_2} \quad . \tag{2.20}$$

As these metrics rely on perturbation, they are also influenced by the nature of the perturbation and may lead to different results, which is a major issue (see Section 2.4). Other metrics were also proposed and depend on the task learned by the network: for example in the case of a classification, statistical tests can be conducted between attribution maps of different classes to assess whether they differ according to the class they explain.

2.3 Application of interpretability methods to neuroimaging data

In this section, we look at the applications of interpretability methods to neuroimaging data in the literature. In most cases, the focus of articles is the prediction method rather than the interpretability one, which is just seen as a tool to analyze the results. Thus, authors did not usually explain their choice of interpretability method. Another key consideration here is the spatial registration of brain images, which enables having brain regions roughly at the same position between samples. This technique is of paramount importance as attribution maps computed for registered images can then be averaged or used to automatically determine the most important brain areas, which would not be possible with unaligned images. Finally, in the following, standard interpretability methods are used if no precision is added. All the studies presented in this section are summarized in Table 2.1.

Study	Data set	Modality	Task	Interpretability method	Section
Abrol et al., 2020	ADNI	T1w	AD classification	FM visualization Perturbation	2.3.2, 2.3.4
Bae et al., 2019	ADNI	sMRI	AD classification	Perturbation	2.3.4
Ball et al., 2021	PING	T1w	Age prediction	Weight visualization SHAP	2.3.1, 2.3.5
Biffi et al., 2020	ADNI	T1w	AD classification	FM visualization	2.3.2
Böhle et al., 2019	ADNI	T1w	AD classification	LRP Guided back-propagation	2.3.3
Burduja et al., 2020	RSNA	CT scan	Intracranial hemorrhage detection	Grad-CAM	2.3.3
Cecotti and Gräser, 2011	in-house	EEG	P300 signals detection	Weight visualization	2.3.1
Dyrba et al., 2020	ADNI	T1w	AD classification	DeconvNet Deep Taylor decomposition Gradient ⊙ Input LRP Grad-CAM	2.3.3
Eitel and Ritter, 2019	ADNI	T1w	AD classification	Gradient ⊙ Input Guided back-propagation LRP Perturbation	2.3.3, 2.3.4
Eitel et al., 2019	ADNI, in-house	T1w	Multiple sclerosis detection	Gradient ⊙ Input LRP	2.3.3
Fu et al., 2021	CQ500, RSNA	CT scan	Detection of critical findings in head CT scan	Attention mechanism	2.3.6

Study	Data set	Modality	Task	Interpretability method	Section
Gutiérrez-Becker and Wachinger, 2018	ADNI	T1w	AD classification	Perturbation	2.3.4
Hu et al., 2021	ADNI, NIFD	T1w	AD/CN/FTD classification	Guided back-propagation	2.3.3
Jin et al., 2020	ADNI, in-house	T1w	AD classification	Attention mechanism	2.3.6
Lee et al., 2019a	ADNI	T1w	AD classification	Modular transparency	2.3.6
Leming et al., 2020	OpenFMRI, ADNI, ABIDE, ABIDE II, ABCD, NDAR ICBM, UK Biobank, 1000FC	fMRI	Autism classification Sex classification Task vs rest classification	FM visualization Grad-CAM	2.3.2, 2.3.3
Magesh et al., 2020	PPMI	SPECT	Parkinson's disease detection	LIME	2.3.5
Martinez-Murcia et al., 2020	ADNI	T1w	AD classification Prediction of neuropsychological tests & other clinical variables	FM visualization	2.3.2
Nigri et al., 2020	ADNI, AIBL	T1w	AD classification	Perturbation Swap test	2.3.4
Oh et al., 2019	ADNI	T1w	AD classification	FM visualization Standard back-propagation Perturbation	2.3.2, 2.3.3, 2.3.4
Qiu et al., 2020	ADNI, AIBL, FHS, NACC	T1w	AD classification	Modular transparency	2.3.6

Study	Data set	Modality	Task	Interpretability method	Section
Ravi et al., 2019	ADNI	T1w	CN/MCI/AD reconstruction	Modular transparency	2.3.6
Rieke et al., 2018	ADNI	T1w	AD classification	Standard back-propagation Guided back-propagation Perturbation Brain area occlusion	2.3.3, 2.3.4
Tang et al., 2019	UCD-ADC, Brain Bank	Histology	Detection of amyloid-β pathology	Guided back-propagation Perturbation	2.3.3, 2.3.4
Wood et al., 2019	ADNI	T1w	AD classification	Modular transparency	2.3.6

TABLE 2.1: Summary of the studies presented in Section 2.3.
Data sets: 1000FC, 1000 Functional Connectomes; ABCD, Adolescent Brain Cognitive Development; ABIDE, Autism Brain Imaging Data Exchange; ADNI, Alzheimer's Disease Neuroimaging Initiative; AIBL, Australian Imaging, Biomarkers and Lifestyle; FHS, Framingham Heart Study; ICBM, International Consortium for Brain Mapping; NACC, National Alzheimer's Coordinating Center; NDAR, National Database for Autism Research; NIFD, frontotemporal lobar degeneration neuroimaging initiative; PING, Pediatric Imaging, Neurocognition and Genetics; PPMI, Parkinson's Progression Markers Initiative; RSNA, Radiological Society of North America 2019 Brain CT Hemorrhage dataset; UCD-ADC Brain Bank, University of California Davis Alzheimer's Disease Center Brain Bank.
Modalities: CT, computed tomography; EEG, electroencephalography; fMRI, functional magnetic resonance imaging; sMRI, structural magnetic resonance imaging; SPECT, single-photon emission computed tomography; T1w, T1-weighted [magnetic resonance imaging].
Tasks: AD, Alzheimer's disease; CN, cognitively normal; FTD, fronto-temporal dementia; MCI, mild cognitive impairment.
Interpretability methods: FM, feature maps; Grad-CAM, gradient-weighted class activation mapping; LIME, local interpretable model-agnostic explanations; LRP, layer-wise relevance; SHAP, SHapley Additive exPlanations.

2.3.1 Weight visualization applied to neuroimaging

As the focus of this chapter is on non-transparent models, such as deep learning ones, weight visualization was only rarely found. However this was the method chosen by Cecotti and Gräser, 2011, who developed a CNN architecture adapted to weight visualization to detect P300 signals in electroencephalograms. The input of this network is a matrix with rows corresponding to the 64 electrodes and columns to 78 time points. The two first layers of the networks are convolutions with rectangular filters: the first filters (size 1×64) combines the electrodes, whereas the second ones (13×1) find time patterns. Then it is possible to retrieve a coefficient per electrode by summing the weights associated with this electrode across the different filters, and to visualize the results in electroencephalogram space as show in Figure 2.10.

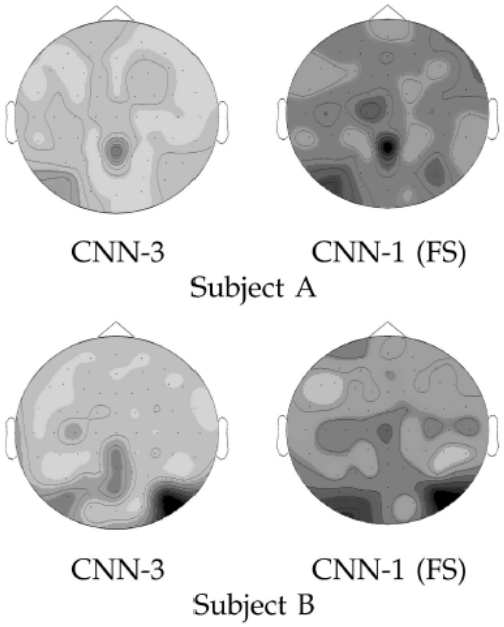


FIGURE 2.10: Relative importance of the electrodes for P300 signal detection using two different architectures (CNN-1 and CNN-3) and two subjects (A and B) using CNN weight visualization. Dark values correspond to weights with a high absolute value while white values correspond to weights close to 0.
Reproduced from (Cecotti and Gräser, 2011).

2.3.2 Feature map visualization applied to neuroimaging

Contrary to the limited application of weight visualization, there is an extensive literature about leveraging individual feature maps and latent spaces to better understand how models work. This goes from the visualization of these maps or their projections (Abrol et al., 2020; Biffi et al., 2020; Oh et al., 2019), to the analysis of neuron behavior (Leming et al., 2020; Martinez-Murcia et al., 2020), through the sampling in latent spaces (Biffi et al., 2020).

Oh et al., 2019 displayed the features maps associated with the convolutional layers of CNNs trained for various Alzheimer's disease status classification tasks (Figure 2.11). In

the first two layers, the extracted features were similar to white matter, cerebrospinal fluid and skull segmentations, while the last layer showcased sparse, global and nearly binary patterns that were assumed to be linked to important biomarkers by the authors.They used this example to emphasize the advantage of using CNNs to extract very abstract and complex features rather than using custom algorithms for features extraction, allowing the discovery of new biomarkers for neuroimaging analysis (Oh et al., 2019).

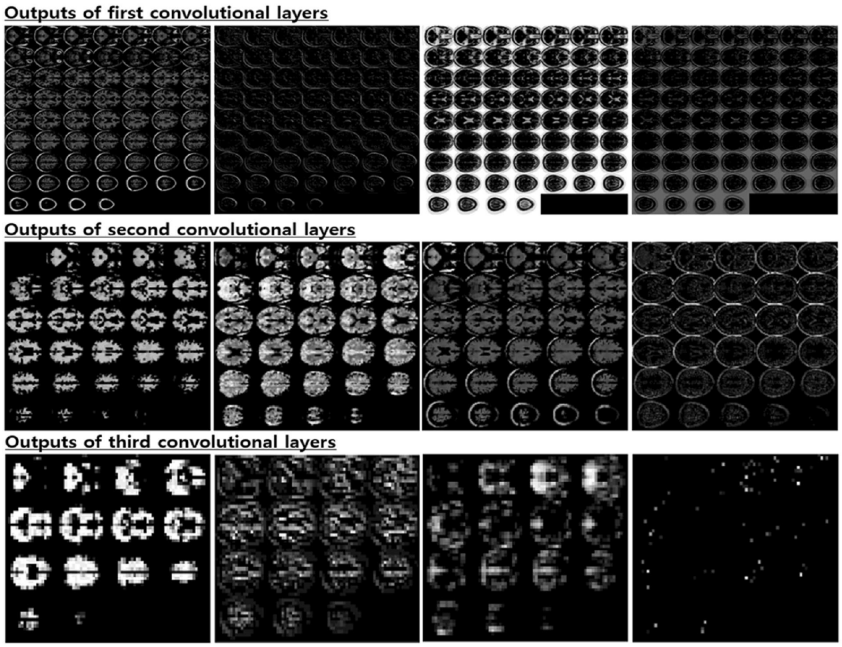


FIGURE 2.11: Representation of a selection of feature maps (outputs of 4 filters on 10 for each layer) obtained for a single individual.
Reproduced from (Oh et al., 2019).

Another way to visualize a feature map is to project it in a two or three-dimensional space to understand how it is positioned with respect to other feature maps. Abrol et al., 2020 projected the features obtained after the first dense layer of a ResNet architecture onto a two-dimensional space using the t-distributed stochastic neighbor embedding (t-SNE) dimensionality reduction technique. For the classification task of more than two Alzheimer's disease statuses, they observed that the projections were correctly ordered according to the disease severity, supporting the correctness of the model (Abrol et al., 2020). As shown in Figure 2.12, the authors also partitioned these projections into three groups: two homogeneous groups were at the extremities of the projection spectra (far-AD and far-CN respectively) and one heterogeneous group at the center of the projection spectra. Using a t-test, they were able to detect and highlight voxels presenting significant differences between groups. In particular, they showed that samples from homogeneous groups had stronger significant differences compared to samples from heterogeneous groups.

Biffi et al., 2020 not only used feature map visualization, but also sampled the feature space. Indeed, they trained a ladder variational autoencoder framework to learn hierarchical latent representations of 3D hippocampus segmentations of control subjects

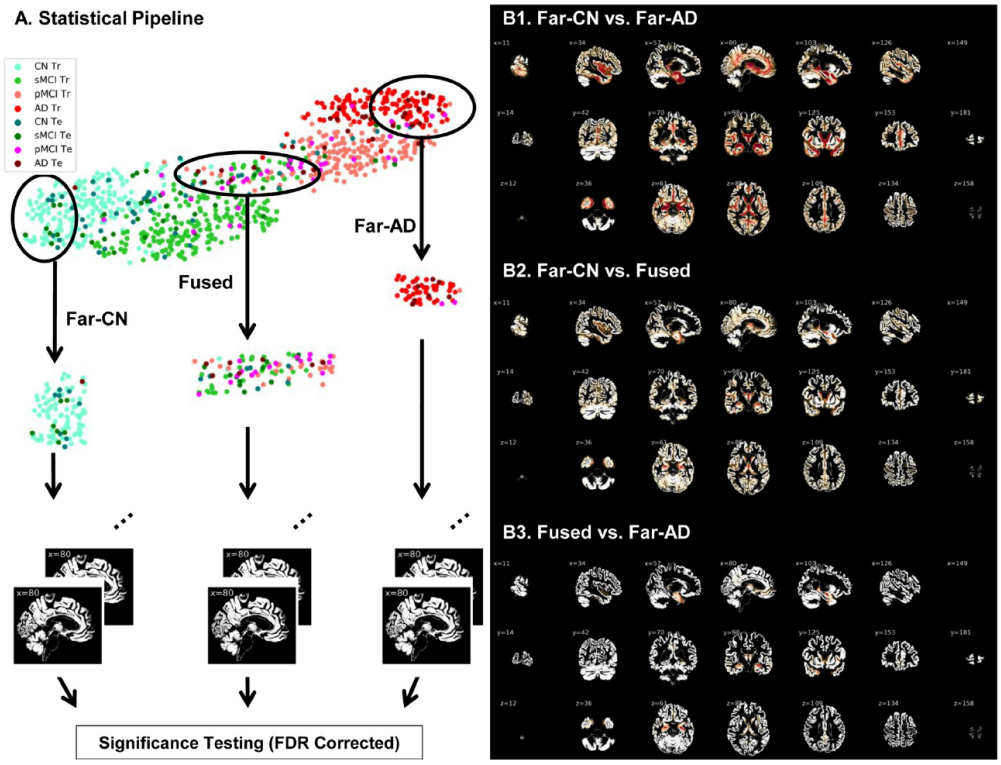


FIGURE 2.12: (A) t-distributed stochastic neighbor embedding (t-SNE) projections of the feature maps. Two homogenous groups (far-CN and far-AD) and a heterogeneous group (fused) were sampled and evaluated for significant differences in the input (preprocessed gray matter) space. Voxels showing significant differences post false discovery rate (FDR) correction ($p < 0.05$) are highlighted in panels B1, B2 and B3.
Reproduced from (Abrol et al., 2020).

and Alzheimer's disease patients. A multi-layer perceptron was jointly trained on top of the highest two-dimensional latent space to classify anatomical shapes. While lower spaces needed a dimensionality reduction technique (i.e. t-SNE), the highest latent space could directly be visualized, as well as the anatomical variability it captured in the initial input space, by leveraging the generative process of the model. This sampling enabled an easy visualization and quantification of the anatomical differences between each class.

Finally, it may be very informative to better understand the behavior of neurons and what they are encoding. After training deep convolutional autoencoders to reconstruct MR images, segmented gray matter maps and white matter maps, Martinez-Murcia et al., 2020 computed correlations between each individual hidden neuron value and clinical information (e.g. age, mini-mental state examination) which allowed them to determine to which extent this information was encoded in the latent space. This way they determined that the most strongly associated clinical data was the 11-question variant of the Alzheimer's Disease ASsessment Scale (ADAS-11) score. Using a collection of nine different MRI data sets, Leming et al., 2020 trained CNNs for various classification tasks (autism vs typical developing, male vs female and task vs rest). They computed a diversity coefficient for each filter of the second layer based on its output feature map. They counted how many different data sets maximally activated each value of this feature map: if they were mainly

activated by one source of data the coefficient would be close to 0, whereas if they were activated by all data sets it would be close to 1. This allows assessing the layer stratification, i.e. to understand if a given filter was mostly maximally activated by one phenotype or by a diverse population. They found out that a few filters were only maximally activated by images from a single MRI data set, and that the diversity coefficient was not normally distributed across filters, having generally two peaks at the beginning and at the end of the spectrum, respectively exhibiting the stratification and strongly diverse distribution of the filters.

2.3.3 Back-propagation methods applied to neuroimaging

Back-propagation methods are the most popular methods to interpret models, and a wide range of these algorithms have been used to study brain disorders: standard and guided back-propagation (Böhle et al., 2019; Eitel and Ritter, 2019; Hu et al., 2021; Oh et al., 2019; Rieke et al., 2018), gradient⊙input (Dyrba et al., 2020; Eitel and Ritter, 2019; Eitel et al., 2019), Grad-CAM (Burduja et al., 2020; Dyrba et al., 2020), guided Grad-CAM (Tang et al., 2019), LRP (Böhle et al., 2019; Dyrba et al., 2020; Eitel and Ritter, 2019; Eitel et al., 2019), DeconvNet (Dyrba et al., 2020) and deep Taylor Decomposition (Dyrba et al., 2020).

Single interpretation

Some studies implemented a single back-propagation method, and exploited it to find biomarkers (Hu et al., 2021; Leming et al., 2020; Oh et al., 2019), to show the usefulness of a method (Eitel et al., 2019) or to improve clinical guidance (Burduja et al., 2020).

Oh et al., 2019 used the standard back-propagation method to interpret CNNs trained for different binary classifications of Alzheimer's disease statuses (cognitively normal, stable mild cognitive impairment, progressive mild cognitive impairment and demented). They showed that the attribution maps associated with the prediction of the conversion of prodromal patients to dementia included more complex representations, less focused on the hippocampi, than the ones associated with the networks differentiating demented patients from cognitively normal participants (see Figure 2.13). They also exhibited that for a given classification task, models focused on the same biomarkers regardless of the target output node used to build the attribution map. In the context of autism, Leming et al., 2020 implemented the Grad-CAM algorithm to determine the most important brain connections from functional connectivity matrices used by CNNs detecting the phenotype. However, the authors pointed out that without further work, this visualization method did not allow understanding the underlying reason of the attribution of a given feature: for instance, one cannot know if a set of edges is important because it is under-connected or over-connected. Finally, Hu et al., 2021 used attribution maps produced by guided back-propagation to quantify the difference in the regions used by their network to characterize Alzheimer's disease or fronto-temporal dementia.

The goal of Eitel et al., 2019 was different. Instead of defining biomarkers of diseases, they exhibited with LRP that transfer learning between networks trained on different

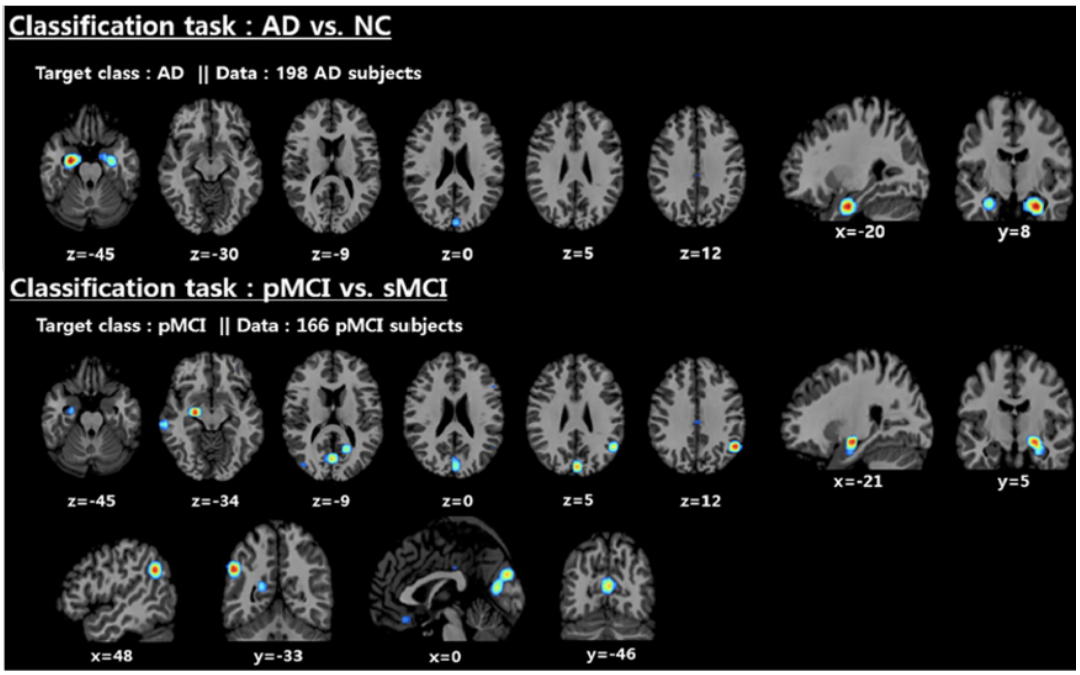


FIGURE 2.13: Distribution of discriminant regions obtained with gradient back-propagation in the classification of demented patients and cognitively normal participants (top part, AD vs CN) and the classification of stable and progressive mild cognitive impairment (bottom part, sMCI vs pMCI).
Reproduced and modified from (Oh et al., 2019).

diseases (Alzheimer's disease to multiple sclerosis) and different MRI sequences enabled obtaining attribution maps focused on a smaller number of lesion areas. However, due to the low number of samples they had, the authors pointed out that it would be necessary to check their results with a larger data set in future studies.

Finally, Burduja et al., 2020 trained a CNN-LSTM model to detect various hemorrhages from brain computed tomography (CT) scans. For each positive slice coming from controversial or difficult scans, they generated Grad-CAM based attribution maps and made team of radiologists analyze and classify them between correct, partially correct and incorrect. This classification allowed them to determine patterns for each class of maps, and better understand which characteristics radiologists expected from these maps to be considered as correct and thus useful in practice. In particular, radiologists described maps including any type of hemorrhage as incorrect as soon as some of the hemorrhages were not highlighted, while the model only needed to detect one hemorrhage to correctly classify the slice as pathological.

Comparison of several interpretability methods

Rather than employing back-propagation methods to extract medical knowledge from data sets, some papers have focused on the relative reliability of interpretability methods in their particular context. However, as the benchmark of interpretability methods is the focus of section 2.4.2, which also include other types of interpretability than back-propagation, we

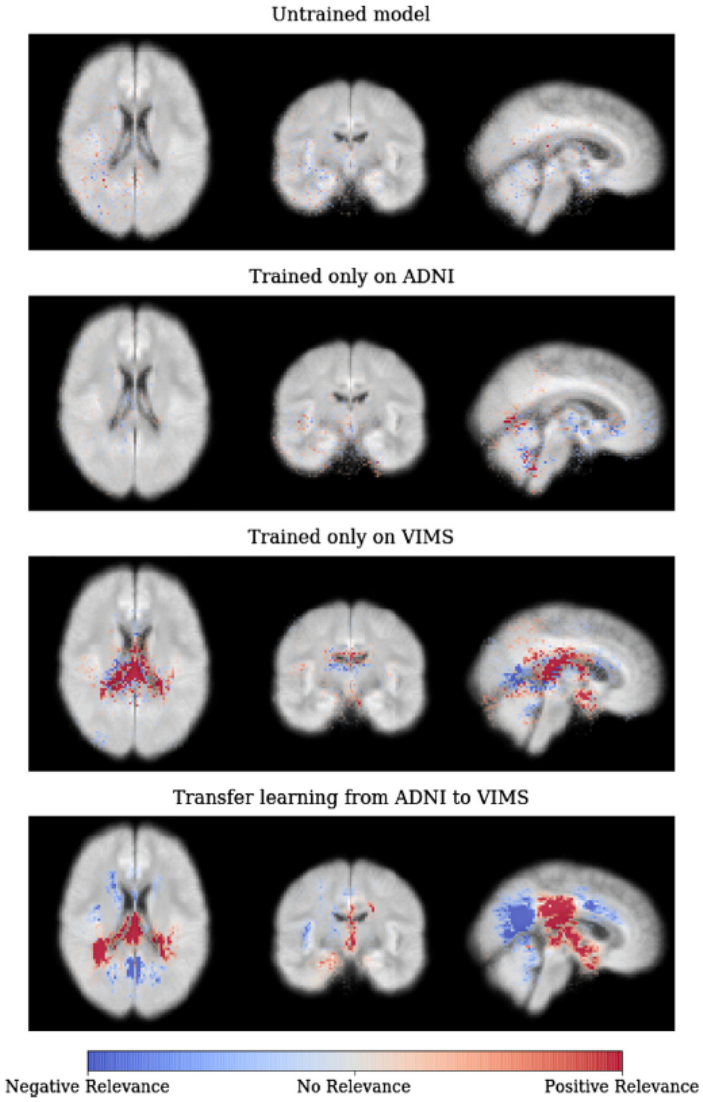


FIGURE 2.14: Average LRP attribution maps for different CNNs applied to multiple sclerosis. Different training strategies are compared, starting from an untrained CNN model with random parameters over a CNN trained only on either Alzheimer's disease or multiple sclerosis data to a CNN pre-trained on Alzheimer's disease and fine-tuned on multiple sclerosis.
Reproduced from (Eitel and Ritter, 2019).

will only focus here on the medical context and which conclusions were drawn from the attribution maps.

Dyrba et al., 2020 implemented and compared DeconvNet, guided back-propagation, deep Taylor decomposition, gradient⊙input, LRP (with various rules) and Grad-CAM methods on a CNN trained to classify MR images of Alzheimer's disease, mild cognitive impairment and normal cognition participants. In accordance with the literature, they obtained a highest attention given to the hippocampus for both prodromal and demented patients, but with different signs of contribution between methods.

Eitel and Ritter, 2019 tested the robustness of three different back-propagation based methods: gradient⊙input, guided back-propagation and LRP. They trained networks to differentiate cognitively normal participants from Alzheimer's disease patients, and

analyzed the results of their attribution maps thanks to a neuroanatomical atlas. Without normalization, only large regions (in the cerebellum) were highlighted. By normalizing the total intensity by the size of the regions, they found other regions such as the basal forebrain, the fourth ventricle, the hippocampus and the amygdala. They did not comment much these results as their focus was the comparison of the methods (see Section 2.4.2).

Böhle et al., 2019 compared two methods, LRP with β-rule and guided back-propagation, on a CNN trained for Alzheimer's disease status classification. They found that LRP attribution maps could highlight the individual differences between patients, and then that it could be used as a tool for clinical guidance.

2.3.4 Perturbation methods applied to neuroimaging

The standard perturbation method has been widely used in the study of Alzheimer's disease (Bae et al., 2019; Eitel and Ritter, 2019; Nigri et al., 2020; Rieke et al., 2018) and related symptoms (amyloid-β pathology) (Tang et al., 2019). However, most of the time, authors do not train their model with perturbed images. Hence, to generate explanation maps, the perturbation method uses images outside the distribution of the training set, which may call into question the relevance of the predictions and thus the reliability of attention maps.

Variants of the perturbation method tailored to neuroimaging

Several variations of the perturbation method have been developed to adapt to neuroimaging data. The most common variation in brain imaging is the brain area perturbation method, which consists in perturbing entire brain regions according to a given brain atlas, as done in (Abrol et al., 2020; Oh et al., 2019; Rieke et al., 2018). In their study of Alzheimer's disease, Abrol et al., 2020 obtained high values in their attribution maps for the usually discriminant brain regions, such as the hippocampus and amygdala subcortical regions, but also others that are more rarely exploited to analyze the conversion of mild cognitive impairment to Alzheimer's disease, such as the inferior and superior temporal gyruses and the fusiform gyrus. Thus, for a more complete characterization of the disease progression, the authors concluded on the necessity of more investigation to potentially improve the set of Alzheimer's disease biomarkers with these new regions. Rieke et al., 2018 also obtained results in accordance with the medical literature, and noted that the brain area perturbation method led to a less scattered attribution map than the standard method (Figure 2.15). Oh et al., 2019 used the method to compare the attribution maps of two different tasks: (1) demented patients vs cognitively normal participants and (2) stable vs progressive prodromal patients, and noted that the regions targeted for the first task were shared with the second one (medial temporal lobe), but that some regions were specific to the second task (parts of the parietal lobe).

Gutiérrez-Becker and Wachinger, 2018 adapted the standard perturbation method to a network that classified clouds of points extracted from neuroanatomical shapes of brain regions (e.g. left hippocampus) between different states of Alzheimer's disease. For the perturbation step, the authors set to 0 the coordinates of a given point x and the ones of its

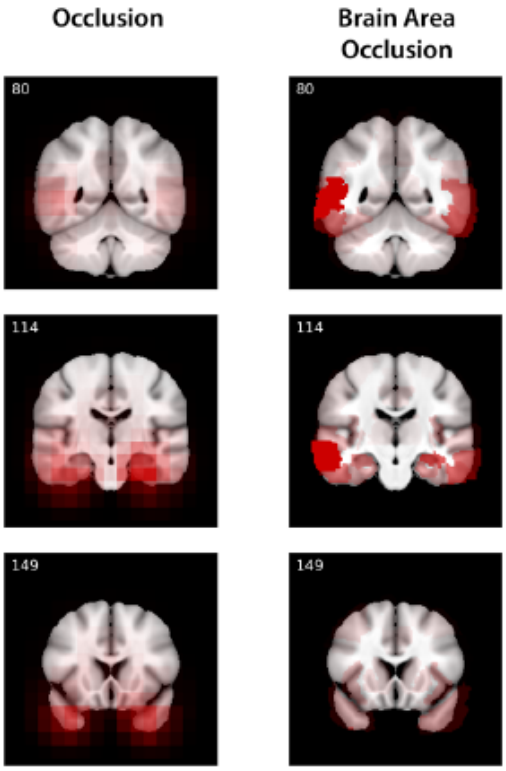


FIGURE 2.15: Mean attribution maps obtained on demented patients. The first column corresponds to the standard and the second one to the brain area perturbation method. Reproduced and modified from (Rieke et al., 2018)

neighbors to then assess the relevance of the point x. This method allows easily generating and visualizing a 3D attribution map of the shapes under study.

Advanced perturbation methods

More advanced perturbation based methods have also been used in the literature. Nigri et al., 2020 compared a classical perturbation method to a swap test. The swap test replaces the classical perturbation step by a swapping step where patches are exchanged between a brain image (to be explained) and a reference image chosen according to the model prediction. This exchange is possible as brain images were registered and thus brain regions are positioned in roughly the same location in each image.

Finally, Thibeau-Sutre et al., 2020 used the optimized version of the perturbation method to assess the robustness of CNNs in identifying regions of interest for Alzheimer's disease detection. They computed optimized perturbations on gray matter maps extracted from T1w MR images, and the perturbation method consisted in increasing the value of the voxels to transform patients into cognitively normal participants. This process aimed at simulating gray matter reconstruction to identify the most important regions that needed to be "de-atrophied" to be considered again as normal. After assessing the robustness of optimized perturbation, they found that region discovery by a CNN was not robust as it relied on different regions after retraining, as shown in Figure 2.16.

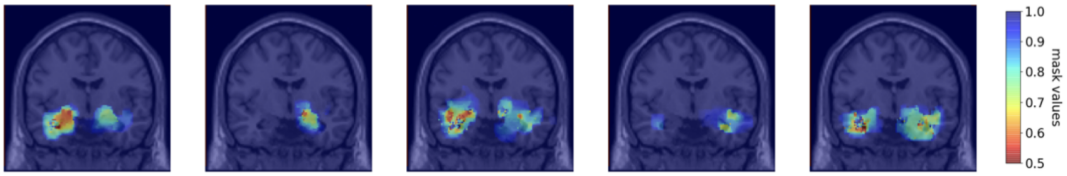


FIGURE 2.16: Coronal view of the mean attribution masks on demented patients obtained for five reruns of the same network with the optimized perturbation method.

2.3.5 Distillation methods applied to neuroimaging

Distillation methods are less commonly used, but some very interesting use cases can be found in the literature on brain disorders, with LIME (Magesh et al., 2020) and SHAP (Ball et al., 2021) methods.

Magesh et al., 2020 used LIME to interpret a CNN trained on Parkinson's disease detection from single-photon single-photon emission computed tomography (SPECT) scans. An illustration of these scans and the corresponding attribution maps for Parkinson's disease patients is available in Figure 2.17. Most of the time the most relevant regions are the putamen and the caudate (which is clinically relevant), and some patients (see 3rd and 4th columns in Figure 2.17) also showed an anomalous increase in dopamine activity in nearby areas, which is a characteristic feature of late-stage Parkinson's disease. The authors did not specify how they extracted the "super-pixels" necessary to the application of the method, though it could have been interesting to consider neuroanatomical regions instead of an automatic image slicing.

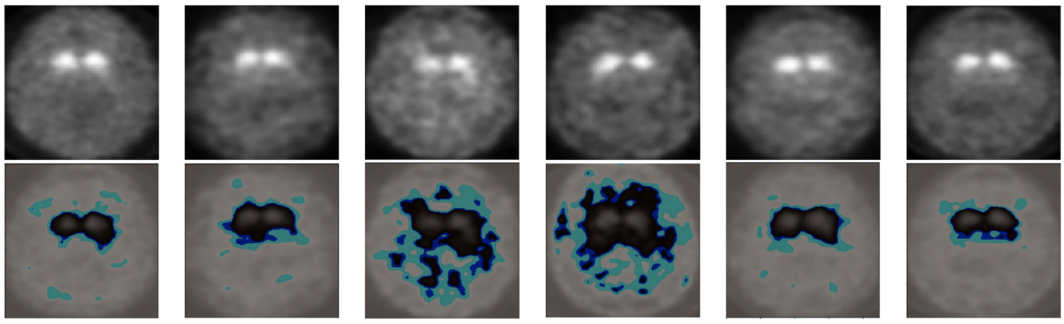


FIGURE 2.17: Attribution maps obtained with LIME applied to a network trained to detect Parkinson's disease. The first row corresponds to the SPECT scans of patients, and the second row to their corresponding attribution maps.
Reproduced and modified from (Magesh et al., 2020).

Ball et al., 2021 used SHAP to obtain explanations at the individual level from three different models, regularised linear model, Gaussian process regression and XGBoost, trained to predict participants' age from regional cortical thicknesses and areas (Figure 2.18). The authors exhibited a set of regions driving predictions for all models, and showed that regional attention was highly correlated on average with weights of the regularised linear model. However, they showed that while being consistent across models and training folds, explanations of SHAP at the individual level were generally not correlated with feature importance obtained from the weight analysis of the regularised linear model. The authors

also exemplified that the global contribution of a region to the final prediction error ("brain age delta"), even with a high SHAP value, was in general small, which indicated that this error was best explained by changes spread across several regions (Ball et al., 2021).

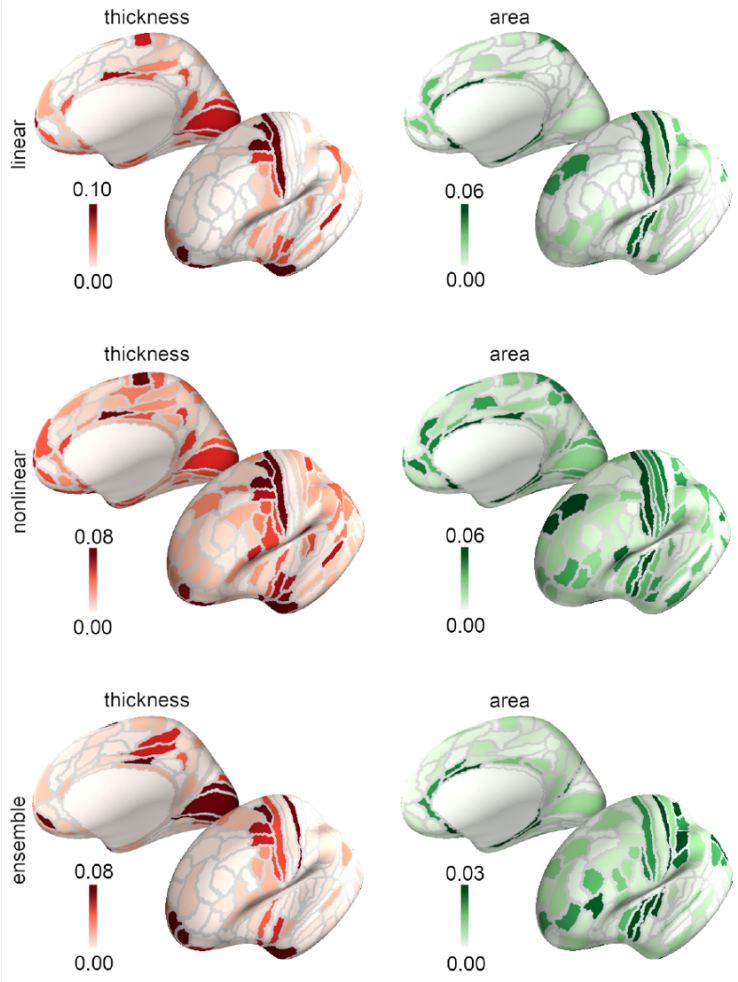


FIGURE 2.18: Mean absolute feature importance (SHAP values) averaged across all subjects for regional thickness (red) and area (green). Each row refer to a model: regularised linear model (linear), Gaussian process regression (nonlinear) and XGBoost (ensemble).
Reproduced and modified from (Ball et al., 2021).

2.3.6 Intrinsic methods applied to neuroimaging

Attention modules

Attention modules have been increasingly used in the past couple of years, as they often allow a boost in performance while being rather easy to implement and interpret. To diagnose various brain diseases from brain CT images, Fu et al., 2021 built a model integrating a "two step attention" mechanism that selects both the most important slices and the most important features in each slice. The authors then leveraged these attention modules to retrieve the five most suspicious slices and highlight the areas with the more significant attention, as shown in Figure 2.19 (Fu et al., 2021).

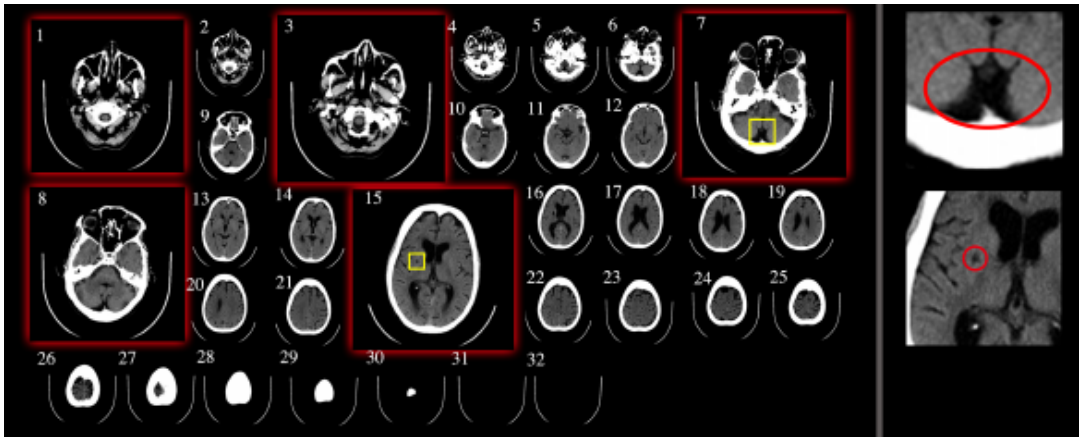


FIGURE 2.19: Display of the five most important slices (framed in red) found with attention mechanisms. These slices include indeed zones of interest (framed in yellow, zoom on the right).
Reproduced from (Fu et al., 2021).

In their study of Alzheimer's disease, Jin et al., 2020 used a 3D attention module to capture the most discriminant brain regions used for Alzheimer's disease diagnosis. As shown in Figure 2.20, they obtained significant correlations between attention patterns for two independent databases. They also obtained significant correlations between regional attention scores and classification accuracy or mini-mental state examination scores, which indicated a strong reproducibility of the results and the relevance of key discriminant regions as potential biomarkers for Alzheimer's disease diagnosis.

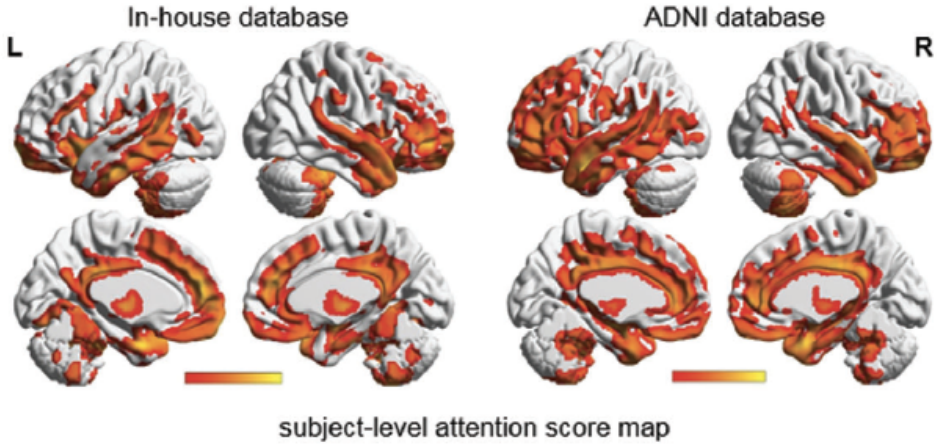


FIGURE 2.20: Attribution maps (left: in-house database, right: ADNI database) generated by an attention mechanism module, indicating the discriminant power of various brain regions for Alzheimer's disease diagnosis.
Reproduced and modified from (Jin et al., 2020).

Modular transparency

Modular transparency has often been used in brain imaging analysis. A possible practice consists in first generating a target probability map of a black-box model, before feeding

this map to a classifier to generate a final prediction, as done in (Lee et al., 2019a; Qiu et al., 2020).

Qiu et al., 2020 used a convolutional network to generate an attribution map from patches of the brain, highlighting brain regions associated with Alzheimer's disease diagnosis (see Figure 2.21). Lee et al., 2019a first parcellated gray matter density maps

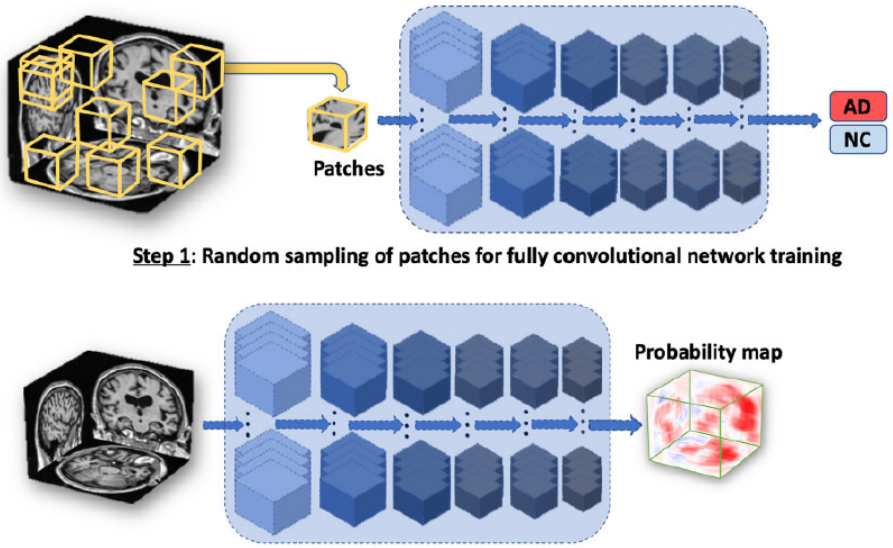


FIGURE 2.21: Randomly selected samples of T1-weighted full MRI volumes are used as input to learn the Alzheimer's disease status at the individual level (Step 1). The application of the model to whole images leads to the generation of participant-specific disease probability maps of the brain (Step 2).
Reproduced from (Qiu et al., 2020).

into 93 regions. For each of these regions, several deep neural networks were trained on randomly selected voxels and their outputs were averaged to obtain a mean regional disease probability. Then, by concatenating these regional probabilities, they generated a region-wise disease probability map of the brain, which was further used to perform Alzheimer's disease detection.

The model of Ba et al., 2015 was also translated to the Alzheimer's disease detection (Wood et al., 2019). Though that work is not published, the idea is interesting as it aims at reproducing the way a radiologist looks at an MR image. The main difference with (Ba et al., 2015) is the initialization, as the context network does not take as input the whole image but clinical data of the participant. Then the framework browses the image in the same way as the original one: a patch is processed by a recurrent neural network and from its internal state the glimpse network learns which patch should be looked at next. After a fixed number of iterations, the internal state of the recurrent neural network is processed by a classification network that gives the final outcome. The whole system is interpretable as the trajectory of the locations (illustrated in Figure 2.22) processed by the framework allows understanding which regions are more important for the diagnosis. We guess that this framework remains a preprint because of its high dependency to clinical data: as the initialization depends on scores used to diagnose Alzheimer's disease, the classification

network may learn to classify based on the initialization only and most of the trajectory may be negligible to assess the correct label.

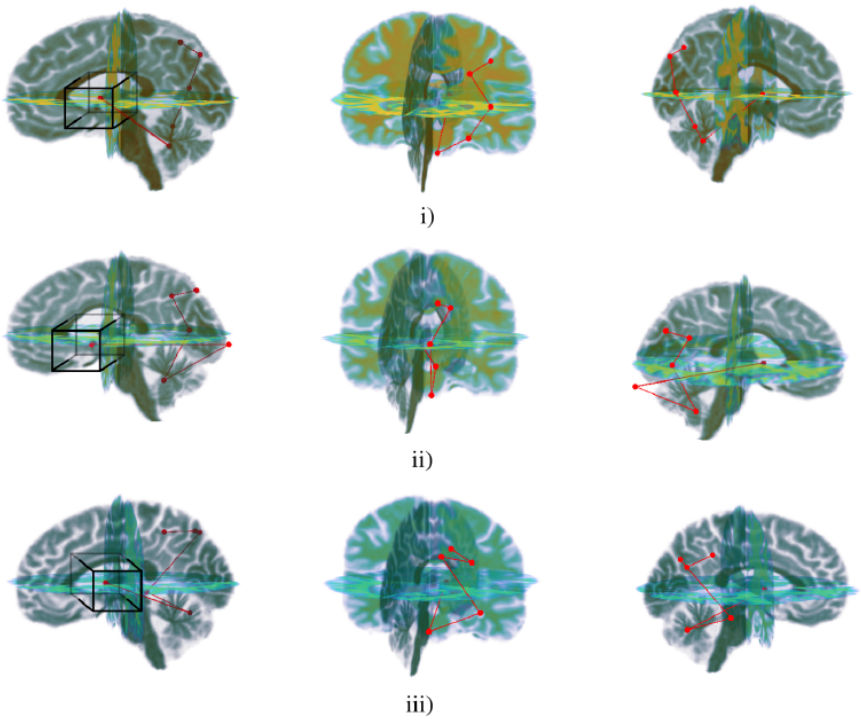


FIGURE 2.22: Trajectory taken by the framework for three representative participants from the ADNI test set (one per row). A bounding box around the first location attended to is included to indicate the approximate size of the glimpse that the recurrent neural network receives; this is the same for all subsequent locations.
Reproduced and modified from (Wood et al., 2019).

Another framework, the DaniNet, proposed by Ravi et al., 2019, is composed of multiple networks, each with a defined function, as illustrated in Figure 2.23.

- The preprocessing step (in blue) extracts a slice x from the whole MRI.
- The conditional deep autoencoder (in gray) learns to reduce the size of the slice x to a latent variable Z (encoder part), and then to reconstruct the original image based on Z and two additional variables: the diagnosis and age (generator part).
- Discriminator networks (in green) either force the encoder to take temporal progression into account (D_z) or try to determine if the output of the generator are real or generated images (D_b).
- Biological constraints (in orange) force the previous generated image of the same participant to be less atrophied than the next one (voxel-based progression loss) and learn to find the diagnosis thanks to regions of the generated images (region-based progression loss).

- The deformation loss (in yellow) aims at minimizing the difference between images belonging to the same participant.

The assembly of all these components allows learning a longitudinal model that characterizes the progression of the atrophy of each region of the brain. This atrophy evolution can then be visualized thanks to a neurodegeneration simulation generated by the trained model by sampling missing intermediate values (see Figure 2.24).

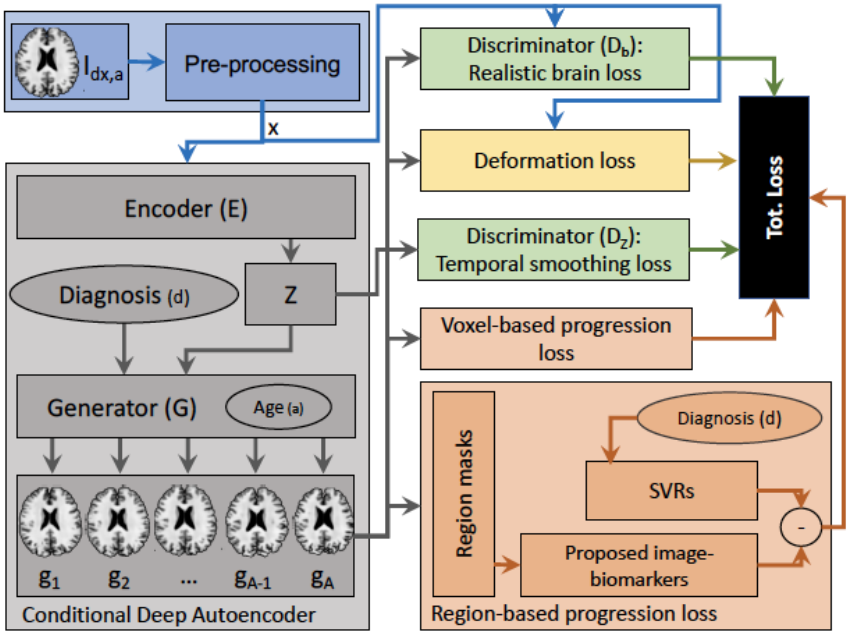


FIGURE 2.23: Pipeline used for training the proposed DaniNet framework that aims to learn a longitudinal model of the progression of Alzheimer's disease.
Reproduced from (Ravi et al., 2019).

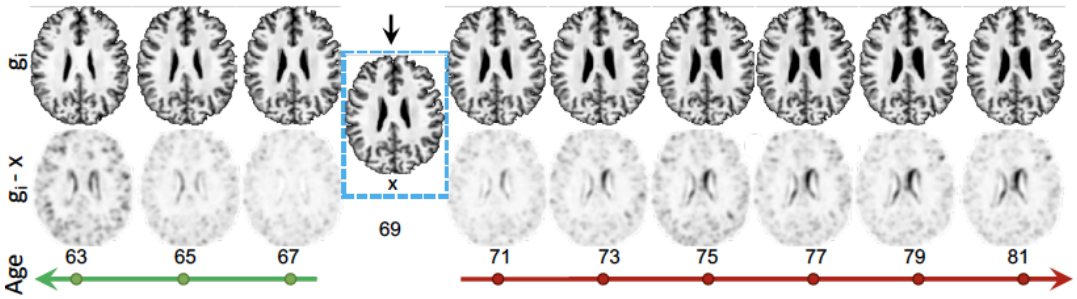


FIGURE 2.24: Neurodegeneration simulation of a 69-year old ADNI participant with a trained DaniNet.
Reproduced from (Ravi et al., 2019).

2.4 Limitations and comparison of methods

Though a large panel of studies were applied to neuroimaging data, the value of the results obtained from the interpretability methods is often still not clear. Moreover, many applications suffer from methodological issues, making their results (partly) irrelevant.

2.4.1 Theoretical limitations

It is not often clear whether the interpretability methods really highlight features relevant to the algorithm they interpret. This way, Adebayo et al., 2018 showed that the attribution maps produced by some interpretability methods (guided back-propagation and guided Grad-CAM) may not be correlated at all with the weights learnt by the network during its training procedure. They prove it with a simple test called "cascading randomization". In this test, the weights of a network trained on natural images are randomized layer per layer, until the network is fully randomized. At each step they produce an attribution map with a set of interpretability methods to compare it to the original ones (attribution maps produced without randomization). In the case of Guided Back-Propagation and Guided Grad-CAM, all attribution maps were identical, which means that the results of theses methods were independent from the training procedure.

Unfortunately, this type of failures does not only affect interpretability methods but also the metrics designed to evaluate their reliability, which makes the problem even more complex. Tomsett et al., 2020 investigated this issue by evaluating interpretability metrics with three properties:

- **inter-rater interpretability** assesses that a metric always rank different interpretability methods in the same way for different samples in the data set,
- **inter-method reliability** checks that the scores given by a metric on each interpretability method fluctuate in the same way between images,
- **internal consistency** observes if different metrics measuring the same property (for example fidelity) produce correlated scores on a set of attribution maps.

They concluded that the investigated metrics were not reliable, though it is difficult to know the origin of this unreliability due to the tight coupling of model, interpretability method and metric.

2.4.2 Benchmarks conducted in the literature

We separated in this section evaluations based on metrics from those purely qualitative. Indeed, even if the interpretability metrics are not mature yet, or that custom metrics may not be validated by the community, it is still better to try to measure quantitatively the difference between methods than only relying on human perception which may be biased.

Quantitative evaluations

Eitel and Ritter, 2019 tested the robustness of four methods: standard perturbation, gradient⊙input, guided back-propagation and LRP. To assess these methods, the authors trained 10 times the same model with random initialisations and generated attribution maps for each of the 10 runs. For each method, they exhibited significant differences between the averaged true positives/negatives attribution maps of the 10 runs. To quantify this variance, they computed the L2-norms between the attribution maps, and determined

for each model the brain regions with the highest attribution. They concluded that LRP and guided back-propagation were the most consistent methods, both in terms of distance between attribution maps and most relevant brain regions. However this study makes a strong assumption: to draw these conclusions, network retraining should be deterministic (or at least lead to very close results). Unfortunately, Thibeau-Sutre et al., 2020 showed that the study of the robustness of the interpretability method and of the network should be done separately, as their network retraining was not robust. Indeed, they first showed that the interpretability method they chose (optimized perturbation) was robust according to different criteria, then they observed that network retraining led to different attribution maps. Then, the robustness of an interpretability method cannot be assessed from the protocol of Eitel and Ritter, 2019. On the contrary, the fact that guided back-propagation is one of the most robust method meets the results of Adebayo et al., 2018, which observe that guided back-propagation always give the same result independently from the weights learnt by a network (see Section 2.4.1).

Böhle et al., 2019 measured the benefit of LRP with β-rule compared to guided back-propagation by comparing the intensities of the mean attribution map of demented patients and the one of cognitively normal controls. They concluded that LRP allowed a stronger distinction between these two classes than guided back-propagation, as there was a greater difference between the mean maps for LRP. Moreover, they found a stronger correlation between the intensities of the LRP attribution map in the hippocampus and the hippocampal volume than for guided back-propagation (see Figure 2.25). But as Adebayo et al., 2018 explained that guided back-propagation was not relevant at all, it does not allow to draw strong conclusions.

Nigri et al., 2020 compared the standard perturbation method to a swap test (see Section 2.3.4) using two properties: the continuity and the sensitivity. The continuity property is verified if two similar input images have similar explanations. The sensitivity property affirms that the most salient areas in an explanation map should have the greater impact in the prediction when removed. Nigri et al., 2020 carried out experiments with several types of models, and both properties were consistently verified for the swap test, while the standard perturbation method showed a significant absence of continuity and no conclusive fidelity values.

Finally Rieke et al., 2018 compared four visualization methods: standard back-propagation, guided back-propagation, standard perturbation and brain area perturbation. They computed the euclidean distance between the mean attribution maps of the same class for two different methods and observed that both gradient methods were close, whereas brain area perturbation was different from all others. They concluded that as interpretability methods lead to different attribution maps, one should compare the results of available methods and not trust only one attribution map.

Qualitative evaluation

Some methods also compared interpretability methods, but only qualitatively. First, Eitel et al., 2019 generated attribution maps using the LRP and gradient$\odot$input methods and

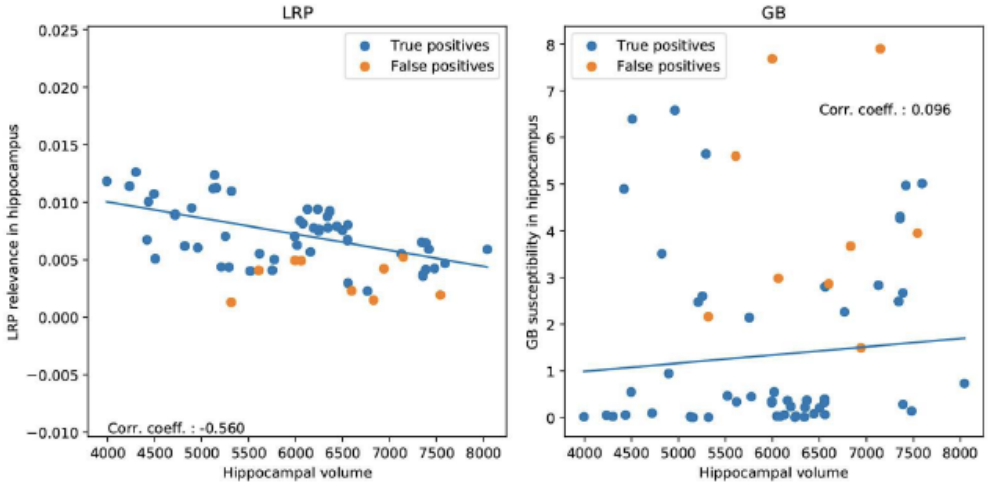


FIGURE 2.25: Correlation between hippocampal volume and attribution maps intensities in hippocampus for correctly classified AD patients (true positives). Left:LRP, Right: guided back-propagation (GB). For illustration, the false positive classifications were added but not taken into account for the regression computation.
Reproduced from (Böhle et al., 2019).

obtained very similar results. This could be expected as it was shown a strong link between LRP and gradient⊙input (see Section 2.2.3).

Dyrba et al., 2020 compared DeconvNet, guided back-propagation, Deep Taylor decomposition, gradient⊙input, LRP (with various rules) and Grad-CAM methods. The different methods roughly exhibited the same highlighted regions, but with a significant variability in focus, scatter and smoothness, especially for the Grad-CAM method. These conclusions were derived from a visual analysis. According to Dyrba et al., 2020, LRP and Deep Taylor Decomposition delivered the most promising results with a highest focus and less scatter.

Reaching the conclusion of Rieke et al., 2018, Tang et al., 2019 used two interpretability methods as they seemed to have different properties: guided Grad-CAM would provide a fine-grained view of feature salience, whereas standard perturbation highlights the interplay of features among classes.

Conclusion

The most extensively compared method is LRP, and it was shown each time that it was the best method compared to others. However, its equivalence with gradient⊙input for networks using ReLU activations still questions the usefulness of the method, as gradient⊙input is much easier to implement. Moreover, the studies reaching this conclusion are not very insightful: the study of Eitel and Ritter, 2019 may suffer from methodological biases, Böhle et al., 2019 compared LRP only to guided back-propagation, which was shown irrelevant (Adebayo et al., 2018), and Dyrba et al., 2020 only performed a qualitative assessment.

As proposed in conclusion by Rieke et al., 2018, a good way to assess the quality of interpretability methods could be to produce some form of ground truth for the attribution

maps, for example by implementing simulation models that control for the level of separability or location of differences.

Chapter 3

Convolutional neural networks for classification of Alzheimer's disease: A reproducible evaluation

---

This chapter is a part of an article published in *Medical Image Analysis*:

- **Title:** Convolutional Neural Networks for Classification of Alzheimer's Disease: Overview and Reproducible Evaluation
- **Authors:** Junhao Wen[†], Elina Thibeau-Sutre[†], Mauricio Diaz-Melo, Jorge Samper-González, Alexandre Routier, Simona Bottani, Didier Dormont, Stanley Durrleman, Ninon Burgos, Olivier Colliot, for the Alzheimer's Disease Neuroimaging Initiative and the Australian Imaging Biomarkers and Lifestyle flagship study of ageing.
- **DOI:** doi:10.1016/j.media.2020.101694
- **Contributions:** Literature review, 3D subject-level experiments, code programming and refactoring, manuscript redaction

[†] denotes shared first authorship

3.1 Introduction

Over the past decades, neuroimaging data have been increasingly used to characterize Alzheimer's disease (AD) by means of machine learning methods, offering promising tools for individualized diagnosis and prognosis (Falahati et al., 2014; Haller et al., 2011; Rathore et al., 2017). A large number of studies has proposed to use predefined features (including regional and voxel-based measurements) obtained from image preprocessing pipelines in combination with different types of classifiers, such as support vector machines (SVM) or random forests. Such approach is often referred to as conventional machine learning (LeCun et al., 2015). More recently, deep learning, as a newly emerging machine learning methodology, has made a big leap in the domain of medical imaging (Bernal et al., 2018; Liu

et al., 2018a; Lundervold and Lundervold, 2018; Razzak et al., 2018; Wen et al., 2018). As the most widely used architecture of deep learning , convolutional neural network (CNN) has attracted huge attention due to its great success in image classification (Krizhevsky et al., 2012). Contrary to conventional machine learning, deep learning allows the automatic abstraction of low-to-high level latent feature representations (e.g. lines, dots or edges for low level features, and objects or larger shapes for high level features). Thus, one can hypothesize that deep learning depends less on image preprocessing and requires less prior on other complex procedures, such as feature selection, resulting in a more objective and less bias-prone process (LeCun et al., 2015).

Numerous studies have proposed to assist diagnosis of AD by means of CNNs (Aderghal et al., 2018; Aderghal et al., 2017a,b; Bäckström et al., 2018; Basaia et al., 2019; Cheng and Liu, 2017; Cheng et al., 2017; Farooq et al., 2017; Gunawardena et al., 2017; Hon and Khan, 2017; Hosseini Asl et al., 2018; Islam and Zhang, 2017, 2018; Korolev et al., 2017; Li et al., 2017, 2018; Lian et al., 2018; Lin et al., 2018; Liu et al., 2018b; Liu et al., 2018c,e; Qiu et al., 2018; Senanayake et al., 2018; Shmulev et al., 2018; Taqi et al., 2018; Valliani and Soni, 2017; Vu et al., 2017, 2018; Wang et al., 2019; Wang et al., 2018a; Wang et al., 2017b; Wu et al., 2018). However, classification results among these studies are not directly comparable because they differ in terms of: i) sets of participants; ii) image preprocessing procedures, iii) cross-validation procedure and iv) reported evaluation metrics. It is thus impossible to determine which approach performs best. The generalization ability of these approaches also remains unclear. In deep learning, the use of fully independent test sets is even more critical than in conventional machine learning, because of the very high flexibility with numerous possible model architecture and training hyperparameter choices. Assessing generalization to other studies is also critical to ensure that the characteristics of the considered study have not been overfitted. In previous works, the generalization may be questionable due to inadequate validation procedures, the absence of an independent test set, or a test set chosen from the same study as the training and validation sets.

In previous studies led by our team, (Samper-González et al., 2018; Wen et al., 2021), we have proposed an open source framework for reproducible evaluation of AD classification using conventional machine learning methods. The framework comprises: i) tools to automatically convert three publicly available data sets into the Brain Imaging Data Structure (BIDS) format (Gorgolewski et al., 2016) and ii) a modular set of preprocessing pipelines, feature extraction and classification methods, together with an evaluation framework, that provide a baseline for benchmarking the different components. We demonstrated the use of this framework on positron emission tomography (PET), T1-weighted (T1w) MRI (Samper-González et al., 2018) and diffusion MRI data (Wen et al., 2021).

In this chapter, we rigorously assess the performance of different CNN architectures, representative of the literature. We studied the influence of key components on classification accuracy, we compared the proposed CNNs to a conventional machine learning approach based on a linear SVM, and we assessed the generalization ability of the CNN models within (training and testing on ADNI) and across data sets (training on ADNI

and testing on AIBL or OASIS).

All the code of the framework and the experiments is publicly available: general-purpose tools have been integrated into Clinica[1] (Routier et al., 2021), an open-source software platform that we developed to process data from neuroimaging studies, and the deep learning experiments were run with AD-DL `https://github.com/aramis-lab/AD-DL`. This software is the forerunner of ClinicaDL, a framework described in chapter 6. The tagged version v.0.0.1 corresponds to the version of the code used to obtain the results of the paper. The trained models are available on Zenodo and their associated DOI is 10.5281/zenodo.3491003.

3.2 Materials

Three publicly available data sets have been mainly used for the study of AD: the Alzheimer's Disease Neuroimaging Initiative (ADNI), the Australian Imaging, Biomarkers and Lifestyle (AIBL) and the Open Access Series of Imaging Studies (OASIS). In the following, we describe the diagnosis labels extracted from each data set.

We used the T1w MRI available in each of these studies. For the detailed MRI protocols, one can see (Samper-González et al., 2018).

The ADNI data set used in our experiments comprises 1455 participants for whom a T1w image was available at at least one visit. Five diagnosis groups were considered:

- CN: sessions of subjects who were diagnosed as CN at baseline and stayed stable during the follow-up;
- AD: sessions of subjects who were diagnosed as AD at baseline and stayed stable during the follow-up;
- MCI: sessions of subjects who were diagnosed as MCI, early MCI or late MCI at baseline, who did not encounter multiple reversions and conversions and who did not convert back to CN;
- pMCI: sessions of subjects who were diagnosed as MCI, early MCI or late MCI at baseline, and progressed to AD during the 36 months following the current visit;
- sMCI: sessions of subjects who were diagnosed as MCI, early MCI or late MCI at baseline, and neither progress nor regress to AD during the 36 months following the current visit.

AD and CN participants whose label changed over time were excluded. This was also the case for MCI patients with two or more label changes (for instance progressing to AD and then reverting back to MCI). We made this choice because one can assume that the diagnosis of these subjects is less reliable. Naturally, all the sessions of the pMCI and sMCI groups are included in the MCI group. Note that the reverse is false, as some MCI subjects did not convert to AD but were not followed long enough to state whether they were

[1] http://www.clinica.run/

	Subjects	Sessions	Age	Gender	MMSE	CDR
CN	330	1 830	74.4 (5.8) [59.8, 89.6]	160 M / 170 F	29.1 (1.1) [24, 30]	0: 330
MCI	787	3 458	73.3 (7.5) [54.4, 91.4]	464 M / 323 F	27.5 (1.8) [23, 30]	0: 2; 0.5: 785
sMCI	298	1 046	72.3 (7.4) [55.0, 88.4]	175 M / 123 F	28.0 (1.7) [23, 30]	0.5: 298
pMCI	295	865	73.8 (6.9) [55.1, 88.3]	176 M / 119 F	26.9 (1.7) [23, 30]	0.5: 293; 1: 2
AD	336	1 106	75.0 (7.8) [55.1, 90.9]	185 M / 151 F	23.2 (2.1) [18, 27]	0.5: 160; 1: 175; 2: 1

TABLE 3.1: Summary of participant demographics, mini-mental state examination (MMSE) and global clinical dementia rating (CDR) scores at baseline for ADNI. Values are presented as mean (standard deviation) [range]. M: male, F: female

	Subjects	Age	Gender	MMSE	CDR
CN	429	72.5 (6.2) [60, 92]	183 M / 246 F	28.8 (1.2) [25, 30]	0: 406; 0.5: 22; 1: 1
MCI	93	75.4 (6.9) [60, 96]	50 M / 43 F	27.0 (2.1) [20, 30]	0: 6; 0.5: 86; 1: 1
sMCI	13	76.7 (6.5) [64, 87]	8 M / 5 F	28.2 (1.5) [26, 30]	0.5: 13
pMCI	20	78.1 (6.6) [63, 91]	10 M / 10 F	26.7 (2.1) [22, 30]	0.5: 20
AD	76	73.9 (8.0) [55, 93]	33 M / 43 F	20.6 (5.5) [6, 29]	0.5: 31; 1: 36; 2: 7; 3: 2

TABLE 3.2: Summary of participant demographics, mini-mental state examination (MMSE) and global clinical dementia rating (CDR) scores at baseline for AIBL. Values are presented as mean (standard deviation) [range]. M: male, F: female

sMCI. For 30 sessions, the preprocessing did not pass the quality check (see Section 3.3.2) and these images were removed from our data set. Two pMCI subjects were entirely removed because the preprocessing failed for all their sessions. Table 3.1 summarizes the demographics, and the MMSE and global CDR scores of the ADNI participants.

The AIBL data set considered in this work is composed of 598 participants for whom a T1w MR image and an age value was available at at least one visit. The criteria used to create the diagnosis groups are identical to the ones used for ADNI. Table 3.2 summarizes the demographics, and the MMSE and global CDR scores of the AIBL participants. After the preprocessing pipeline, seven sessions were removed without changing the number of subjects.

The OASIS data set considered in this work is composed of 193 participants aged 62 years or more (minimum age of the participants diagnosed with AD). As this data set is not longitudinal, we consider as AD (resp. CN) participants who were diagnosed as AD (resp. CN) at baseline. Table 3.3 summarizes the demographics, and the MMSE and global CDR scores of the OASIS participants. After the preprocessing pipeline, 22 AD and 17 CN subjects were excluded.

Note that for the ADNI and AIBL data sets, three diagnosis labels (CN, MCI and AD) exist and are assigned by a physician after a series of clinical tests (Ellis et al., 2009, 2010; Petersen et al., 2010) while for OASIS only two diagnosis labels exist, CN and AD (the MCI

	Subjects	Age	Gender	MMSE	CDR
CN	76	76.5 (8.4) [62, 94]	14 M / 62 F	29.0 (1.2) [25, 30]	0: 76
AD	78	75.6 (7.0) [62, 96]	35 M / 43 F	24.4 (4.3) [14, 30]	0.5: 56; 1: 20; 2: 2

TABLE 3.3: Summary of participant demographics, mini-mental state examination (MMSE) and global clinical dementia rating (CDR) scores at baseline for OASIS. Values are presented as mean (standard deviation) [range]. M: male, F: female

subjects are labelled as AD), and it is assigned based on the CDR only (Marcus et al., 2007). As the diagnostic criteria of these studies differ, there is no strict equivalence between the labels of ADNI and AIBL, and those of OASIS.

3.3 Methods

In this section, we present the main components of our framework: automatic converters of public data sets for reproducible data management (Section 3.3.1), preprocessing of MRI data (Section 3.3.2), classification models (Section 3.3.3), transfer learning approaches (Section 3.3.4), classification tasks (Section 3.3.5), evaluation strategy (Section 3.3.6) and framework implementation details (Section 3.3.7).

3.3.1 Converting data sets to a standardized data structure

ADNI, AIBL and OASIS, as public data sets, are extremely useful to the research community. However, they may be difficult to use because the downloaded raw data do not possess a clear and uniform organization. We thus used our Clinica converters (Routier et al., 2021) to convert the raw data into the BIDS format (Gorgolewski et al., 2016). Finally, we organized all the outputs of the experiments into a standardized structure, inspired from BIDS.

3.3.2 Preprocessing of T1w MRI

In principle, CNNs require only minimal preprocessing because of their ability to automatically extract low-to-high level features. However, in AD classification where data sets are relatively small and thus deep networks may be difficult to train, it remains unclear whether they can benefit from more extensive preprocessing. Moreover, previous studies have used varied preprocessing procedures but without systematically assessing their impact. Thus, in the current study, we compared two different procedures: “Minimal” and “Extensive”. Both procedures included bias field correction, and (optional) intensity rescaling. In addition, the “Minimal” processing included a linear registration while the “Extensive” included non-linear registration and skull-stripping.

In brief, the “Minimal” preprocessing procedure performs the following operations. The N4ITK method (Tustison et al., 2010) was used for bias field correction. Next, a linear (affine) registration was performed using the SyN algorithm from ANTs (Avants et al., 2008) to register each image to the MNI space (ICBM 2009c nonlinear symmetric template) (Fonov et al., 2009; Fonov et al., 2011). To improve the computational efficiency, the registered images were further cropped to remove the background. The final image size is 169×208×179 with 1 mm3 isotropic voxels. Intensity rescaling, which was performed based on the min and max values, denoted as MinMax, was set to be optional to study its influence on the classification results.

In the “Extensive” preprocessing procedure, bias field correction and non-linear registration were performed using the Unified Segmentation approach (Ashburner and

Friston, 2005) available in SPM12[2]. Note that we do not use the tissue probability maps but only the nonlinearly registered, bias corrected, MR images. Subsequently, we perform skull-stripping based on a brain mask drawn in MNI space. We chose this mask-based approach over direct image-based skull-stripping procedures because the later did not prove robust on our data. This mask-based approach is less accurate but more robust. In addition, we performed intensity rescaling as in the "Minimal" pipeline.

We performed quality check on the outputs of the "Minimal" preprocessing procedure. We used a quality check framework based on a deep learning network[3] (Fonov et al., 2018) to automatically check the quality of the linearly registered data. This software outputs a probability indicating how accurate the registration is. We excluded the scans with a probability lower than 0.5 and visually checked the remaining scans whose probability were lower than 0.70. As a result, 30 ADNI scans, 7 AIBL scans, and 39 OASIS scans were excluded. The quality check procedure excluded more OASIS scans than in ADNI and AIBL in proportion to the whole data set. We guess that this difference may come from the anonymization of OASIS data: as this data set is public, the face and ears of patients have been removed. These images may be more difficult to register, but also to evaluate during the quality check procedure.

3.3.3 Classification models

We considered the same four classification approaches as in chapter 1: i) 3D subject-level CNN, ii) 3D ROI-based CNN, iii) 3D patch-level CNN and iv) 2D slice-level CNN.

In the case of deep learning, one challenge is to find the "optimal" model (i.e. global minimum), including the architecture hyperparameters (e.g. number of layers, dropout, batch normalization) and the training hyperparameters (e.g. learning rate, weight decay). We first reviewed the architectures used in the literature among the studies in which no data leakage problem was found (Table 1.1). As there was no consensus, we used the following heuristic strategy for each of the four approaches. For the 3D subject-level approach, we began with an overfitting model that was very heavy because of the high number of FC layers (4 convolutional blocks + 5 FC layers). Then, we iteratively repeated the following operations:

- the number of FC layers was decreased until accuracy on the validation set decreased substantially;
- we added one more convolutional block.

In this way, we explored the architecture space from 4 convolutional blocks + 5 FC layers to 7 convolutional blocks + 2 FC layers. Among the best performing architectures, we chose the shallowest one: 5 convolutional blocks + 3 FC layers.

As the performance was very similar for the different architectures tested with the 3D subject-level approach, and as this search method is time costly, it was not used for the 3D patch-level approach for which only four different architectures were tested:

[2] http://www.fil.ion.ucl.ac.uk/spm/software/spm12
[3] https://github.com/vfonov/deep-qc

- 4 convolutional blocks + 2 FC layers
- 4 convolutional blocks + 1 FC layer
- 7 convolutional blocks + 2 FC layers
- 7 convolutional blocks + 1 FC layer

The best architecture (4 convolutional blocks + 2 FC layers) was used for both the 3D patch-level and ROI-based approaches. Note that the other architectures were only slightly worse.

For these 3 approaches, other architecture hyperparameters were explored: with or without batch normalization, with or without dropout.

For the 2D slice-level approach, we chose to use a classical architecture, the ResNet-18 with FC layers added at the end of the network. We explored from 1 to 3 added FC layers and the best results were obtained with one. We then explored the number of layers to fine-tune (2 FC layers or the last residual block + 2 FC layers) and chose to fine-tune the last block and the 2 FC layers. We always used dropout and tried different dropout rates.

For all four approaches, training hyperparameters (learning rate, weight decay) were adapted for each model depending on the evolution of the training accuracy.

The list of the chosen architecture hyperparameters is given in appendix in Tables B.1, B.2 and B.3. The list of the chosen training hyperparameters is given in appendix in Tables B.4 and B.5.

3D subject-level CNN

For the 3D subject-level approach, the proposed CNN architecture is shown in Figure 3.1. The CNN consisted of 5 convolutional blocks and 3 FC layers. Each convolutional block was sequentially made of one convolutional layer, one batch normalization layer, one ReLU and one max pooling layer (more architecture details are provided in appendix Table B.1).

3D ROI-based and 3D patch-level CNN

For the 3D ROI-based and 3D patch-level approaches, the chosen CNN architecture, shown in Figure 3.2, consisted of 4 convolutional blocks (with the same structure as in the 3D subject-level) and 3 FC layers (more architecture details are provided in appendix Table B.2).

To extract the 3D patches, a sliding window (50×50×50 mm3) without overlap was used to convolve over the entire image, generating 36 patches for each image.

For the 3D ROI-based approach, we chose the hippocampus as a ROI, as done in previous studies. We used a cubic patch (50×50×50 mm3) enclosing the left (resp. right) hippocampus. The center of this cubic patch was manually chosen based on the MNI template image (ICBM 2009c nonlinear symmetric template). We ensured visually that this cubic patch included all the hippocampus.

For the 3D patch-level approach, two different training strategies were considered. First, all extracted patches were fitted into a single CNN (denoting this approach as 3D

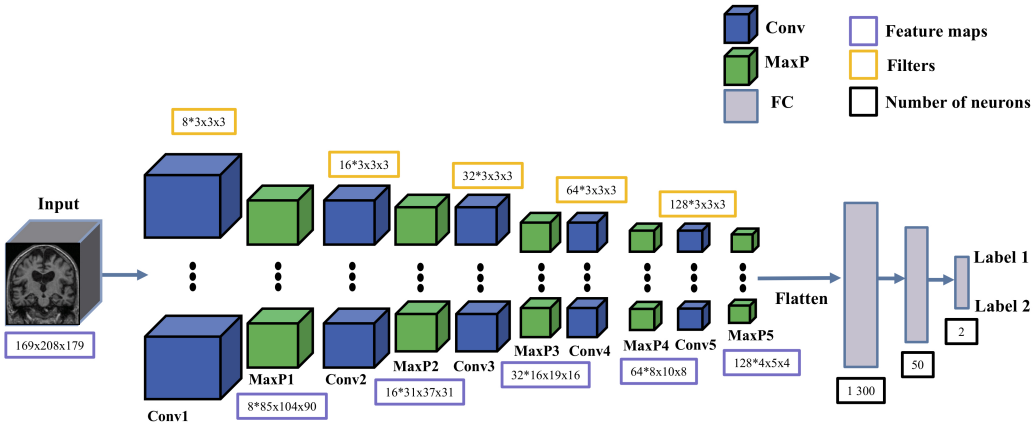


FIGURE 3.1: Architecture of the 3D subject-level CNN. For each convolutional block, we only display the convolutional and max pooling layers. Filters for each convolutional layer represent the number of filters * filter size. Feature maps of each convolutional block represent the number of feature maps * size of each feature map. Conv: convolutional layer; MaxP: max pooling layer; FC: fully connected layer.

patch-level single-CNN). Secondly, we used one CNN for each patch, resulting in finally 36 (number of patches) CNNs (denoting this approach as 3D patch-level multi-CNN).

2D slice-level CNN

For the 2D slice-level approach, the ResNet pre-trained on ImageNet was adopted and fine-tuned. The architecture is shown in Section 3.3. The architecture details of ResNet can be found in (He et al., 2016). We added one FC layer on top of the ResNet (more architecture details are provided in appendix Table B.3). The last five convolutional layers and the last FC layer of the ResNet, as well as the added FC layer, were fine-tuned. The weight and bias of the other layers of the CNN were frozen during fine-tuning to avoid overfitting.

For each subject, each sagittal slice was extracted and replicated into R, G and B channels respectively, in order to generate a RGB image. The first and last twenty slices were excluded due to the lack of information, which resulted in 129 RGB slices for each image.

Majority voting system

For 3D patch-level, 3D ROI-based and 2D slice-level CNNs, we adopted a soft voting system (Raschka, 2015) to generate the subject-level decision. The subject-level decision is generated based on the decision for each slice (resp. for each patch for 3D patch-level / resp. for the left and right hippocampus for ROI-based). More precisely, it was computed based on the predicted probability p obtained after softmax normalization of the outputs of all the slices/patches/ROIs from the same patient:

$$\hat{y} = \underset{i}{\operatorname{argmax}} \, w_j \, p_{\mathrm{ij}}$$

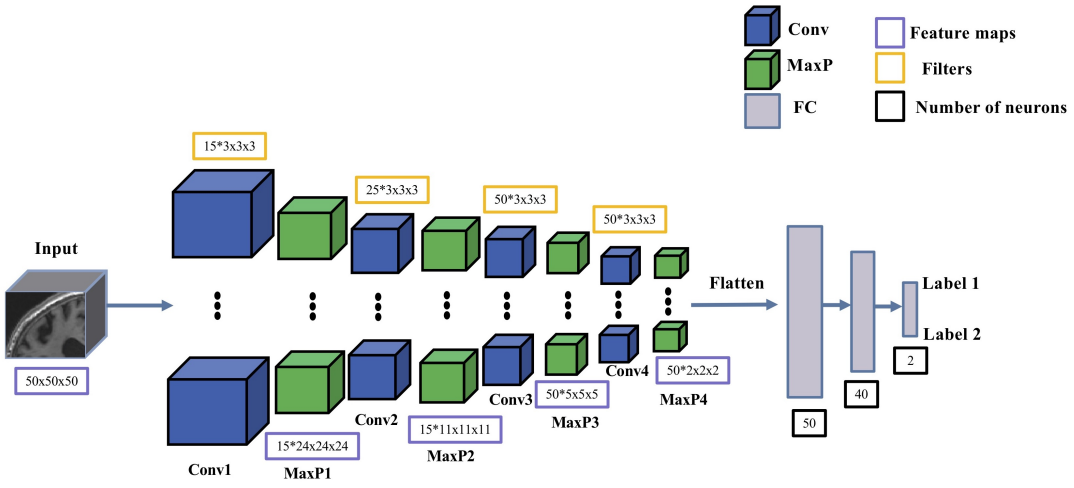


FIGURE 3.2: Architecture of the 3D ROI-based and 3D patch-level CNNs. For each convolutional block, we only display the convolutional and max pooling layers. Filters for each convolutional layer represent the number of filters * filter size. Feature maps of each convolutional block represent the number of feature maps * size of each feature map. Conv: convolutional layer; MaxP: max pooling layer; FC: fully connected layer.

where w_j is the weight assigned to the j-th patch/slice/ROI. w_j reflects the importance of each slice/patch/ROI and is weighted by the normalized accuracy of the j-th slice/patch/ROI. For the evaluation on the test sets, the weights computed on the validation set were used. Note that the predicted probability p is not calibrated and should be interpreted with care as it is not reflective of the true underlying probability of the sample applied to CNNs (Guo et al., 2017; Kuhn and Johnson, 2013).

For the 3D patch-level multi-CNN approach, the 36 CNNs were trained independently. In this case, the weaker classifiers' weight (balanced accuracy < 0.7) was set to be 0 with the consideration that the labels' probabilities of these classifiers could harm the majority voting system.

Comparison to a linear SVM on voxel-based features

For comparison purposes, classification was also performed with a linear SVM classifier. We chose the linear SVM as we previously showed that it obtained higher or at least comparable classification accuracy compared to other conventional models (logistic regression and random forest) (Samper-González et al., 2018). Moreover, given the very high-dimensionality of the input, a non-linear SVM, e.g. with a radial basis function kernel, may not be advantageous since it would only transport the data into an even higher dimensional space. The SVM took as input the modulated gray matter density maps non-linearly registered to the MNI space using the DARTEL method (Ashburner, 2007), as in (Samper-González et al., 2018).

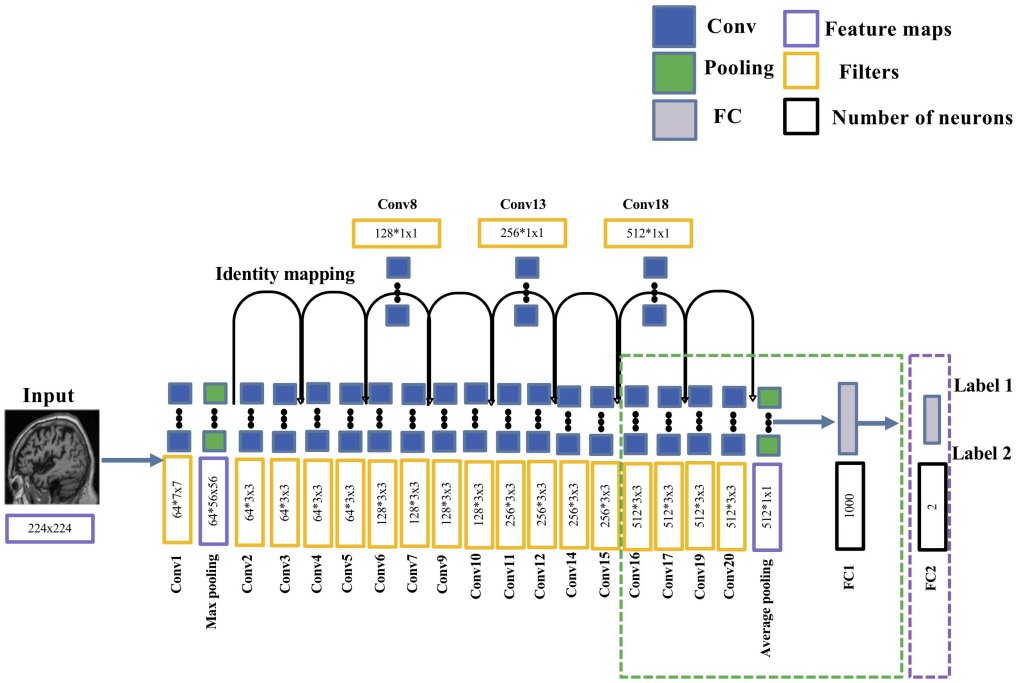


FIGURE 3.3: Architecture of the 2D slice-level CNN. An FC layer (FC2) was added on top of the ResNet. The last five convolutional layers and the last FC of ResNet (green dotted box) and the added FC layer (purple dotted box) were fine-tuned and the other layers were frozen during training. Filters for each convolutional layer represent the number of filters * filter size. Feature maps of each convolutional block represent the number of feature maps * size of each feature map. Conv: convolutional layer; FC: fully connected layer.

3.3.4 Transfer learning

Two different approaches were used for transfer learning: i) AE pre-training for 3D CNNs; and ii) ResNet pre-trained on ImageNet for 2D CNNs.

AE pre-training

An AE was designed based on the architecture of the classification CNN it initializes. The encoder part of the AE is composed of the same sequence of convolutional blocks as the corresponding CNN. Each block has one convolutional layer, one batch normalization layer, one ReLU and one max pooling layer. The architecture of the decoder mirrored that of the encoder, except that the order of the convolution layer and the ReLU was swapped. Of note, the pre-training with AE and classification with CNNs in our experiments used the same training and validation data splits in order to avoid potential data leakage problems. Also, each AE was trained on all available data in the training sets. This means that all MCI, AD and CN subjects in the training data set were used to train the AE.

ImageNet pre-training

For the 2D slice-level experiments, we investigated the possibility to transfer a ResNet pre-trained on ImageNet (He et al., 2016) to our specific tasks. Next, the fine-tuning procedure

was performed on some of the final layers (see Figure 3.3).

3.3.5 Classification tasks

We performed two tasks in our experiments. AD vs CN was used as baseline task to compare the results of our different frameworks. Then the best frameworks were selected to perform the prediction task sMCI vs pMCI: the weights and biases of the model learned on the source task (AD vs CN) were transferred to a new model fine-tuned on the target task (sMCI vs pMCI). For the SVM, the sMCI vs pMCI experiment was performed either by training directly on sMCI vs pMCI or by training on AD vs CN and applying the trained model to sMCI vs pMCI.

3.3.6 Evaluation strategy

Validation procedure

Rigorous validation is essential to objectively assess the performance of a classification framework. This is particularly critical in the case of deep learning as one may easily overfit the validation data set when manually performing model selection and hyperparameter fine-tuning. An independent test set should be, at the very beginning, generated and concealed. It should not be touched until the cross-validation, based on the training and validation data sets, is finished and the final model is chosen. This test data set should be used only to assess the performance (i.e. generalization) of a fully specified and trained classifier (Kriegeskorte et al., 2009; Ripley, 1996; Sarle, 1997). Considering this, we chose a classical split into training/validation/test sets. Training/validation sets were used in a cross-validation procedure for model selection while the test set was left untouched until the end of the peer-review process. Only the best performing model for each approach (3D subject-level, 3D patch-level, 3D ROI-based, 2D slice-level), as defined by the cross-validation on training/validation sets, was tested on the test set.

The ADNI test set consisted of 100 randomly chosen age- and sex-matched subjects for each diagnostic class (i.e. 100 CN subjects, 100 AD patients). The rest of the ADNI data was used as training/validation set. We ensured that age and sex distributions between training/validation and test sets were not significantly different. Two other test sets were composed of all subjects of OASIS and AIBL. The ADNI test set is used to assess model generalization within the same data set (thereby assessing that the model has not overfitted the training/validation set). The AIBL test set is used to assess generalization to another data set that has similar inclusion criteria and image acquisition parameters to those of the training set. The OASIS test is used to assess generalization to a data set with different inclusion criteria and image acquisition parameters. As mentioned above, it is important to note that the diagnosis labels are not based on the same criteria in OASIS on the one hand and ADNI/AIBL on the other. Thus we do not hypothesize that the models trained on ADNI will generalize well to OASIS.

The model selection procedure, including model architecture selection and training hyperparameter fine-tuning, was performed using only the training/validation data set.

For that purpose, a 5-fold cross-validation was performed, which resulted in one fold (20%) of the data for validation and the rest for training. Note that the 5-fold data split was performed only once for all the experiments with a fixed seed number (random_state = 2), thus guaranteeing that all the experiments used exactly the same subjects during cross-validation. Also, no overlapping exists between the MCI subjects used for AE pre-training (using all available AD, CN and MCI) and the test data set of sMCI vs pMCI. Thus, the evaluation of the cross-task transfer learning (from AD vs CN to sMCI vs pMCI) is unbiased. Finally, for the linear SVM, the hyperparameter C controlling the amount of regularization was chosen using an inner loop of 10-fold cross-validation (thereby performing a nested cross-validation).

Metrics

We computed the following performance metrics: balanced accuracy (BA), area under the receiver operating characteristic (ROC) curve (AUC), accuracy, sensitivity and specificity. In the manuscript, for the sake of concision, we report only the BA but all other metrics are available on Zenodo under the DOI 10.5281/zenodo.3491003.

3.3.7 Implementation details

The image preprocessing procedures were implemented with Nipype (Gorgolewski et al., 2011). The deep learning models were built using the Pytorch library[4] (Paszke et al., 2017). TensorboardX[5] was embedded into the current framework to dynamically monitor the training process. Specifically, we evaluated and reported the training and validation BA/loss after each epoch or certain iterations. Of note, instead of using only the current batch of data, the BA was evaluated based on all the training/validation data. Moreover, we organized the classification outputs in a hierarchical way inspired from BIDS, including the TSV files containing the classification results, the outputs of TensorboardX for dynamic monitoring of the training and the best performing models selected based on the validation BA. The linear SVM was implemented using scikit-learn (Pedregosa et al., 2011; Samper-González et al., 2018).

We applied the following early stopping strategy for all the classification experiments: the training procedure does not stop until the validation loss is continuously higher than the lowest validation loss for N epochs. Otherwise, the training continues to the end of the pre-defined number of epochs. The selected model was the one which obtained the highest validation BA during training. For the AE pre-training, the AE was trained to the end of the pre-defined number of epochs. We then visually check the validation loss and the quality of the reconstructed images. The mean square loss was used for the AE pre-training and the cross-entropy loss, which combines a log softmax normalization and the negative log likelihood loss, was used for the CNNs.

[4] https://pytorch.org/
[5] https://github.com/lanpa/tensorboardX

3.4 Experiments and results

3.4.1 Results on training/validation set

The different classification experiments and results (validation BA during 5-fold cross-validation) are detailed in Table 3.4.

3D subject-level

Influence of intensity rescaling We first assessed the influence of intensity rescaling. Without rescaling, the CNN did not perform better than chance (BA = 0.50) and there was an obvious generalization gap (high training but low validation BA). With intensity rescaling, the BA improved to 0.80. Based on these results, intensity rescaling was used in all subsequent experiments.

Influence of transfer learning (AE pre-training) The performance was slightly higher with AE pre-training (0.82) than without (0.80). Based on this, we decided to always use AE pre-training, even though the difference is small.

Influence of the training data set size We then assessed the influence of the amount of training data, comparing training using only baseline data to those with longitudinal data. The performance was moderately higher with longitudinal data (0.85) compared to baseline data only (0.82). We choose to continue exploring the influence of this choice because the four different approaches have a very different number of learnt parameters and the sample size is intrinsically augmented in 2D slice-level and 3D single-CNN patch-level approaches.

Influence of preprocessing We then assessed the influence of the preprocessing comparing the "Extensive" and "Minimal" preprocessing procedures. The performance was almost equivalent with the "Minimal" preprocessing (0.85) and with the "Extensive" preprocessing (0.86). Hence in the following experiments we kept the "Minimal" preprocessing.

Classification of sMCI vs pMCI The BA was the same for baseline data and for longitudinal data (0.73).

3D ROI-based

For AD vs CN, the BA was 0.88 for baseline data and 0.86 for longitudinal data. This is slightly higher than that of the subject-level approach. For sMCI vs pMCI, the BA was 0.77 for baseline data and 0.78 for longitudinal data. This is substantially higher than with the 3D subject-level approach.

3D patch-level

Single CNN For AD vs CN, the BA was 0.74 for baseline data and 0.76 for longitudinal data.

Multi CNN For AD vs CN, the BA was 0.81 for baseline data and 0.83 for longitudinal data, thereby outperforming the single CNN approach. For sMCI vs pMCI, the BA was 0.75 for baseline data and 0.77 for longitudinal data. The performance for both tasks is slightly lower than that of the 3D ROI-based approach.

2D slice-level

In general, the performance of the 2D slice-level approach was lower to that of the 3D ROI-based, 3D patch-level multi CNN and 3D subject-level (when trained with longitudinal data) approaches but higher than that of the 3D patch-level single CNN approach. For 2D slice-level, the use of longitudinal data for training did not improve the performance (0.79 for baseline data; 0.74 for longitudinal data). Finally, we studied the influence of data leakage using a slice-level data split strategy. As expected, the BA was 1.00.

Linear SVM

For task AD vs CN, the balanced accuracies were 0.88 when trained with baseline data and 0.87 when trained with longitudinal data. For task sMCI vs pMCI, when training from scratch, the balanced accuracies were 0.68 when trained with baseline data and 0.68 when trained with longitudinal data. When using transfer learning from the task AD vs CN to the task sMCI vs pMCI, the balanced accuracies were 0.70 (when trained with baseline data) and 0.70 (when trained with longitudinal data). The performance of the SVM on AD vs CN is thus higher than that of most deep learning models and comparable to the best ones. Whereas for task sMCI vs pMCI, the BA of the SVM is lower than that of deep learning models.

3.4.2 Results on the test sets

Results on the three test sets (ADNI, OASIS and AIBL) are presented in Table 3.5. For each category of approach, we only applied the best models for both baseline and longitudinal data.

3D subject-level

For AD vs CN, all models generalized well to the ADNI and AIBL test sets but not to the OASIS test set (losing over 0.15 points of BA).

For sMCI vs pMCI, the models generalized relatively well to the ADNI test set but not to the AIBL test set (losing over 0.20 points). Note that the generalization was better for longitudinal than for baseline.

Classification architectures	Training data	Image preprocessing	Intensity rescaling	Data split	Training approach	Transfer learning	Task	Validation balanced accuracy	Exp #
3D subject-level CNN	Baseline	Minimal	None	subject-level	single-CNN	None	AD vs CN	0.50 (0.00) [0.50, 0.50, 0.50, 0.50, 0.50]	1
		MinMax						0.80 (0.05) [0.76, 0.86, 0.81, 0.85, 0.74]	2
						AE pre-train		0.82 (0.05) [0.74, 0.90, 0.83, 0.77, 0.83]	3
	Longitudinal	Minimal	MinMax					0.85 (0.04) [0.88, 0.88, 0.84, 0.85, 0.78]	4
		Extensive						0.86 (0.06) [0.88, 0.94, 0.85, 0.85, 0.76]	5
		Minimal					sMCI vs pMCI	0.73 (0.03) [0.73, 0.73, 0.67, 0.76, 0.74]	6
	Baseline							0.73 (0.05) [0.73, 0.73, 0.63, 0.77, 0.76]	7
3D ROI-based CNN	Baseline	Minimal	MinMax	subject-level	single-CNN	AE pre-train	AD vs CN	0.88 (0.03) [0.84, 0.89, 0.90, 0.89, 0.85]	8
							sMCI vs pMCI	0.77 (0.05) [0.81, 0.81, 0.67, 0.78, 0.76]	9
	Longitudinal						AD vs CN	0.86 (0.02) [0.83, 0.86, 0.86, 0.88, 0.86]	10
							sMCI vs pMCI	0.78 (0.07) [0.87, 0.73, 0.68, 0.82, 0.78]	11

Classification architectures	Training data	Image preprocessing	Intensity rescaling	Data split	Training approach	Transfer learning	Task	Validation balanced accuracy	Exp #
3D patch-level CNN	Baseline	Minimal	MinMax	subject-level	single-CNN	AE pre-train	AD vs CN	0.74 (0.08) [0.75, 0.84, 0.78, 0.75, 0.59]	12
	Longitudinal							0.76 (0.04) [0.78, 0.77, 0.80, 0.78, 0.69]	13
	Baseline				multi-CNN		AD vs CN	0.81 (0.03) [0.82, 0.84, 0.83, 0.77, 0.79]	14
							sMCI vs pMCI	0.75 (0.04) [0.80, 0.72, 0.72, 0.79, 0.72]	15
	Longitudinal						AD vs CN	0.83 (0.02) [0.83, 0.85, 0.84, 0.82, 0.79]	16
							sMCI vs pMCI	0.77 (0.04) [0.77, 0.75, 0.71, 0.82, 0.79]	17
2D slice-level CNN	Baseline	Minimal	MinMax	subject-level	single-CNN	ImageNet pre-train	AD vs CN	0.79 (0.04) [0.83, 0.83, 0.72, 0.82, 0.73]	18
	Longitudinal							0.74 (0.03) [0.76, 0.80, 0.74, 0.71, 0.69]	19
	Baseline			slice-level				1.00 (0) [1.00, 1.00, 1.00, 1.00, 1.00]	20

Classification architectures	Training data	Image preprocessing	Intensity rescaling	Data split	Training approach	Transfer learning	Task	Validation balanced accuracy	Exp #
SVM	Baseline	DartelGM	SPM-based	subject-level	None	None	AD vs CN	0.88 (0.02) [0.92, 0.89, 0.85, 0.89, 0.84]	21
							sMCI vs pMCI (trained on sMCI vs pMCI)	0.68 (0.02) [0.71, 0.68, 0.66, 0.67, 0.71]	22
							sMCI vs pMCI (trained on AD vs CN)	0.70 (0.06) [0.66, 0.75, 0.70, 0.79, 0.63]	23
	Longitudinal						AD vs CN	0.87 (0.01) [0.86, 0.86, 0.88, 0.87, 0.85]	24
							sMCI vs pMCI (trained on sMCI vs pMCI)	0.68 (0.06) [0.75, 0.77, 0.62, 0.62, 0.67]	25
							sMCI vs pMCI (trained on AD vs CN)	0.70 (0.02) [0.68, 0.72, 0.67, 0.69, 0.73]	26

TABLE 3.4: Summary of all the classification experiments and validation results in our analyses. For each model, we report the balanced accuracy for each of the five folds within square brackets and the average and standard-deviation across the folds. Note that this is not the standard-deviation of the estimator of balanced accuracy.

MinMax: for CNNs, intensity rescaling was done based on min and max values, resulting all values to be in the range of [0, 1]; SPM-based: the SPM-based gray matter maps are intrinsically rescaled; AE: autoencoder. For DL models, sMCI vs pMCI tasks were done with as follows: the weights and biases of the model learnt on the source task (AD vs CN) were transferred to a new model fine-tuned on the target task (sMCI vs pMCI). For SVM, the sMCI vs pMCI was done either training directly on sMCI vs pMCI or using training on AD vs CN and applying the trained model to sMCI vs pMCI.

3D ROI-based

For AD vs CN, the models generalized well to the ADNI test set, slightly worse to the AIBL test set (losing 0.04 to 0.05 points) and considerably worse for OASIS (losing from 0.13 to 0.19 points).

For sMCI vs pMCI, there was a slight decrease in BA on the ADNI test set and a severe decrease for the AIBL test set. Note that on the ADNI test set, the performance of the 3D ROI-based is almost the same as that of the 3D subject-level (when using longitudinal data) while it was better on the validation set.

3D patch-based

For AD vs CN, the generalization pattern was similar to that of the other models: good for ADNI and AIBL, poor for OASIS.

For sMCI vs pMCI, the BA on the ADNI test set was 0.07 points lower than on the ADNI validation set. The BA on the AIBL test set was very poor.

2D slice-level

For AD vs CN, there was a slight decrease in performance on the ADNI test set (losing from 0 to 0.03 points) and the AIBL test set (losing from 0.01 to 0.03 points) and a considerable decrease on the OASIS test set (losing from 0.13 to 0.14 points). As expected, the "data-leakage" model did not generalize well.

Linear SVM

For AD vs CN, we observed the same pattern as for the other models: excellent generalization to ADNI and AIBL but not to OASIS.

For sMCI vs pMCI, the generalization was excellent for ADNI but not for AIBL. Of note, the BA on the ADNI test set was even higher to that of the validation, reaching a level which is comparable to the best deep learning models.

3.5 Discussion

In this chapter we rigorously compared different CNN approaches and studied the impact of key components on the performance. We hope that these results will provide a more objective assessment of the performance of CNNs for AD classification and constitute a solid baseline for future research.

The proposed framework was applied to images from three public data sets, ADNI, AIBL and OASIS. On the ADNI test data set, the diagnostic BA of CNNs ranged from 0.76 to 0.89 for the AD vs CN task and from 0.69 to 0.74 for the sMCI vs pMCI task. These results are in line with the state-of-the-art (studies without data leakage in Table 1.1), where classification accuracy typically ranged from 0.76 to 0.91 for AD vs CN and 0.62 to 0.83 for sMCI vs pMCI. Nevertheless, the performance that we report is lower than that of the

Classification architectures	Training data	Image preprocessing	Intensity rescaling	Data split	Training approach	Transfer learning	Task	Validation BA	ADNI test BA	AIBL test BA	OASIS test
3D subject-level CNN	Baseline	Minimal	MinMax	subject-level	single-CNN	AE pre-train	AD vs CN	0.82 (0.05)	**0.82** [0.79, 0.85, 0.82, 0.81, 0.85]	**0.83** [0.81, 0.85, 0.84, 0.78, 0.86]	**0.67** [0.59, 0.69, 0.72, 0.64, 0.69]
	Longitudinal							0.85 (0.04)	**0.85** [0.88, 0.84, 0.84, 0.84, 0.84]	**0.86** [0.89, 0.85, 0.86, 0.85, 0.86]	**0.68** [0.65, 0.70, 0.70, 0.71, 0.65]
	Baseline						sMCI vs pMCI	0.73 (0.05)	**0.69** [0.68, 0.71, 0.64, 0.73, 0.67]	**0.52** [0.51, 0.47, 0.55, 0.54, 0.55]	–
	Longitudinal							0.73 (0.03)	**0.73** [0.75, 0.72, 0.72, 0.74, 0.72]	**0.50** [0.48, 0.47, 0.54, 0.52, 0.51]	–
3D ROI-based CNN	Baseline	Minimal	MinMax	subject-level	single-CNN	AE pre-train	AD vs CN	0.88 (0.03)	**0.89** [0.87, 0.88, 0.90, 0.91, 0.89]	**0.84** [0.83, 0.88, 0.84, 0.85, 0.83]	**0.69** [0.62, 0.74, 0.70, 0.69, 0.71]
	Longitudinal							0.86 (0.02)	**0.85** [0.87, 0.82, 0.87, 0.86, 0.87]	**0.81** [0.79, 0.81, 0.79, 0.82, 0.85]	**0.73** [0.71, 0.73, 0.72, 0.76, 0.71]
	Baseline						sMCI vs pMCI	0.77 (0.05)	**0.74** [0.75, 0.72, 0.76, 0.75, 0.75]	**0.60** [0.56, 0.56, 0.66, 0.62, 0.59]	–
	Longitudinal							0.78 (0.07)	**0.74** [0.70, 0.73, 0.73, 0.75, 0.81]	**0.57** [0.56, 0.53, 0.52, 0.66, 0.56]	–
3D patch-level CNN	Baseline	Minimal	MinMax	subject-level	multi-CNN	AE pre-train	AD vs CN	0.81 (0.03)	**0.81** [0.82, 0.81, 0.84, 0.80, 0.79]	**0.81** [0.81, 0.75, 0.81, 0.84, 0.82]	**0.64** [0.61, 0.65, 0.60, 0.69, 0.67]
	Longitudinal							0.83 (0.02)	**0.86** [0.86, 0.86, 0.87, 0.85, 0.84]	**0.80** [0.82, 0.78, 0.81, 0.81, 0.79]	**0.71** [0.70, 0.70, 0.71, 0.71, 0.67]
	Baseline						sMCI vs pMCI	0.75 (0.04)	**0.68** [0.71, 0.64, 0.64, 0.71, 0.69]	**0.64** [0.63, 0.52, 0.67, 0.74, 0.63]	–
	Longitudinal							0.77 (0.04)	**0.70** [0.70, 0.71, 0.69, 0.71, 0.69]	**0.44** [0.45, 0.39, 0.55, 0.42, 0.39]	–

Classification architectures	Training data	Image preprocessing	Intensity rescaling	Data split	Training approach	Transfer learning	Task	Validation BA	ADNI test BA	AIBL test BA	OASIS test
2D slice-level CNN	Baseline	Minimal	MinMax	subject-level	single-CNN	ImageNet pre-train	AD vs CN	0.79 (0.04)	**0.76** [0.76, 0.75, 0.77, 0.75, 0.78]	**0.76** [0.74, 0.76, 0.78, 0.75, 0.75]	**0.65** [0.67, 0.62, 0.64, 0.65, 0.69]
	Longitudinal							0.74 (0.03)	**0.74** [0.81, 0.76, 0.70, 0.74, 0.72]	**0.73** [0.72, 0.77, 0.72, 0.66, 0.79]	**0.61** [0.62, 0.63, 0.64, 0.58, 0.60]
	Baseline			slice-level				1.00 (0)	**0.75** [0.74, 0.76, 0.75, 0.76, 0.75]	**0.80** [0.80, 0.79, 0.82, 0.80, 0.81]	**0.68** [0.68, 0.67 0.69, 0.70, 0.66]
SVM	Baseline	DartelGM	SPM-based	subject-level	None	None	AD vs CN	0.88 (0.02)	**0.88** [0.88, 0.87, 0.90, 0.90, 0.88]	**0.88** [0.87, 0.90, 0.87, 0.89, 0.90]	**0.70** [0.71, 0.71, 0.70, 0.68, 0.72]
	Longitudinal							0.87 (0.01)	**0.87** [0.85, 0.84, 0.90, 0.89, 0.87]	**0.87** [0.88, 0.86, 0.88, 0.87, 0.89]	**0.71** [0.73, 0.68, 0.72, 0.70, 0.71]
	Baseline						sMCI vs pMCI (trained on AD vs CN)	0.70 (0.06)	**0.75** [0.75, 0.75, 0.74, 0.76, 0.76]	**0.60** [0.62, 0.54, 0.62, 0.59, 0.64]	–
	Longitudinal							0.70 (0.02)	**0.76** [0.74, 0.75, 0.80, 0.77, 0.76]	**0.68** [0.67, 0.66, 0.68, 0.67, 0.71]	–

TABLE 3.5: Summary of the results of the three test datasets in our analyses. 3D subject-level CNNs were trained using intensity rescaling and our "Minimal" preprocessing, with a data split on the subject level and transfer learning (AE pretraining for AD vs CN tasks and cross-task transfer learning was applied for sMCI vs pMCI tasks). For each model, we first copied the validation balanced accuracy (mean and standard deviation across the five folds) that is reported in Table 3.4. Then, we report the balanced accuracy for each test set (ADNI, AIBL, OASIS), more specifically within square brackets we report the balanced accuracy for each of the trained models of the 5 folds of the validation set and then the average across the five folds.

MinMax: for CNNs, intensity rescaling was done based on min and max values, resulting all values to be in the range of [0, 1]; SPM-based: the SPM-based gray matter maps are intrinsically rescaled; AE: autoencoder.

top-performing studies. This potentially comes from the fact that our test set was fully independent and was never used to choose the architectures or parameters. The proposed framework can be used to provide a baseline performance when developing new methods.

One interesting question is whether deep learning could perform better than conventional machine learning methods for AD classification. Here, we chose to compare CNN to a linear SVM. SVM has been used in many AD classification studies and obtained competitive balanced accuracies (Falahati et al., 2014; Haller et al., 2011; Rathore et al., 2017). In the current study, the SVM was at least as good as the best CNNs for both the AD vs CN and the sMCI vs pMCI task. Note that we used a standard linear SVM with standard voxel-based features. It could be that more sophisticated conventional machine learning methods could provide even higher performance. Similarly, we do not claim that more sophisticated deep learning architectures would not outperform the SVM. However, this is not the case with the architectures that we tested, which are representative of the existing literature on AD classification. Besides, it is possible that CNNs will outperform SVM when larger public data sets will become available. Overall, a major result of the present paper is that, with the sample size which is available in ADNI, CNNs did not provide an increase in performance compared to SVM.

Unbiased evaluation of the performance is an essential task in machine learning. This is particularly critical for deep learning because of the extreme flexibility of the models and of the numerous architecture and training hyperparameters that can be chosen. In particular, it is crucial that such choices are not made using the test set. We chose a very strict validation strategy in that respect: the test sets were left untouched until the end of the peer-review process. This guarantees that only the final models, after all possible adjustments, are carried to the test set. Moreover, it is important to assess generalization not only to unseen subjects but also to other studies in which image acquisitions or patient inclusion criteria can vary. In the present paper, we used three test sets from the ADNI, AIBL and OASIS databases to assess different generalization aspects.

We studied generalization in three different settings: i) on a separate test set from ADNI, thus from the same study as those of the training set; ii) on AIBL, i.e. a different study but with similar inclusion criteria and imaging acquisitions; iii) on OASIS, i.e. a study with different inclusion criteria and imaging acquisitions. Overall, the models generalized well to ADNI (for both tasks) and to AIBL (for AD vs CN). On the other hand, we obtained a very poor generalization to sMCI vs pMCI for AIBL. We hypothesize that it could be because pMCI and sMCI participants from AIBL are substantially older than those of ADNI, which is not the case for AD and CN participants. Nevertheless, note that the sample size for sMCI vs pMCI in AIBL is quite small (33 participants). Also, the generalization to OASIS was poor. This may stem from the diagnosis criteria which are less rigorous (in OASIS, all participants with CDR>0 are considered AD). Overall, these results bring important information. First, good generalization to unseen, similar, subjects demonstrate that the models did not overfit the subjects at hand in the training/validation set. On the other hand, poor generalization to different age, protocols and inclusion criteria show that trained models are too specific of these characteristics. Generalization across different populations

thus remains an unsolved problem and will require training on more representative data sets but maybe also new strategies to make training more robust to heterogeneity. This is critical for the future translation to clinical practice in which conditions are much less controlled than in research data sets like ADNI. Moreover, we showed with some additional experiments that the distribution of field strength in ADNI may have biased some of our tasks, but also the rest of the literature (see Appendix A).

We studied the influence of several key choices on the performance. First, we studied the influence of AE pre-training and showed that it slightly improved the average over training from scratch. Three previous papers studied the impact of AE pre-training (Hosseini-Asl et al., 2016; Vu et al., 2017, 2018) and found that it improved the results. However, they are all suspected of data leakage. We thus conclude that, to date, it is not proven that AE pre-training leads to a significant increase in BA. A difficulty in AD classification using deep learning is the limited amount of data samples available for training. However, training with longitudinal instead of baseline data gave only a slight increase of BA in most approaches. The absence of a major improvement may be due to several factors. First, training with longitudinal data implies training with data from more advanced disease stages, since patients are seen at a later point in the disease course. This may have an adverse effect on the performance of the model when tested on baseline data, at which the patients are less advanced. Also, since the additional data come from the same patients, this does not provide a better coverage of inter-individual variability. We studied the impact of image preprocessing. First, as expected, we found that CNNs cannot be successfully trained without intensity rescaling. We then studied the influence of two different preprocessing procedures ("Minimal" and "Extensive"). The "Minimal" procedure is limited to an affine registration of the subject's image to a standard space, while for the "Extensive" procedure non-linear registration and skull stripping are performed. They led to comparable results. In principle, this is not surprising as deep learning methods do not require extensive preprocessing. In the litterature, varied types of preprocessing have been used. Some studies used non-linear registration (Bäckström et al., 2018; Basaia et al., 2019; Lian et al., 2018; Lin et al., 2018; Liu et al., 2018c,e; Wang et al., 2019; Wang et al., 2018a) while others used only linear (Aderghal et al., 2018; Aderghal et al., 2017a,b; Hosseini Asl et al., 2018; Li et al., 2018; Liu et al., 2018b; Shmulev et al., 2018) or no registration (Cheng and Liu, 2017). None of them compared these different preprocessings with the exception of (Bäckström et al., 2018) which compared preprocessing using FreeSurfer to no preprocessing. They found that training the network with the raw data resulted in a lower classification performance (drop in accuracy of 38 percent points) compared to the preprocessed data using FreeSurfer (Bäckström et al., 2018). However, FreeSurfer comprises a complex pipeline with many preprocessing steps so it is unclear, from their results, which part drives the superior performance, while we clearly demonstrated that the intensity rescaling is essential for the CNN training whereas there is no improvement in using a non-linear registration over a linear one. Finally, we found that, for the 3D patch-level framework, the multi-CNN approach gave better results than the single-CNN one. However, this may be mainly because the multi-CNN approach benefits from a

thresholding system which excludes the worst patches, a system that was not present in the single-CNN approach. To test this hypothesis, we performed supplementary experiments in which the multi-CNN was trained without threshold and the single-CNN was trained using the same thresholding system as in the main experiments of the multi-CNN. Results are reported in Tables B.6 and B.7. We observed that the results of the multi-CNN and the single-CNN are comparable when they use the same thresholding system. For example, for the AD vs CN task, without thresholding, the BA of the multi-CNN was 0.76 using baseline data and 0.72 using longitudinal data while that of the single-CNN were respectively 0.74 and 0.76. A similar observation can be made when both approaches used the thresholding. These supplementary experiments suggest that, under similar conditions, the multi-CNN architecture does not always perform better than the single-CNN architecture. In light of this, it would seem preferable to choose a framework that offers a better compromise between performance and conceptual complexity, e.g. the ROI-based or the 3D subject-level approaches.

Our study has the following limitations. First, a large number of options exist when choosing the model architecture and training hyperparameters. Even though we did our best to make meaningful choices and test a relatively large number of possibilities, we cannot exclude that other choices could have led to better results. To overcome this limitation, we provided an open-source framework. Researchers can use it to propose and validate potentially better performing models. In particular, with this framework, researchers can easily try their own models without touching the test data sets. Secondly, the cross-validation procedures were performed only once. Of course, the training is not deterministic and one would ideally want to repeat the cross-validation to get a more robust estimate of the performance. However, we did not perform this due to limited computational resources. Finally, overfitting always exists in our experiments, even though different techniques have been tried (e.g. transfer learning, dropout or weight decay). This phenomenon occurs mainly due to the limited size of the data sets available for AD classification. It is likely that training with much larger data sets would result in higher performance.

Chapter 4

Interpretability method to assess the robustness of convolutional neural networks for medical image classification

---

This chapter has been published in the proceedings of SPIE Medical Imaging 2020: Image Processing conference.

- **Title:** Visualization approach to assess the robustness of neural networks for medical image classification
- **Authors:** Elina Thibeau-Sutre, Olivier Colliot, Didier Dormont, Ninon Burgos.
- **DOI:** 10.1117/12.2548952

4.1 Introduction

In this chapter, we adapted the method of Fong and Vedaldi, 2017 to 3D medical images to find on which basis a network classifies quantitative data. Indeed, quantitative data can be obtained from different medical imaging modalities, for example binding potential maps obtained with positron emission tomography (PET) or gray matter probability maps extracted from structural magnetic resonance imaging (MRI).

Our application focuses on the detection of Alzheimer's disease (AD), a neurodegenerative syndrome that induces gray matter atrophy. We used as inputs gray matter probability maps, a proxy for atrophy, extracted from T1-weighted (T1w) MRI. The process includes two distinct parts: first a convolutional neural network (CNN) is trained to classify AD from control subjects, then the weights of the network are fixed and a mask is trained to prevent the network from classifying correctly all the subjects it has correctly classified after training. The goals of this work are to assess whether the interpretability method initially

developed for natural images is suitable for 3D medical images and to exploit it to better understand the decisions taken by classification networks.

4.2 Materials and Methods

4.2.1 Data description and preprocessing

Data used in the preparation of this article were obtained from two public data sets: the Alzheimer's Disease Neuroimaging Initiative (ADNI) database and the Australian Imaging, Biomarkers and Lifestyle (AIBL) study. We used the T1w MRI available in each of these studies. Two diagnosis groups were considered:

- CN: sessions of subjects who were cognitively normal (CN) at baseline and stayed stable during the follow-up;
- AD: sessions of subjects who were diagnosed as AD at baseline and stayed stable during the follow-up.

The populations of ADNI and AIBL are described in Table 4.1.

Data set	Label	Subjects	Sessions	Age	Gender	MMSE	CDR
ADNI	CN	330	1 830	74.4 (5.8) [59.8, 89.6]	160 M / 170 F	29.1 (1.1) [24, 30]	0: 330
	AD	336	1 106	75.0 (7.8) [55.1, 90.9]	185 M / 151 F	23.2 (2.1) [18, 27]	0.5: 160; 1: 175; 2: 1
AIBL	CN	429	730	72.5 (6.2) [60, 92]	183 M / 246 F	28.8 (1.2) [25, 30]	0: 406, 0.5: 22, 1: 1
	AD	76	108	73.9 (8.0) [55, 93]	33 M / 43 F	20.6 (5.5) [6, 29]	0.5: 31; 1: 36; 2: 7, 3: 2

TABLE 4.1: Summary of ADNI and AIBL participant demographics, mini-mental state examination (MMSE) and global clinical dementia rating (CDR) scores at baseline. Values are presented as mean (standard deviation) [range]. M: male, F: female

Preprocessing of T1w MR images was performed with the Clinica software platform (Routier et al., 2021). First the data sets were converted to the BIDS format, then `the t1-volume` preprocessing pipeline of Clinica was applied (Samper-González et al., 2018). This pipeline performs bias field correction, non-linear registration and tissue segmentation using the Unified Segmentation approach (Ashburner and Friston, 2005) available in SPM12. The gray matter maps in MNI space were retrieved for the image analysis.

4.2.2 CNN classification

The following sections describe the evaluation procedure, the hyperparameters selection and implementation details that are linked to the classification of AD vs CN subjects with CNNs. During training, the weights and biases of the network are optimized to maximize the score function f on a set of images X.

Evaluation procedure

The ADNI data set was split into training/validation and test sets. The ADNI test set consisted of 100 randomly chosen age- and sex-matched subjects for each diagnostic class (i.e. 100 CN subjects, 100 AD patients). The rest of the ADNI data set was used as training/validation set. We ensured that age and sex distributions between training/validation and test sets were not significantly different. The model selection procedure, including model architecture selection and training hyperparameter fine-tuning, was performed using only the training/validation data set. For that purpose, a 5-fold cross-validation was performed, which resulted in one fold (20%) of the data for validation and the rest for training. Note that the 5-fold data split was performed only once for all the experiments with a fixed seed number (*random_state = 2*), thus guaranteeing that all the experiments used exactly the same subjects during cross-validation. The AIBL data set was used as an independent test set to assess the CNN generalization ability. Test and validation sets included only one session per subject.

Hyperparameter selection

We performed a random search (Bergstra and Bengio, 2012) to select the architecture and optimization hyperparameters of our CNN. The hyperparameters explored for the architecture were the number of convolutional blocks, of filters in the first layer and of convolutional layers in a block, the dimension reduction strategy (by using a max pooling layer or by setting the stride of the last convolutional layer of the convolutional block to 2), the number of fully-connected layers and the dropout rate. Other hyperparameters such as the learning rate, the weight decay, the batch size, the data preprocessing and the intensity normalization were also part of the search.

Only one experiment was performed per architecture tested using the first split of the cross-validation due to the computational cost of the random search. The chosen architecture was the one that obtained the best balanced accuracy on the validation set. This architecture (displayed in Figure 4.1) is composed of 7 convolutional blocks followed by a dropout layer and a fully-connected layer. Each convolutional block (C1, C2 or C3) is made of 1 to 3 sub-blocks and a max pooling layer with a kernel size and a stride of 2. Each sub-block is composed of a convolutional layer with kernel size of 3, a batchnormalization layer and a leaky ReLU activation. The predicted label of the input image is the class with the highest output probability.

CNN training

The weights of the convolutional and fully connected layers were initialized as described in (He et al., 2015), which corresponds to the default initialization method in PyTorch. We applied the following early stopping strategy for all the classification experiments: the training procedure does not stop until the validation loss is continuously higher than the lowest validation loss for N epochs (N=5); otherwise, the training continues to the end of a pre-defined number of epochs (30). The training and validation loss were computed with

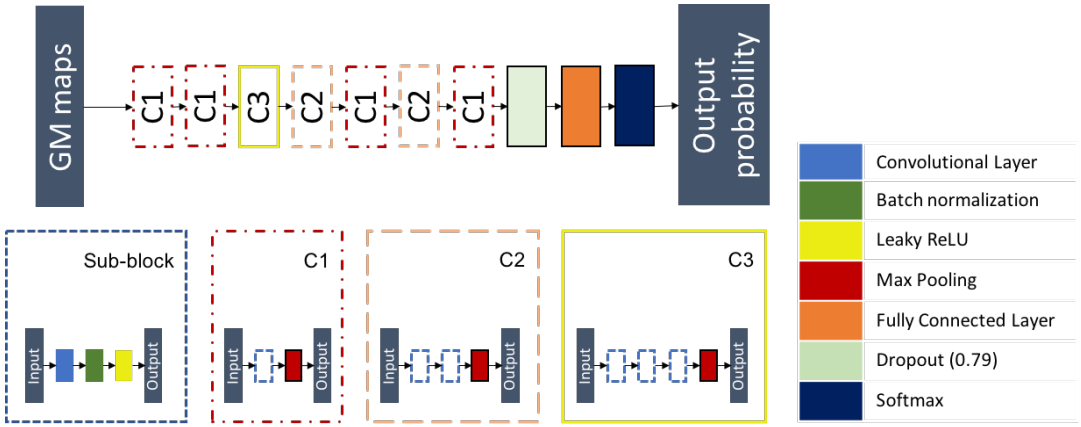


FIGURE 4.1: Architecture of the CNN classifier determined following to a random search procedure.

the cross-entropy loss. For each experiment, the final model was the one that obtained the highest validation balanced accuracy during training. The balanced accuracy of the model was evaluated at the end of each epoch.

4.2.3 Interpretability method

The proposed interpretability method extends the framework of (Fong and Vedaldi, 2017), and corresponds to the optimized perturbation method described in chapter 2.

Once the classification network has been trained, its parameters are fixed to the best value found. Then the interpretability method consists in computing a mask that will overlay the most meaningful parts of an image to prevent the network from classifying it correctly. In the following, the goal is to mask AD images that were correctly classified by the CNN so that it systematically classifies them with the CN label. The mask m is a 3D volume of the same size as the input image and hide parts of the image in a voxel-wise manner. In this application, each voxel u of the input image X will be masked by a constant value μ according to the value of the mask for this voxel. The mask values are included in $[0, 1]$. The masked input image X^m at voxel u is defined as:

$$X^m(u) = m(u)X(u) + (1 - m(u))\mu \tag{4.1}$$

As AD patients suffer from gray matter atrophy, the goal of the masking method would be to artificially simulate gray matter restoration in a minimal number of brain regions to make them look like CN subjects. By setting $\mu = 1$, the mask was trained to artificially increase the probability of gray matter for the minimum set of voxels which will lead to the maximum decrease of the performance of the CNN. The optimal mask m^* is the mask for which the following loss function is minimized:

$$m^* = \underset{m}{\operatorname{argmin}}\, f(X^m) + \lambda_1 \|1 - m_1\|_{\beta_1}^{\beta_1} + \lambda_2 \sum_u \|\nabla m(u)\|_{\beta_2}^{\beta_2} \tag{4.2}$$

The first term prevents the network from finding the correct class when the mask is applied, the second term ensures that a minimum set of voxels is selected, and the third one ensures that the mask is smooth enough and is not made of scattered voxels.

Once the mask training is finished, values above 0.95 are set to 1. This ensures that the CNN is only perturbed by the zones identified by the mask, and not by the small gradients that can be found on all the surface of the mask.

Quality check procedure

As the interpretability method is very sensitive to outliers when applied to a group of images, a quality check procedure was performed before mask optimization. Figure 4.2 displays images that passed or failed this procedure. This quality check includes two steps:

1. The gray matter maps were sorted in increasing order by their maximal value. Images with a maximal value lower than 0.95 were automatically rejected. 8 sessions were removed during this procedure.

2. One image was removed after training a group mask. During the training of the first group mask this session led to a significant increase of the loss. This image was removed as it suffered from defects (the eyes were segmented as gray matter).

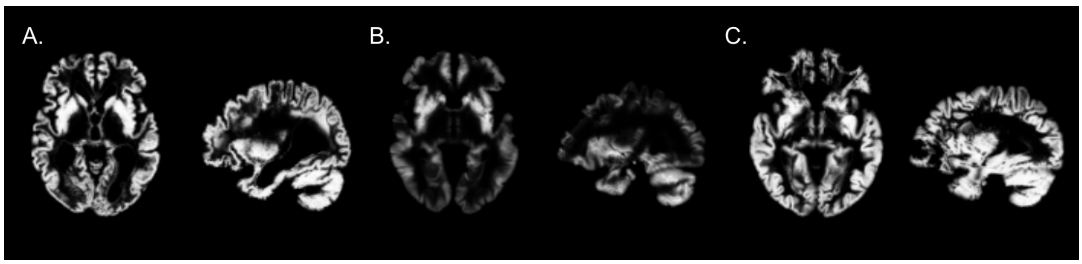


FIGURE 4.2: A. Example of image that passed the quality check. B. Example of image removed during the first step of the quality check. C. Image removed during the second step of the quality check.

Grid search on interpretability hyperparameters

A grid search was performed to choose the set of hyperparameters linked to the computation of the mask: the coefficients for the regularization λ_1, λ_2, β_1, β_2 in equation 4.2. The learning rate was arbitrarily fixed to 0.1. The grid search was only performed on the group level masking for AD label.

Group level masking

To find the regions most related to Alzheimer's disease according to the CNN, we computed a mask based on the subset of images of AD subjects that were in the training set and validation set and all correctly classified by the network. For AIBL, the subset of all AD sessions correctly classified by the network was used. We exploit the fact that a voxel-wise

correspondence exists between the gray matter maps, thanks to the non-linear registration, to iteratively build a group mask: the mask is initialized with all its values set to 1 and is then updated each time with a different image. The subset of well classified AD of the validation set used for the CNN training was again used as a validation set for mask optimization. At the end of each epoch, the masking loss was evaluated on this set to save the best mask according to the validation masking loss. To assess the robustness of the CNN training, the masks obtained for different folds (i.e. different input images and initializations) and different runs of the same folds (i.e. same input images but different initializations) were compared by pairs. The mean value of all pairs gave the similarity between folds or runs of the same fold.

Session level masking

Masks were also produced at the session level based on a single image. To avoid the overfitting risk due to the use of only one image instead of a set of images as in the previous section, the regularization terms λ_1 and λ_2 were multiplied by 100. No validation set was used for these experiments as the goal is precisely to fit the individual pattern of one image instead of finding a general pattern that may correspond to a group of images. Session level experiments include a longitudinal and cross-sectional analysis. In the longitudinal analysis, all the sessions of one subject were compared by pairs and the mean value of these comparisons gave the intra-subject similarity for this subject. The mean intra-subject similarity is then the mean value of the intra-subject similarity of all AD subjects. For the cross-sectional analysis, the mean value of all pairwise comparisons of baseline sessions of all AD subjects gave the inter-subject similarity measure. These analyses are performed to assess the stability of the interpretability method and provide a baseline value by using the inter-subject similarity for the different metrics.

Interpretability method training

The mask was initialized with a matrix of the same size than the input images (121x145x121) full of ones. We applied a similar early stopping strategy than for the classification experiments: the training procedure for group level masking on ADNI does not stop until the relative difference between the validation loss and the lowest validation loss is superior to a tolerance of 0.05 for a patience of N epochs (N=5); otherwise, the training continues to the end of a pre-defined maximum number of epochs (150). For the group level masking on AIBL, the patience was increased to 25 and the maximum number of epochs to 300 as the number of AD subjects is smaller than for ADNI. For the session level masking, the patience was increased to 200 and the maximum number of epochs to 5,000, while the tolerance was decreased to 0.01. The loss corresponds to the argmin argument of Equation (2). For each experiment, the final mask was the one that obtained the lowest validation loss during training. The loss of the mask was evaluated at the end of each epoch.

4.2.4 Metrics of evaluation

The similarity between masks was evaluated in two ways. The output probabilities of the CNN for the true class ($prob_{CNN}$) for an input masked by two masks optimized in two different contexts (e.g. different runs for the group level masking, different sessions of the same subject for the session level masking) are used to establish a comparison based on the CNN perception of the input. A mean output probability close to 1 means that the first model is not perturbed by the mask optimized for the second model, meaning that the two models are dissimilar. A ROI-based similarity was also computed to assess the similarity of two masks according to the 120 regions-of-interest (ROIs) of the AAL2 atlas (Rolls et al., 2015). For each ROI, 1 minus the sum of the values in the ROI is computed, resulting in a ROI-vector of size 120 for each mask. Each value in the ROI-vector represents the density of the mask in the associated ROI. The ROI-based similarity between two masks is then the cosine similarity of two ROI-vectors. A value close to 1 means that the densities of the masks are the same between the ROIs, a value close to 0 means that the locations of the masks have no intersection.

4.3 Results

Once the architecture was chosen and the CNN was trained on all folds, the classification performance was evaluated on the independent test set to ensure the absence of overfitting. The validation balanced accuracies on the five folds were 0.95, 0.82, 0.96, 0.85 and 0.87, giving an average of 0.89. The test balanced accuracies on the five folds were 0.89, 0.87, 0.90, 0.86 and 0.87, giving an average of 0.88. Moreover, the balanced accuracies obtained on the independent test set AIBL were 0.85, 0.92, 0.91, 0.92 and 0.92, giving an average value of 0.90. We could thus conclude that the network was not overfitting and we could use it for the interpretability task.

4.3.1 Grid search on interpretability hyperparameters

First the hyperparameters β_1 and β_2 were chosen with fixed $\lambda_1 = 0.0001$ and $\lambda_2 = 0.001$. The choice of the values $\beta_1 = 0.1$ and $\beta_2 = 1$ was made based on visual inspection. We observe on Figure 4.3 that when β_1 decreases, the minimal value of the mask decreases and this prevents from producing a mask with a large set of values close but different from 1. When β_2 increases, the value of the second term becomes negligible before the first term of equation 4.2. This leads to a very scattered mask as it is dominated by the first term of the regularization.

The hyperparameters λ_1 and λ_2 were then chosen with fixed $\beta_1 = 0.1$ and $\beta_2 = 1$. The choice of the values $\lambda_1 = 0.0001$ and $\lambda_2 = 0.01$ was made based on visual inspection and the stability of the loss during mask training. We observe on Figure 4.4 that when λ_1 increases, the surface covered by the mask decreases until it only becomes scattered points. When λ_2 increases, the surface covered by the mask increases.

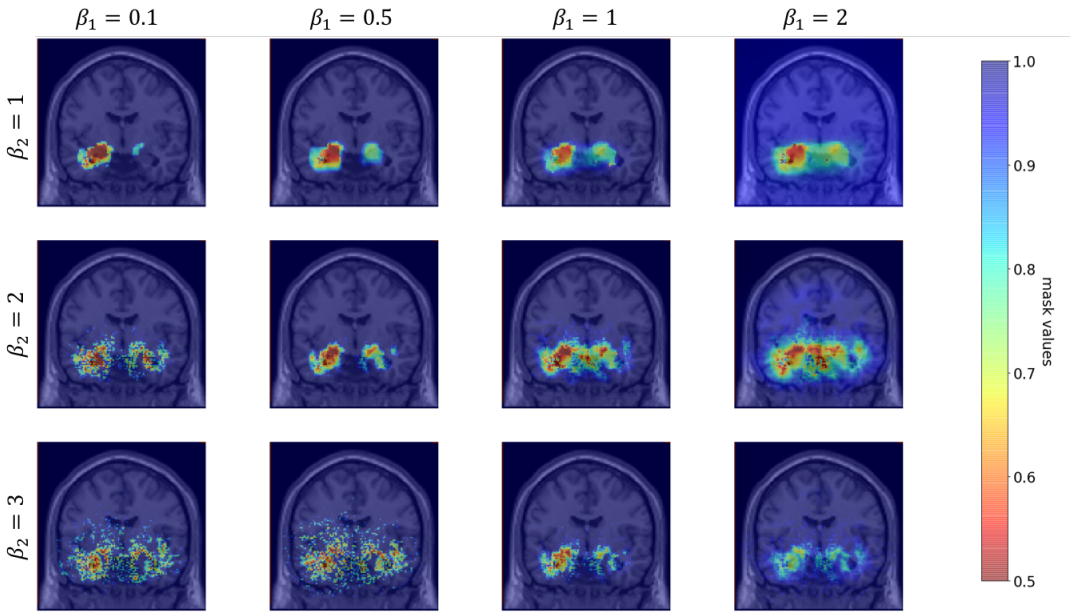


FIGURE 4.3: Comparison of masks obtained for different values of the interpretability hyperparameters β_1 and β_2.

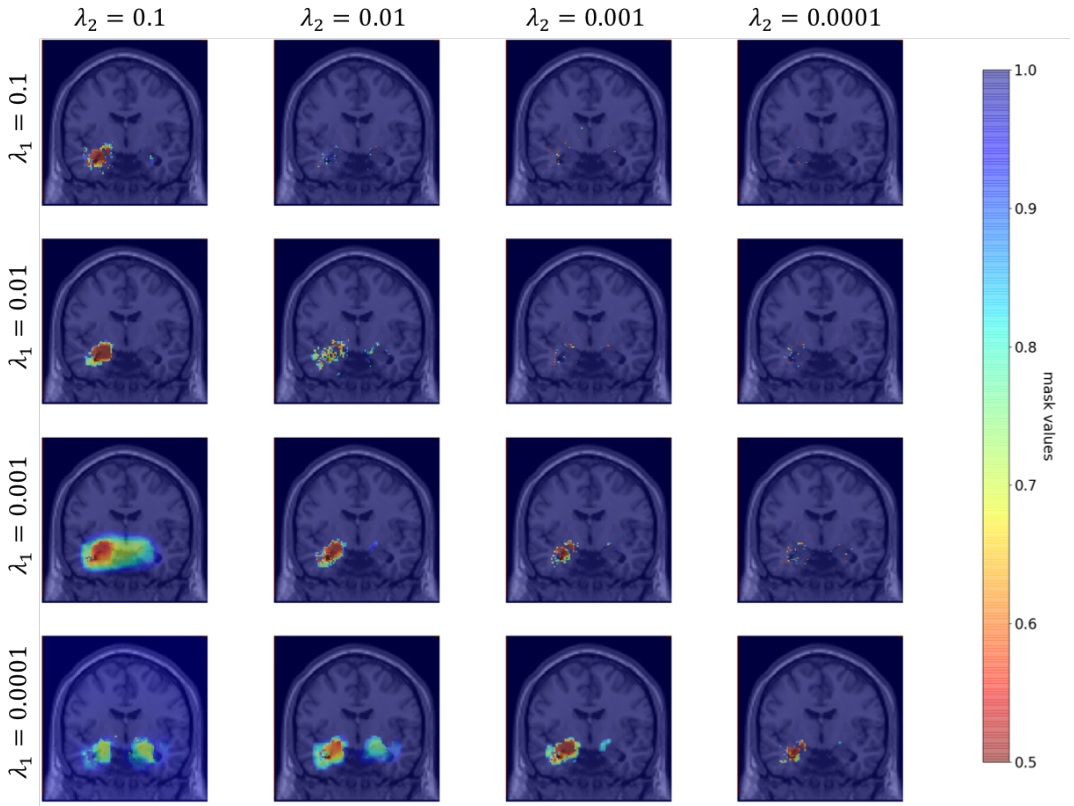


FIGURE 4.4: Comparison of masks obtained for different values of the interpretability hyperparameters λ_1 and λ_2. Note that for $\lambda_1 = 0.0001$ and $\lambda_2 = 0.1$, the learning rate was fixed to 0.01 as the mask optimization did not converge with a learning rate of 0.1.

4.3.2 Robustness of the interpretability method

Different experiments were conducted to assess whether the method was robust enough to help interpret the results of the CNN. Indeed Adebayo et al., 2018 highlighted that some interpretability methods developed to interpret the results of neural networks (for example guided back-propagation and guided grad-CAM) did not depend on model parameters, as they gave the same result with a pretrained network or a randomized one. Hence we need to check if the interpretability method gives coherent results based on the CNN training.

Group level masking

This experiment aims to assess the coherence of the proposed interpretability approach with the a priori knowledge of the disease. One mask was optimized for each of the five models trained on the five folds of the cross-validation. Though these masks do not always overlap, they focus on a set of ROIs known to be particularly affected during AD progression. To confirm this visual observation, the list of the 5 ROIs in which the mask has the lowest values was extracted for each fold. All masks include in this list at least one hippocampus and parahippocampal gyrus. Moreover, the fusiform gyri (4 masks out of 5) and the amygdalae (3 masks out of 5) are frequently highlighted by the masks. Other regions such as the putamen, the pallidum, the inferior temporal gyrus and the thalamus appear only once in these lists.

Moreover, to assess the robustness of the method towards data used for mask optimization we compared the masks obtained by applying the interpretability method on ADNI or AIBL data using the five networks trained on the five folds of the cross-validation on ADNI training/validation set. The corresponding masks are displayed on Figure 4.5. The ROI-based similarities between the pairs of masks were 0.92, 0.99, 0.93, 0.89 and 0.97. These are comparable to the intra-subject ROI-based similarity (0.94). The prob_{CNN} dissimilarities were very small as all the dissimilarities were smaller than 10^{-3}. This indicates that for a given pretrained network, a mask optimized for images of ADNI (resp. AIBL) correctly occludes the images of AIBL (resp. ADNI). However, the comparison of the masks in this way may not be completely fair. Even though the number of epochs and the patience of the early stopping procedure were increased for AIBL masking, the masks on ADNI and AIBL did not benefit from the same number of iterations. This factor leads to masks that comprise a different number of points, as the effect of the regularization terms is correlated to the number of iterations. This means that though the masks highlight the same locations in the brain, the difference in regularization makes the masks more dissimilar that they would be if we could find an equivalence for the hyperparameters that control the number of epochs (patience, tolerance and maximum number of epochs). Hence the dissimilarity here may not be due to the difference of data sets, but to the different number of iterations done during the mask optimization.

Finally, as the method is not deterministic we computed ten times the mask on the first fold of the cross-validation to ensure that the mask optimization is robust to rerun. We

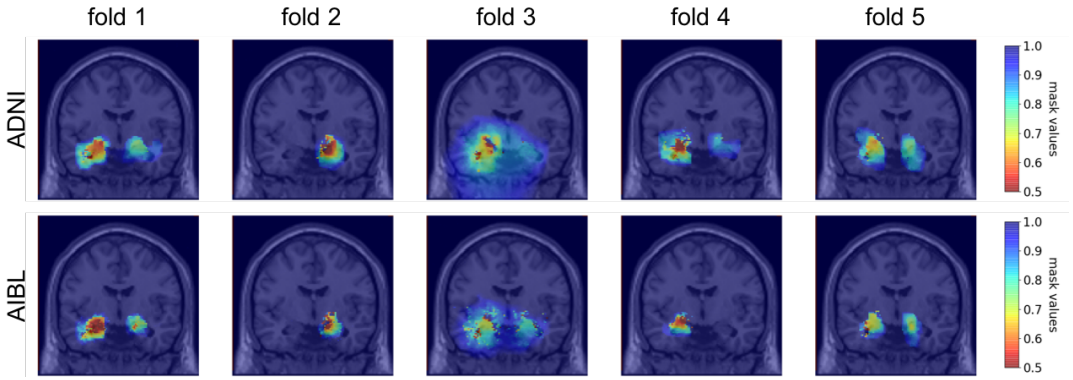


FIGURE 4.5: Coronal view of the group masks trained on ADNI (first line) and AIBL (second line). Each column corresponds to a model trained on one fold of the cross-validation on training/validation ADNI set.

obtained high ROI-based similarities for all pairs of runs (>= 0.97) with a mean similarity of 0.99.

Session level masking

The inter-subject similarity and dissimilarity were evaluated to 0.80 and 0.58 for the ROI-based and the prob_{CNN} metrics respectively. The intra-subject similarity and dissimilarity were evaluated to 0.94 and 0.11 for the ROI-based and the prob_{CNN} metrics respectively. The higher intra-subject similarity compared to the inter-subject similarity ensures that the interpretability metric is robust as the same pattern is generated for different sessions of the same subject.

Similarity across hyperparameters

With the ROI-based similarity, we can assess whether the masks produced by varying one hyperparameter value are similar. To observe this similarity, we reused the same masks as those produced for the grid search (see Figure 4.3 and 4.4).

First, the similarities using different β_1 and β_2 values were computed with fixed $\lambda_1 = 0.0001$ and $\lambda_2 = 0.001$ and a learning rate of 0.1. The similarities between masks produced with β_1 values in 0.1, 0.5, 1, 2 and fixed $\beta_2 = 1$ are given in Table 4.2a. As expected when looking at the masks obtained in Figure 4.3, the masks are highly similar except for the value $\beta_1 = 2$ for which the first regulation term became negligible in front of the second regulation term in equation 4.2. It resulted in a very smooth mask which is dense in all regions of the brain as 97.5% of values are below 0.95. This explains why the ROI-based similarity is so low between this mask and the others, though the regions identified seem similar at visual inspection. Other masks have a high similarity (> 0.95 in all cases). The similarities between masks produced with β_2 values in 1, 2, 3 and fixed $\beta_1 = 0.1$ are given in Table 4.2b. There is more variability for this hyperparameter, though the similarity between two consecutive values is still high (> 0.90).

	$\beta_1 = 0.1$	$\beta_1 = 0.5$	$\beta_1 = 1$	$\beta_1 = 2$
$\beta_1 = 0.1$		0.98	0.97	0.32
$\beta_1 = 0.5$	0.98		1.00	0.36
$\beta_1 = 1$	0.97	1.00		0.37
$\beta_1 = 2$	0.32	0.36	0.37	

(A) Similarity across different β_1 with fixed $\beta_2 = 1$.

	$\beta_2 = 1$	$\beta_2 = 2$	$\beta_2 = 3$
$\beta_2 = 1$		0.90	0.70
$\beta_2 = 2$	0.90		0.91
$\beta_2 = 3$	0.70	0.91	

(B) Similarity across different β_2 with fixed $\beta_1 = 0.1$.

TABLE 4.2: Similarity across different β_1 and β_2 values.

The similarities using different λ_1 and λ_2 values were then computed with fixed $\beta_1 = 0.1$ $\beta_2 = 1$ and a learning rate of 0.1. For both hyperparameters the similarity is high between two consecutive values (>0.90), as can be seen in Table 4.3.

	$\lambda_1 = 0.1$	$\lambda_1 = 0.01$	$\lambda_1 = 0.001$	$\lambda_1 = 0.0001$
$\lambda_1 = 0.1$		0.93	0.84	0.83
$\lambda_1 = 0.01$	0.93		0.95	0.91
$\lambda_1 = 0.001$	0.84	0.95		0.91
$\lambda_1 = 0.0001$	0.83	0.91	0.91	

(A) Similarity across different λ_1 with fixed $\lambda_2 = 0.01$.

	$\lambda_2 = 0.1$	$\lambda_2 = 0.01$	$\lambda_2 = 0.001$	$\lambda_2 = 0.0001$
$\lambda_2 = 0.1$		0.98	0.85	0.72
$\lambda_2 = 0.01$	0.98		0.92	0.82
$\lambda_2 = 0.001$	0.85	0.92		0.96
$\lambda_2 = 0.0001$	0.72	0.82	0.91	

(B) Similarity across different λ_2 with fixed $\lambda_1 = 0.0001$.

TABLE 4.3: Similarity across different λ_1 and λ_2 values.

These results highlight the stability of the method toward the hyperparameters choice, as two consecutive hyperparameter values led to two masks with a ROI-based similarity superior to the inter-subject similarity (0.80). Moreover, all the masks involved in this section analysis correctly occlude the CNN perception: for all masks, the mean output probability of the AD class on the validation data set is below 10^{-6}.

4.3.3 Robustness of the CNN training

After having assessed the robustness of the interpretability method, we applied it to better understand the factors influencing the training process of the CNN classifier based on several scenarios: for different folds (different initialization, different training/validation split) and different runs (different initialization, same training/validation split). Figure 4.6 displays the masks obtained for the five folds and five runs of the first fold.

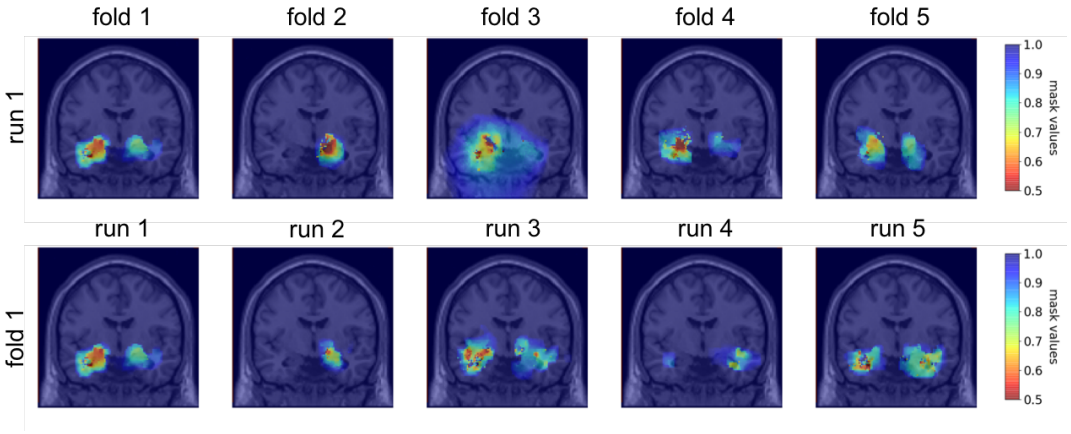


FIGURE 4.6: Coronal view of the group masks obtained for the five folds of the cross-validation on the first run (first line) and of the group masks obtained for five runs of the first fold (second line).

With the prob_{CNN} metric using the validation set, the dissimilarity between folds and 5 runs of the same fold are equivalent with respectively 0.78 and 0.82. The similarities computed with the ROI-based metric are also equivalent with respectively 0.65 and 0.69 between folds and between runs of the same fold. This indicates that the impact of the distribution of the data between training and validation is minimal compared to the initialization of the CNN and the training process. Moreover, we observe that the dissimilarity between folds / runs of the same fold is higher that the intersubject dissimilarity obtained with session level masking. This could mean that the regions on which the CNN relies on to identify the diagnosis mainly depend on the initialization and the training process and that the CNN training is not robust towards the regions identified.

4.4 Discussion

We extended an interpretability method to 3D medical imaging data and used it to better understand the decisions made by a classification network.

We first assessed the robustness of the proposed interpretability approach. We showed that it gave coherent results as the regions identified by the mask are representative of AD (Whitwell et al., 2007). This coherence is also confirmed by the fact that the intra-subject similarity is higher than the inter-subject similarity. Moreover, the high similarity across neighbouring values of the hyperparameters of the masking method indicates that this method is stable towards hyperparameter selection. Finally, we assessed that the method appears robust towards the data used for the construction of group masks by comparing masks computed using ADNI and AIBL data sets.

We then applied the interpretability approach to assess the robustness of the CNN training. We demonstrated that even if the classification performances on the test set are very similar between the different folds, the training of the CNN for our application is not robust as the inter-subject similarity for one training is higher than the similarity between two retrainings or two folds of the network. This problem of robustness in CNN training

may exists for many medical applications in which the number of samples is not sufficient for the network to learn stable meaningful features. This means that it may not be possible to study individual variations using interpretability methods on deep learning applied to imaging data. This problem might be resolved by using more samples and with a better initialization, given for example by an autoencoder pretraining. Moreover, we found that the regions identified by the networks were very small (restricted to the hippocampus, amygdala and part of the temporal lobe) and that most of the image is not exploited by the CNN to find the diagnosis. This confirms the findings in Chapter 3 in which an equal performance was found by learning AD vs CN classification with a CNN on the whole MRI or the hippocampus only. This focus of the CNN on the hippocampi only may be partly due to the data set: ADNI is a research cohort from which patients with multiple phenotypes are excluded, leading to a very homogeneous cohort in which the main symptom of the patients is memory loss. It does not fully represent the diversity of AD phenotype, and they may be biased towards hippocampus atrophy.

The interpretability method we used has several limitations. First, the quality check of the data is crucial otherwise the training of the group level mask is not stable. Second, our method is only meaningful for quantitative data: for T1w MRI it would not have been sensible to increase the value of the voxels as it would have deformed the image in a non-meaningful way. This is an issue as the advantage of deep learning is precisely to be able to adapt to the rawest data possible. Finally, though we explored the effect of four hyperparameters of the interpretability method (β_1, β_2, λ_1, λ_2) we did not conduct an exhaustive study on the impact of the learning rate and the number of epochs performed (correlated to the patience, the tolerance and the maximum number of epochs). As we have seen when comparing masks trained on ADNI and AIBL, these parameters impact the amount of regularization of the masks.

Chapter 5

Identification of unlabeled latent subtypes with attribution maps

5.1 Introduction

In this chapter, we quantified the ability of an interpretability approach to correctly identify the patterns that are the most relevant when differentiating control subjects from patients. Indeed, this is crucial in our context, Alzheimer's disease study, as it is a heterogeneous disease in which some subtypes have already been identified both clinically (see Section Phenotypic heterogeneity) but also with machine learning methods (Habes et al., 2020).

The main drawback of most of the studies presented by Habes et al., 2020 is that patients are grouped along the direction associated with the largest data variability, which may include normal variability. Some studies proposed solutions to reduce this effect. Moradi et al., 2015 improved the accuracy of the prediction of the conversion to AD of patients with mild cognitive impairment (MCI) by removing the effect of the age. The age effect was evaluated using only a sample of cognitively normal (CN) participants, as it should not be mistaken for the progression of the disease over time. This correction took into account only one demographical variable and nevertheless improved the accuracy of the prediction by 4-5 percent points. Another correction of normal variability effects was proposed in (Dong et al., 2016b). The main goal of the method is to find "pathological transformations". These transformations are linear functions that are found by seeking to superimpose the neuroimaging features of the control population over those of the patients (AD and MCI) without altering chosen characteristics. These characteristics (the covariates) are observable variables inducing non-pathological variability. By applying these linear functions to the features of CN individuals, it was possible to determine the features that they would have had if they had undergone a pathological process. In other words, it made them artificially ill without changing the values of the covariates. It was then possible to compare patients with controls with covariates of comparable values, thus taking into account the normal variability. The covariates chosen were sex, age and original cohort. Hence, patients of same age, sex and cohort of recruitment could be compared to patients to study only the pathological process they had underwent. The method was validated in (Dong et al., 2016a), by improving the unsupervised distinction between AD

and Parkinson's disease compared to other classical unsupervised methods (K-means and hierarchical clustering).

Thus, it could be of great value to find new techniques to better characterize the disease taking into account a latent normal variability which may not be entirely related to demographical variables. The chosen approach is a post-hoc interpretability method producing attribution maps (gradient back-propagation) on non-transparent models (convolutional neural networks [CNNs]), and focuses on the ability of CNNs to identify latent subtypes grouped under the same label by clustering feature maps, while attribution maps should accurately highlight patterns of these subtypes. Additionally, we proposed and evaluated simple data augmentation strategies to improve the separability of these subtypes with a limited number of samples. We worked with synthetic data mimicking the patterns of a neurodegenerative disease. In such a controlled setting, we knew which patterns were relevant for the classification, allowing a more reliable comparison of the evaluated techniques. After its validation on synthetic data, this method was then applied to real data. It did not generalize well to this new setting yet.

5.2 Materials

5.2.1 Synthetic data

Synthetic data were used to ensure that the patterns identified by the network correspond to the true patterns specific to the underlying subtypes. It is inspired from the Shepp-Logan phantom (Shepp and Logan, 1974) that was used to develop and test neuroimaging reconstruction algorithms.

We aim to simulate a data set of T1-weighted (T1w) MRI of patients affected by Alzheimer's disease and controls. Three rules were established to generate a synthetic data set:

- the disease is characterized by the atrophy (volume decrease) of specific regions of the brain,
- the disease is heterogeneous, i.e. different atrophy patterns are possible within the patients' group,
- some patients can hardly be distinguished from controls based on their T1w MRI, as the two groups overlap.

Moreover, it is difficult to estimate the proportion of each subtype and undetectable patients within the whole group. This is why we conducted all experiments with two different data sets in which the proportion of the subtypes varies.

The size of the images was fixed to 128×128 pixels. Three subtypes, displayed in Figure 5.1, were sampled by varying the size, center position, orientation and contrast of the ellipses depicting simplified brain regions. A random smoothing was also applied. The size of the regions Top and Bottom allow the identification of three subtypes: subtype 1 has

two large regions, subtype 2 has a large Top region and a small Bottom region, subtype 3 has a small Top region and a large Bottom region.

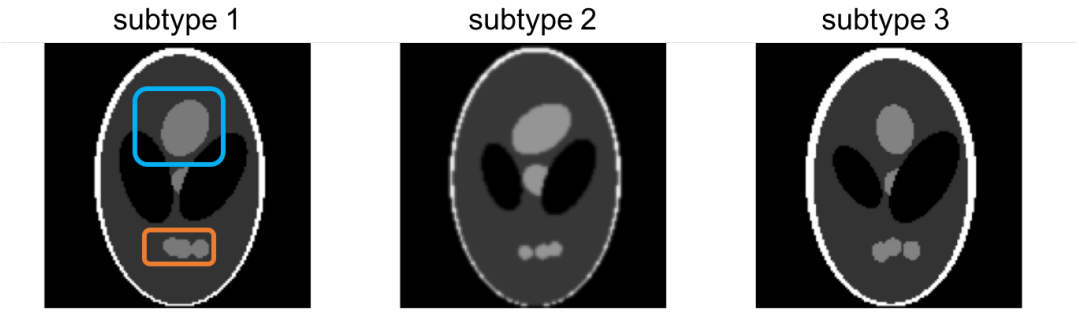


FIGURE 5.1: Display of the three synthetic subtypes. The regions Top and Bottom are highlighted by blue and orange frames, respectively.

Two labels are associated with these subtypes: *Control* and *Atrophied*. The *Control* group comprises only images of subtype 1, whereas the *Atrophied* group comprises images of subtype 1 (errors), subtype 2 (typical) and subtype 3 (atypical).

Two training/validation data sets of 500 samples were sampled with different *Atrophied* groups:

- *homogeneous* comprises 85% of typical subtypes, 10% of atypical subtypes and 5% of errors.
- *heterogeneous* comprises 65% of typical subtypes, 25% of atypical subtypes and 10% of errors.

Finally, *large* has the same properties as *homogeneous*, but is composed of 10,000 samples.

5.2.2 Neuroimaging data

After the validation of the method on synthetic data, we used real data from two public data sets: the Alzheimer's Disease Neuroimaging Initiative (ADNI) database and the Frontotemporal lobar Degeneration Neuroimaging Initiative (NIFD) study. We used the T1w MRI available in each of these studies. Three groups were considered:

- CN: sessions of subjects who were cognitively normal (CN) at baseline and stayed stable during the follow-up (ADNI and NIFD);
- AD: sessions of subjects who were diagnosed as Alzheimer's disease at baseline and stayed stable during the follow-up (ADNI).
- bvFTD: sessions of subjects who were diagnosed with the behavioural variant of fronto-temporal dementia at baseline and stayed stable during the follow-up (NIFD).

The populations of ADNI and NIFD are described in Table 5.1.

The same labels as before were designed based on these groups. The *Control* group corresponds to the fusion of the CN groups of ADNI and NIFD, whereas the *Atrophied* group corresponds to the fusion of the AD group of ADNI (typical subtype) and the bvFTD group of NIFD (atypical subtype).

Data set	Label	Subjects	Sessions	Age	Gender	MMSE	CDR
ADNI	CN	409	2,050	73.2 (6.2) [55.1, 89.6]	185 M / 224 F	29.1 (1.1) [24, 30]	0: 408, 0.5: 1
	AD	390	1,186	74.9 (7.8) [55.1, 90.9]	218 M / 172 F	23.1 (2.2) [17, 29]	0.5: 184; 1: 203; 2: 3
NIFD	CN	131	389	63.2 (7.7) [36, 81]	57 M / 74 F	29.3 (0.8) [27, 30]	0: 84, 0.5: 5
	bvFTD	67	173	61.3 (6.8) [45, 74]	47 M / 20 F	24.2 (4.2) [12, 30]	0: 1; 0.5: 20; 1: 27; 2: 18

TABLE 5.1: Summary of ADNI and NIFD participant demographics, mini-mental state examination (MMSE) and global clinical dementia rating (CDR) scores at baseline. Some participants of NIFD did not have any MMSE or CDR score at baseline.
Values are presented as mean (standard deviation) [range]. M: male, F: female

5.3 Methods

The source code used to generate synthetic data, train the models and generate the attribution maps is part of the ClinicaDL framework[1] described in the last chapter of this PhD thesis.

5.3.1 Baseline CNN classification

The following sections describe the evaluation procedure, the hyperparameter selection and implementation details used in CNN training. All networks learn the binary classification task *Control* versus *Atrophied* with the corresponding images as inputs. The subtypes of the images are not given to the network.

Evaluation procedure

Synthetic data Each training/validation set was split into training and validation sets by performing a 5-fold cross-validation stratified by the subtype. Note that the 5-fold data split was performed only once for all the experiments, thus guaranteeing that all the experiments used exactly the same images during cross-validation. Trained models were then tested on a common test set of 1,000 images in which the *Atrophied* group is composed of half typical and half atypical subtypes. Evaluations were performed without data augmentation. For each model, the balanced accuracy (BA) is given to ensure that the model correctly learnt the task.

Real data This procedure could not be performed on real data, as a more costly data augmentation technique had to be trained. 20% of the whole data set was kept for testing, then the rest of the data set was split only once between training (90%) and validation (10%). All the splits were stratified according to the subtypes (AD, bvFTD and CN), age and sex. As CNN training is not deterministic, in this case each experiment is repeated 20 times to have a better estimation of the performance.

[1] https://github.com/aramis-lab/clinicadl

Architecture and other hyperparameters

As synthetic and real data do not have the same dimensions (2D slices vs 3D images), architectures are specific to the nature of the data.

Synthetic data The chosen architecture is composed of five convolutional blocks followed by a dropout layer with probability 0.5 and a fully-connected layer. Each convolutional block is made of two convolutional layers with kernel size of 3 and padding of 1, a batch-normalization layer, a LeakyReLU activation and a max pooling layer with a kernel size and a stride of 2. Default values were used for other hyperparameters. The predicted label of the input image is the index of the output node having the maximum activation.

Real data The chosen architecture is the same as the one applied to 3D images in Chapter 3. The CNN consisted of five convolutional blocks, a dropout layer with probability 0.5 and three fully-connected layers. Each convolutional block was sequentially made of one convolutional layer with kernel size of 3 and padding of 1, one batch normalization layer, ReLU and one max pooling layer with a kernel size and a stride of 2.

CNN training

The weights of the convolutional and fully-connected layers were initialized as described in (He et al., 2015), which corresponds to the default initialization method in PyTorch. Weights were updated based on the cross-entropy loss. The training continues to the end of a pre-defined number of epochs (100 or 1,000). Unless it is stated otherwise, only the networks trained with 1,000 epochs are presented as data augmentation techniques needed a longer training to converge. The final model is the one that obtained the lowest validation loss during training. The loss of the model is evaluated at the end of each epoch. By default there is no data augmentation.

5.3.2 Data augmentation

Basic procedures

Data augmentation was only applied to the training set during the training procedure. Once an image is loaded it may or may not be randomly altered to produce a slightly different version. The possible transformations are the following:

- **CropPad** samples between 0 and 10 pixels to crop the boundaries on one side of each dimension and pads the other side with the same amount of pixels;
- **Erasing** (originally proposed in (Zhong et al., 2020)) erases a randomly chosen rectangle in the image with a probability of 0.5 by replacing all its values by 0. The default Pytorch values were used;
- **Noise** adds Gaussian noise to the original image with a standard deviation sampled between 0 and 0.1;

- **Smoothing** smooths the image with a Gaussian kernel of standard deviation sampled between 0 and 1.

Note that we limited our search to transformations that could easily be applied to 3D volumes.

Advanced procedure

Another type of data augmentation was applied, to real data only. This advanced data augmentation procedure is based on two variational autoencoders (VAE) learning each the distribution of one label (*Control* or *Atrophied*) on the baseline sessions of the training set only. Then each VAE generates a set of 5,000 synthetic images that is added to real data. The particularity of this framework lies in the way the latent space is sampled, to produce images closer to the original distribution. The detailed methodology can be found in (Chadebec et al., 2021).

We thought that this approach could be relevant as it helped to improve the performance of CNNs on the AD vs CN classification task, particularly when the amount of data was low (which is our case for the bvFTD label) and without an extensive search on the hyperparameters of the network.

5.3.3 Subtype identification using attribution maps

Though many interpretability methods exist, we restricted this study to attribution maps produced by gradient back-propagation (Simonyan et al., 2014), as it is the base of many interpretability methods and is conceptually simple. An individual attribution map corresponds to the gradients of an output node with respect to an image. In our case, the output node is the one corresponding to the *Control* group. Then intensities are related to the changes needed to transform this image into a sample of the *Control* group. A group attribution map is the average of the individual attribution maps of the correctly classified images of this group.

The ability of the network coupled with attribution maps to identify the latent subtypes in the *Atrophied* group was measured with three criteria.

Criterion a. The ability of the feature maps of a network to find latent subtypes was evaluated by clustering ten times the feature maps before the dropout layer and evaluating the mean adjusted rand index between the true subtypes and the clusters found on the evaluation set. This index measures the similarity of two clusterings, corrected for chance.

Criterion b. The ability of group attribution maps to separate the typical from the atypical subtypes was evaluated by comparing the sum of the absolute values of the normalized intensities in the regions Top and Bottom. For the typical subtype, the ratio of the intensities Bottom / Top is computed whereas for the atypical subtype it is Top / Bottom.

$$b = \frac{I_{Bottom}}{I_{Top}} * \frac{S_{Top}}{S_{Bottom}} \qquad b^{\dagger} = \frac{I_{Top}}{I_{Bottom}} * \frac{S_{Bottom}}{S_{Top}} \tag{5.1}$$

where I is the intensity of the mask in a region (sum of its values), and S is the surface of the region.

Criterion c. The specificity of group attribution maps was evaluated by computing the ratio between the sum of the absolute values of the normalized intensities in the regions Top and Bottom and the other regions.

$$c = \frac{I_{Top+Bottom}}{I_{Background}} * \frac{S_{Background}}{S_{Top+Bottom}} \tag{5.2}$$

where I is the intensity of the mask in a region (sum of its values), and S is the surface of the region. Here we consider that the background correspond to the whole image except Top and Bottom regions. Both criteria, c and $c^{\dagger}$, have the same expression.

Intensities in a region are normalized, i.e. divided by the surface of the whole region. When a criterion is applied to the atypical map, the symbol † is used. As discussed in Section 5.6, one should not compare b and $b^{\dagger}$ or c and $c^{\dagger}$.

In the following, the values of criterion a can be compared to the mean adjusted rand index of ten K-means fit directly on the full images of the test set: 0.00 (corresponding to random labeling). For a attribution map in which the intensities are randomly distributed, criteria b, $b^{\dagger}$, c and $c^{\dagger}$ would be equal to ~ 1.

5.4 Results on synthetic data

5.4.1 Baseline results

The three evaluation criteria were applied to networks trained on the *homogeneous* and *heterogeneous* data sets. Criterion values are summarized in Table 5.2 and group attribution maps of the first fold of each network series are displayed in Figure 5.2.

Balanced accuracy is very high ($\geqslant 0.98$) for all the networks. Criterion a is maximal for all the networks, the difference on *homogeneous* being caused by one fold lower than the others. The separability and specificity of attribution maps are equivalent for both data sets. On both data sets the separability of the atypical attribution map ($b^{\dagger}$) is equivalent to a random map, even though these maps are specific ($c^{\dagger} > 1$).

5.4.2 Ideal case: large data set

We studied the impact of the number of samples within a data set by comparing the results obtained with the *large* and *homogeneous* data sets (Table 5.2 and Figure 5.2).

For the *large* data set, networks were only trained during 100 epochs because it was more computationally costly and convergence was reached sooner.

For the *large* data set, the separability (b & $b^{\dagger}$) and the specificity (c & $c^{\dagger}$) of the attribution maps are much higher than for the *homogeneous* data set. Moreover, the

TABLE 5.2: Baseline performance obtained for each data set.

	Latent subtype identification criteria					
data set	a	b	c	$b^{\dagger}$	$c^{\dagger}$	BA
homogeneous	0.95	12.25	5.32	1.59	4.15	0.98
heterogeneous	1.00	8.55	5.07	0.90	4.60	0.98
large	0.99	102.51	12.26	12.88	13.67	1.00

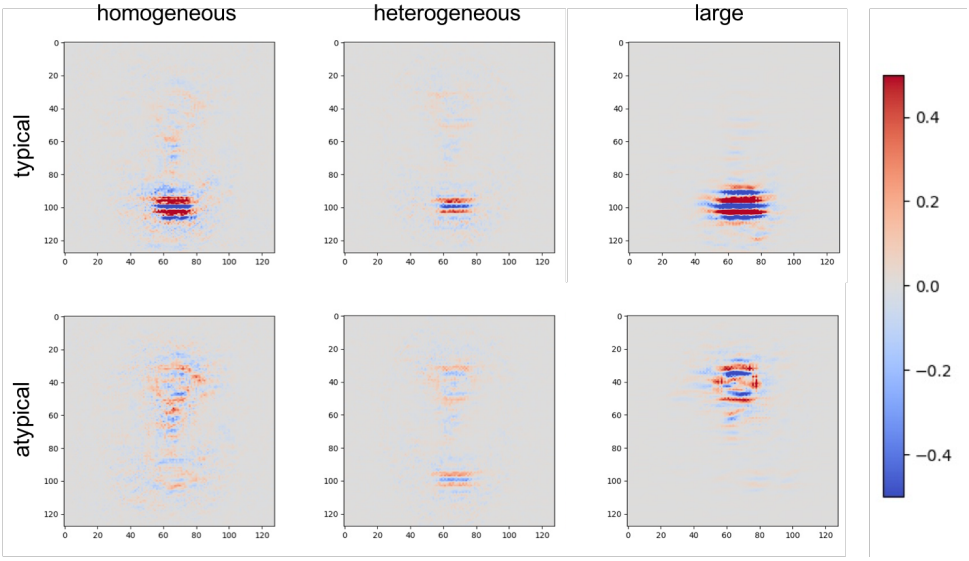


FIGURE 5.2: Group attribution maps obtained for the first fold of the 5-fold cross-validation with each data set (*homogeneous*, *heterogeneous* and *large*) for both the typical and atypical subtypes.

areas highlighted by the attribution maps appear more separated at visual inspection. This indicates that CNNs are able to refine their judgement with a larger amount of data. However, data sets in medical imaging tend to be small so we need to find solutions to improve the separability and specificity of our attribution maps on small data sets.

5.4.3 Benchmark of data augmentation strategies

We proposed to artificially increase the data set size by performing data augmentation, and evaluated whether the techniques described in section 5.3.2 improve the separability and specificity of attribution maps.

Notable improvement was observed for the *heterogeneous* data set only. Only **CropPad** led to a statistical improvement, and on atypical maps only. The two best-performing techniques were selected to be applied in the following order: **CropPad** and **Noise**. This combination led to a statistically significant improvement compared with the baseline values on b, c and $c^{\dagger}$. Criterion values are summarized in Table 5.3 and group attribution maps obtained with the *homogeneous* data set are displayed in Figure 5.3.

TABLE 5.3: Benchmark of the data augmentation techniques applied to the *homogeneous* and *heterogeneous* datasets. * denotes a statistical difference with the baseline values assessed using a T-test (p<0.05) with Bonferroni correction.

	Latent subtype identification criteria					
Technique	a	b	c	$b^{\dagger}$	$c^{\dagger}$	BA
CropPad	1.00	13.15	4.70	2.00	5.09	0.99
Erasing	1.00	19.55	5.57	1.79	3.96	0.99
Noise	1.00	18.37	9.08	2.32	5.48	0.99
Smoothing	1.00	10.36	2.94	1.13	4.26	0.99
CropPad + Noise	1.00	19.05	6.41	4.20	7.12	1.00
None	0.95	12.25	5.32	1.59	4.15	0.98

(A) *Homogeneous* dataset

	Latent subtype identification criteria					
Technique	a	b	c	$b^{\dagger}$	$c^{\dagger}$	BA
CropPad	1.00	20.60	5.75	2.56*	7.29*	1.00
Erasing	1.00	13.72	6.28	1.99	5.22	0.99
Noise	1.00	17.28	8.11	2.34	5.39	0.99
Smoothing	1.00	12.11	5.12	1.34	3.94	0.99
CropPad + Noise	1.00	41.79*	7.62*	3.66	8.16*	1.00
None	1.00	8.55	5.07	0.90	4.60	0.99

(B) *Heterogeneous* dataset

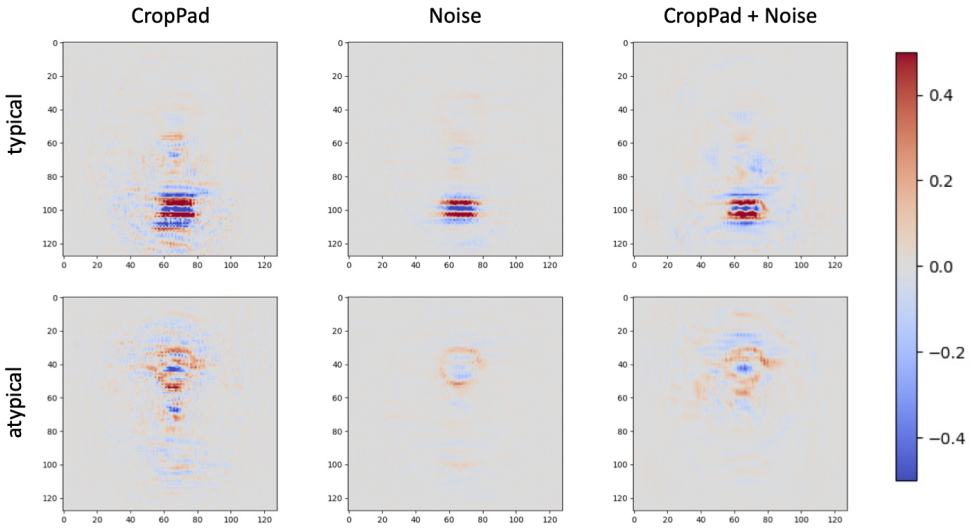


FIGURE 5.3: Group attribution maps obtained for the first fold of the 5-fold cross-validation with the *homogeneous* data set and different data augmentation strategies (**CropPad**, **Noise** and their combination), for both the typical and atypical subtypes.

5.5 Results on real data

5.5.1 Specific patterns of the subtypes

In this setup, the subtypes correspond to two different diseases. Though we have a prior idea on the atrophy pattern which should be found in typical (AD) and atypical (bvFTD) subtypes, we first trained networks learning to differentiate CN from AD or CN from bvFTD. Then, group attribution maps of the patients group were computed (see Figure 5.4). These patterns are our references to establish the "ground truth" which should be found with the interpretability method after wise.

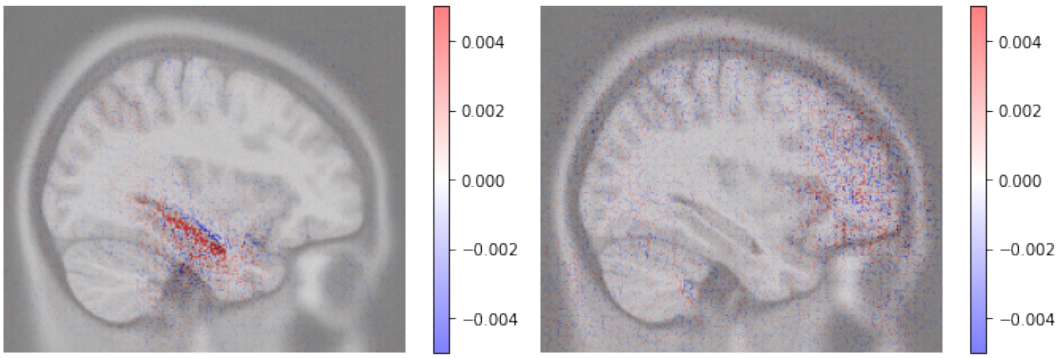


FIGURE 5.4: Group attribution maps of the two CNNs trained on AD vs CN (left) and bvFTD vs CN (right) tasks.

Both CNN obtained a good balanced accuracy on the test set for AD vs CN (0.93) and for bvFTD vs CN (0.85). We observe that the bvFTD pattern is characterized by an atrophy in the frontal lobe. The AD pattern, as already observed in Chapter 4, is characterized by the atrophy of the hippocampi. These findings match with the clinical description of the diseases.

We already observe that the bvFTD group is less well characterized than the AD one: the attribution map is more scattered and the balanced accuracy obtained on bvFTD vs CN is lower than the one obtained on AD vs CN.

5.5.2 Basic data augmentation

The data augmentation procedures performing best on synthetic data, Noise and CropPad, were then applied to real data. The mean test balanced accuracy was 87.7% (*Control* vs *Atrophied*). Typical and atypical subtypes cannot be distinguished as the mean value of criterion a is 0.001. Moreover, we observe that the group attribution maps of both subtypes (displayed in Figure 5.5) cannot be distinguished: they both mostly correspond to the pattern of the typical subtype only.

5.5.3 Advanced data augmentation procedure

The advanced data augmentation procedure did not improve the clustering criterion. The mean test balanced accuracy was 84.3% (*Control* vs *Atrophied*). The mean value of criterion

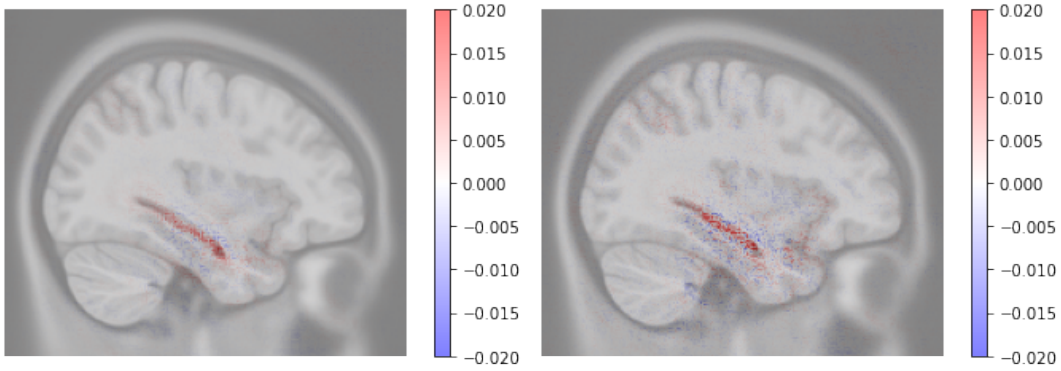


FIGURE 5.5: Group attribution maps of typical (left, AD) and atypical (right, bvFTD) subtypes of the first CNN trained on *Control* vs *Atrophied* with the basic data augmentation procedure.

a is 0.02, then it is not possible to distinguish typical and atypical subtypes. Again, we observed that the group attribution maps of both subtypes (displayed on Figure 5.6) are highly similar, though they seem a bit different than with basic data augmentation.

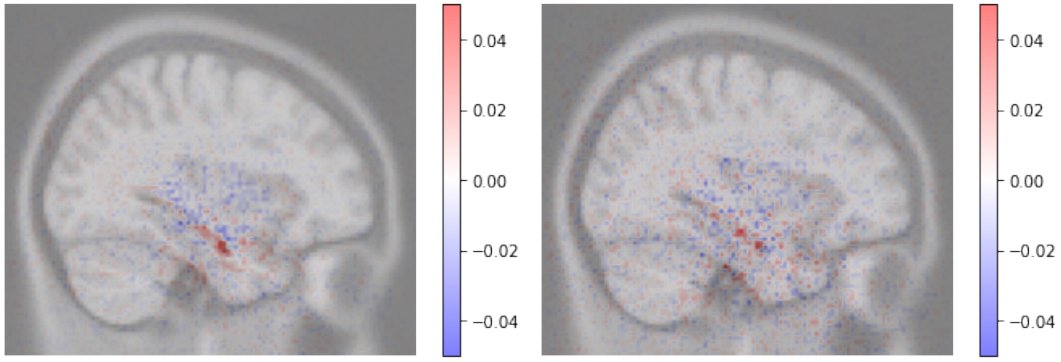


FIGURE 5.6: Group attribution maps of typical (left, AD) and atypical (right, bvFTD) subtypes of the first CNN trained on *Control* vs *Atrophied* with the advanced data augmentation procedure.

5.6 Discussion

We studied the ability of CNNs and attribution maps to correctly identify subtypes composing a heterogeneous label.

On synthetic data, we observed that we could retrieve the subtypes thanks to K-means applied to internal feature maps. However the attribution maps better represent the typical subtype than the atypical one for data sets with a small number of samples, whereas both subtypes are well represented when the data set size is larger. This conclusion applied even though the balanced accuracy of the models trained on a small amount of data is very high. As we cannot easily increase the size of neuroimaging cohorts, we studied the impact of different data augmentation techniques to better identify the latent patterns in our data set with attribution maps. We managed to improve the separability and specificity of group attribution maps on a small data set. We showed that this improvement depends on the

subtype distribution. For the *homogeneous* data set, the increase in performance was generally smaller than for the *heterogeneous* data set.The reduced number of techniques reaching a statistically significant improvement compared with the baseline could also be due to the limited number of models run per technique (five, one for each fold). This analysis could be improved by running more models for each experiment.

Unfortunately, this framework did not generalize to real data, even with the use of a more advanced data augmentation procedure. Our main hypothesis is that as the task is much more difficult, the network tend to gather both subtypes to differentiate them from the normal distribution. Then, when we perform the clustering we cannot find anymore the difference between the two subtypes. One possible solution to avoid this problem could be to learn only the normal distribution (for example with an autoencoder) and to detect the anomalies characterizing the subtypes to differentiate them.

This study is limited to the use of attribution maps, but other methods exist. Comparing them using the proposed criteria would also allow assessing the reliability of these criteria as proposed in (Tomsett et al., 2020), by comparing the inter-rater reliability on individual assessments of each interpretability method.

Finally, solutions must still be found to better compare interpretability of areas with different shapes. Indeed, we found by running simulations with the same shape for the Top and Bottom regions that in this case the b and $b^{\dagger}$ criteria were more similar (see Appendix D). The fact that b is larger than $b^{\dagger}$ may also be correlated to the shapes of the regions and not only to the occurrence of typical and atypical subtypes.

Chapter 6

ClinicaDL: An open-source deep learning software for reproducible neuroimaging processing

---

This chapter has been submitted to Computer Programs in Biomedicine.

- **Title:** ClinicaDL: an open-source deep learning software for reproducible neuroimaging processing
- **Authors:** Elina Thibeau-Sutre[†], Mauricio Díaz[†], Ravi Hassanaly, Alexandre Routier, Didier Dormont, Olivier Colliot, Ninon Burgos.

[†] denotes shared first authorship

6.1 Introduction

In recent years, deep learning has become one of the most used data analysis technique. This statement also applies to computer-aided diagnosis systems in which convolutional neural networks (CNNs) are widely used to provide a diagnosis or predict the future state of patients from neuroimaging data. Unfortunately, this recent massive use of deep learning has also been associated with methodological flaws in many studies (Bussola et al., 2021; Panwar et al., 2020; Samala et al., 2020; Wen et al., 2020; Yagis et al., 2021). Such studies overestimate the performance of their network in performing classification because their test set (when it exists) is contaminated by data leakage. This is a major issue in the field that may lead to troublesome consequences:

- Real-life applications of these algorithms may lead to dramatic failures.
- Authors producing honest results with a sound method may not succeed in publishing their results because the biased state-of-the-art performance is too high.
- Other methods than deep learning are not explored anymore because deep learning performance seems impossible to exceed, leading to a loss of diversity in our research field that may be problematic when the limit of this technique will be reached.

Moreover, Hutson, 2018 points out that the whole deep learning community faces a reproducibility crisis that discredits the results obtained with this method. Hence there is an urgent need in publishing open-source software, data sets and scripts that allow reproducing the methodologies described in deep learning studies. Finally, another main difficulty encountered by deep learning users who are not neuroimaging specialists is the access to properly formatted and preprocessed data sets. In our field, a standard was developed to better organize and share neuroimaging data sets: the Brain Imaging Data Structure (BIDS) (Gorgolewski et al., 2016), then projects such as Clinica (Routier et al., 2021) or BIDS Apps (Gorgolewski et al., 2017) help preprocess these BIDS data sets. But unfortunately, many data sets are still not distributed in this format, and deep learning users are not always familiar with these preprocessing tools.

Several open-source repositories have been made available in the last years to ease deep learning application to medical images. Lakhani et al., 2018 proposed a tutorial-like paper explaining how to use Keras to train a classifier on two specific data sets hosted on a GitHub repository[1]. Others preferred to implement dedicated open-source libraries to propose an easier use of deep learning for medical image analysis. The most important one is Monai[2], a large library merging three other libraries that are not maintained anymore: NiftyNet[3] (Gibson et al., 2018), DeepNeuro[4] and DLTK[5] (Pawlowski et al., 2017). This library goes beyond the context of neuroimaging and allows processing medical images from different body parts. It is meant to work on MedMNIST[6], a series of ten data sets of preprocessed medical images of different modalities (cancer histology, chest X-ray, dermatoscopy, optical coherence tomography, fundus photography, breast ultrasound and abdominal computed tomography) and the ten data sets of the medical segmentation decathlon challenge (Antonelli et al., 2021; Simpson et al., 2019)[7] that aims at segmenting organs or tumours. It is also possible to use Monai on other data sets but this requires additional work. Monai provides low-level functions and classes that must be combined in a Python script to learn a classification or a segmentation task, or to train a generative adversarial network (GAN). Attribution methods are also available: class activation mapping (CAM), gradient-weighted CAM and occlusion sensitivity, which allow interpreting the trained network results. A large diversity of transforms, loss functions, metrics and optimizers are provided. Finally, with the support of Nvidia, Monai provides multi-GPU parallel processing. The other main Python library for deep learning in medical research is TorchIO (Pérez-García et al., 2021). This library does not implement deep neural network training, but implements a large variety of transforms for 3D image preprocessing and/or augmentation to prepare data for deep learning use. As with Monai, two public data sets are integrated with the library for ease of use: IXI[8] and EPISURG (Pérez-García et al., 2020). An interface to manage custom

[1] https://github.com/ImagingInformatics/machine-learning
[2] https://monai.io
[3] https://github.com/NifTK/NiftyNet
[4] https://github.com/QTIM-Lab/DeepNeuro
[5] https://github.com/DLTK/DLTK
[6] https://medmnist.github.io
[7] http://medicaldecathlon.com
[8] https://brain-development.org/ixi-dataset

data sets is also provided. Finally, Nobrainer[9], focuses on learning brain segmentations and is documented by Jupyter notebook tutorials. Other initiatives were launched but seemingly abandoned, such as for example NiftyTorch[10], MildInt[11] (Lee et al., 2019b) or pymia[12] (Jungo et al., 2021).

Though there was an effort from previous works to integrate several data sets that can be easily downloaded through their API, other cohorts might be quite difficult to process. Moreover, they do not easily handle longitudinal data sets, in which several images correspond to the same participants and then should not be distributed between the training and test sets. Finally, the reproducibility of the experiments conducted with these frameworks still heavily relies on the user. Indeed, the hyperparameter values are defined in the scripts, hence their previous values may be lost as new experiments are launched.

To help deep learning users to (1) format and preprocess neuroimaging data sets, (2) prevent data leakage from biasing their results and (3) reproduce their experiments, we implemented ClinicaDL: a command line software written in Python meant to train deep learning networks to reconstruct input images, or to predict a categorical (classification) or a continuous (regression) variable based on neuroimaging data. As the name suggests, it is meant to work on the outputs of Clinica (Routier et al., 2021), an open-source software platform for reproducible clinical neuroimaging studies. The core of Clinica is a set of automatic pipelines for multimodal neuroimaging data preprocessing with standard tools of the community (such as SPM[13], FreeSurfer[14] or ANTS[15]). ClinicaDL takes as inputs these preprocessed images and convert them into tensors to train deep neural networks. Thus, the combination of these two tools allows performing an end-to-end neuroimaging analysis, from the download of raw data sets to the interpretation of trained networks, including neuroimaging preprocessing, quality check, label definition, architecture search, and network training and evaluation.

6.2 Avoiding common pitfalls in deep learning studies with ClinicaDL

6.2.1 Formatting and preprocessing of neuroimaging data

One difficulty faced by data scientists is the manipulation of raw neuroimaging data sets as their organization can be quite difficult to understand. Moreover, raw images coming from different scanners may need some preprocessing to be handled by deep neural networks. These preprocessing steps are easier to perform and manage when data are organized in a standard manner. To allow any researcher to completely or partly reproduce the steps performed in a study with ClinicaDL, we decided to work with data set structures, described

[9] https://github.com/neuronets/nobrainer
[10] https://github.com/NiftyTorch
[11] https://github.com/goeastagent/MildInt
[12] https://github.com/rundherum/pymia/tree/master
[13] https://www.fil.ion.ucl.ac.uk/spm
[14] https://surfer.nmr.mgh.harvard.edu/fswiki/FreeSurferWiki
[15] http://stnava.github.io/ANTs

in the following sections, whose specifications are exhaustive. Moreover, our framework is adapted to the use of 3D images, as it allows the user to choose their own way to cut the image in smaller pieces (patches or slices) to ease network training (see ClinicaDL modes in section 6.2.1).

BIDS format

Clinica and ClinicaDL follow the Brain Imaging Data Structure, described in (Gorgolewski et al., 2016), to organize their datasets. The BIDS standard provides a list of specifications[16], which specify how files in a BIDS data set should be organized, named and formatted. It is widely adopted by the neuroimaging community, and more and more databases try to distribute their data in BIDS format or approaching (see the OpenNeuro[17] platform for a list of BIDS formatted data sets). However, some databases still use a custom format that can be quite difficult to exploit. This is the case for example of the Alzheimer's Disease Neuroimaging Initiative (ADNI)[18], the Australian Imaging, Biomarker & Lifestyle Flagship Study of Ageing (AIBL)[19], or the Open Access Series of Imaging Studies (OASIS)[20], three databases used for the characterization of Alzheimer's disease dementia and its prodromal stage, or of the frontotemporal lobar degeneration neuroimaging initiative (NIFD), a database including patients with frontotemporal dementia (available from the same platform as ADNI and AIBL[21]). Clinica includes BIDS converters to format these four raw data sets to BIDS format to ease their use.

CAPS format

When a database is BIDS-formatted, it is possible to preprocess its images with Clinica pipelines. These pipelines can for example perform intensity and spatial normalization of brain images to allow the extraction and analysis of comparable features from images of different participants. Two pipelines have been developed specifically for deep learning use, though the outputs of the other pipelines can also be used with ClinicaDL. These pipelines mainly perform a linear registration to a standard space for two different modalities: T1-weighted (T1w) magnetic resonance imaging (MRI) and positron emission tomography (PET) images. They output another folder whose structure is derived from BIDS: the ClinicA Processed Structure (CAPS)[22].

ClinicaDL modes

The preprocessing pipelines of Clinica operate at the image level, but many deep learning systems work with one or several parts of the original 3D image. Four possible uses of the

[16] https://bids-specification.readthedocs.io
[17] https://openneuro.org/
[18] http://adni.loni.usc.edu
[19] https://aibl.csiro.au
[20] https://www.oasis-brains.org
[21] https://ida.loni.usc.edu
[22] https://aramislab.paris.inria.fr/clinica/docs/public/latest/CAPS/Introduction/

image (modes) are currently implemented in ClinicaDL. These modes correspond to the practices commonly found in the literature (cf Chapter 2):

1. `image` uses the whole 3D image,
2. `patch` extracts 3D cubic patches with predefined size and stride to cover the whole image,
3. `roi` extracts specific 3D regions defined by binary masks generated by the user,
4. `slice` extracts 2D slices according to a neuroanatomical plane (sagittal, coronal or axial).

ClinicaDL computes the image-level performance by assembling the mode-level performance when it is different from `image`. Advanced users may want to implement their own modes. The documentation describes the steps to follow to implement and use custom modes.

MAPS format

Model Analysis and Processing Structure (MAPS) names the output structure of the ClinicaDL train function. All the functions of ClinicaDL are meant to work on this structure to easily retrieve the parameters of the command line, the weights of the best models, the checkpoints, or the predictions made on the training and validation sets to compute the results at the image level on independent test sets. At the root of the hierarchy, the file `environment.txt` summarizes the environment used for training, and `maps.json` gathers the arguments provided to the command line.

This structure includes a hierarchy of three levels:

1. **Folds** The first level contains one folder per train / validation split. The training procedure of each fold can be launched independently.
2. **Selection metrics** During the training procedure of a particular fold, one network is selected per selection metric given in input. These networks correspond to the network having the best validation performance according to their metric during the training procedure.
3. **Data groups** Finally, the best networks selected are evaluated on data groups. The characteristics of these data groups (TSV file of participant and session IDs with label values, and path to the CAPS directory) are stored at the first level of the hierarchy in the `groups` folder. This specification ensures the consistency between the evaluations of different networks trained on different folds and selected on different metrics.

An example of the MAPS obtained when training a classification CNN on images is displayed in Table 6.1. The MAPS also stores training logs. Two different formats are available: they can be opened with Tensorboard[23] and are also available as TSV files.

[23] https://www.tensorflow.org/tensorboard

```
results
├── environment.txt
├── fold-0
│   ├── best-loss
│   │   ├── model.pth.tar
│   │   ├── train
│   │   │   ├── description.log
│   │   │   ├── train_image_level_metrics.tsv
│   │   │   └── train_image_level_prediction.tsv
│   │   └── validation
│   │       ├── description.log
│   │       ├── validation_image_level_metrics.tsv
│   │       └── validation_image_level_prediction.tsv
│   └── training_logs
│       ├── tensorboard
│       └── training.tsv
├── groups
│   ├── train
│   ├── train+validation.tsv
│   └── validation
└── maps.json
```

TABLE 6.1: Example of the Model Analysis and Processing Structure (MAPS) obtained when training a classification network on whole images. Folders are in bold.
Only the first fold was trained (folder `fold-0`) and one model was selected based on its validation loss (folder `best-loss`).
The only data groups are `train` and `validation`, which are automatically created during training. The characteristics of these groups are defined in `groups`, whereas the folder in `fold-0/best-loss` contains the results for each input image (file `*_prediction.tsv`) and a set of metrics (file `*_metrics.tsv`) for each data group.
Finally, training logs are available for each fold training in the folder `training_logs`. These logs are available in two different formats, Tensorboard compatible and TSV.
As the training procedure ended without raising an error, the checkpoints were erased (this allows saving memory).

Conclusion

Relying on BIDS allows easing the processing of neuroimaging data as it is a standard format. As already mentioned, many BIDS data sets are hosted on OpenNeuro[24]. For the others, tools have been developed to ease their conversion (a list of these tools is available on the BIDS website[25]).

The other formats we introduced (CAPS and MAPS) are useful as these structures are now stable and their elements can be easily processed and retrieved with tools of Clinica or ClinicaDL. For example ClinicaDL includes two quality check procedures taking as input the CAPS generated by pipelines of Clinica (`t1-linear`[26] and `t1-volume`[27]).

6.2.2 Data leakage handling

As explained by Kaufman et al., 2012, data leakage is "the introduction of information about the target of a data mining [a.k.a. machine learning] problem that should not be legitimately available to mine from". They give two main reasons for data leakage:

- leaking features, occurring for example when input data include features that are highly correlated to the target label due to a selection bias or if the target is a cause of the feature,
- leakage in training examples, occurring when data used for training is not legitimate towards data used for performance evaluation (for example, if there is an intersection between training and test data).

Let's take the example of the inference of the diagnosis from neuroimaging data. In this case, the leaking feature scenario may happen as a selection bias. For example, consider a data set that includes several sites. If each site has a different diagnosis distribution (in the worst case, one site only recruits patients, whereas another one only recruits control participants), the site is a leaking feature for the diagnosis. Unfortunately, as these sites use different scanners, the site information may be retrieved from the neuroimaging data. This selection bias requires expert knowledge of the data set used to be avoided.

We mainly focus in this article on leakage in training examples, which is independent from the data sets used. In chapter 1, we reported that data leakage contaminated nearly half of the studies using a CNN on T1w MRI for the diagnosis of Alzheimer's disease. Other studies using deep learning in the health domain also mention that data leakage pollutes their field of application: Samala et al., 2020 in breast cancer detection from mammograms, Panwar et al., 2020 for Covid-19 diagnosis from chest radiography and Bussola et al., 2021 for image classification in digital pathology. Finally, Yagis et al., 2021 quantified the difference between a biased and a right split between train and test sets on the test accuracy for several tasks using neuroimaging data. The differences they measured ranged from 25% on a large data set to 55% on a small data set.

[24] `https://openneuro.org/public/datasets`
[25] `https://bids.neuroimaging.io/benefits.html#converters`
[26] `https://aramislab.paris.inria.fr/clinica/docs/public/latest/Pipelines/T1_Linear/`
[27] `https://aramislab.paris.inria.fr/clinica/docs/public/latest/Pipelines/T1_Volume/`

We identified four scenarios of data leakage in chapter 1 and we add here a last one (biased ensemble learning) that has been identified afterwards. Then the complete list of data leakage scenarios on training examples is the following:

1. **Absence of an independent test set** occurs when the classifier performance is evaluated on the training or the validation set.

2. **Biased split** occurs when highly correlated data (slices or patches extracted from the same volume, visits from the same patient, etc.) are both in the train and the test sets.

3. **Late split** occurs when another procedure is performed prior to the data split.

4. **Biased transfer learning** occurs when data is shared between the source and the target task and that the train / test split has been done differently. Some authors seem to find that there is no risk of data leakage when using transfer learning if the target and the source tasks are different, however it may happen if they share a subset of participants.

5. **Biased ensemble learning** occurs when parts of the images are selected / weighted thanks to the labels of the test set to deduce the image-level prediction of the test set.

For clarity, these scenarios are illustrated in Figure 6.1.

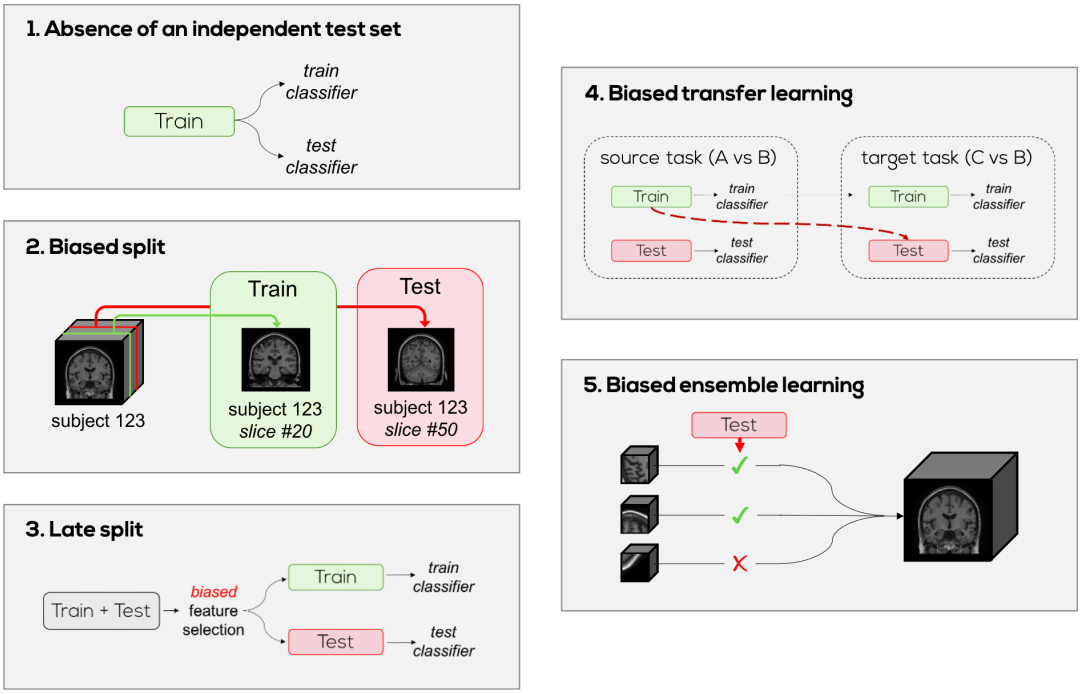


FIGURE 6.1: Illustration of the scenarios that can lead to data leakage.

ClinicaDL prevents the user from these scenarios by implementing the following strategies:

1. Data splits are done at the subject level and cannot be performed on-the-fly but must be done prior to training networks (to avoid a biased split).

2. Data splits are done independently for each label. However, if labels B & C are subsets of a parent label A, transfer learning from a task implying A to a task implying B and/or C may result in a biased transfer learning. Therefore ClinicaDL splits B and C with respect to A split (see Figure 6.2 for more insight).

3. Data augmentation is performed on-the-fly (to avoid late split).

4. In the classification case, the image-level prediction is the weighted sum of parts of the image. These weights are computed from the predictions on the training or the validation sets, but no other set (to avoid biased ensemble learning).

5. At the root of the MAPS, the file `train+validation.tsv` comprises all the participant and session IDs seen during the training procedure. If transfer learning is performed, this list of IDs is updated to include the IDs of participants and sessions seen during the training of the source task. ClinicaDL prevents the user from creating a data group having common IDs with this list (to avoid biased data split and transfer learning).

The absence of an independent test set is still possible, as ClinicaDL does not force the user to give the test set to evaluate the performance in an unbiased way during training. We do not wish to enforce such system as the user could want to do some hyperparameter optimization based on the training and validation performance only before evaluating the unbiased performance on the test set. Then it is the user's responsibility to evaluate the final performance once they have done all the research they wanted on hyperparameters.

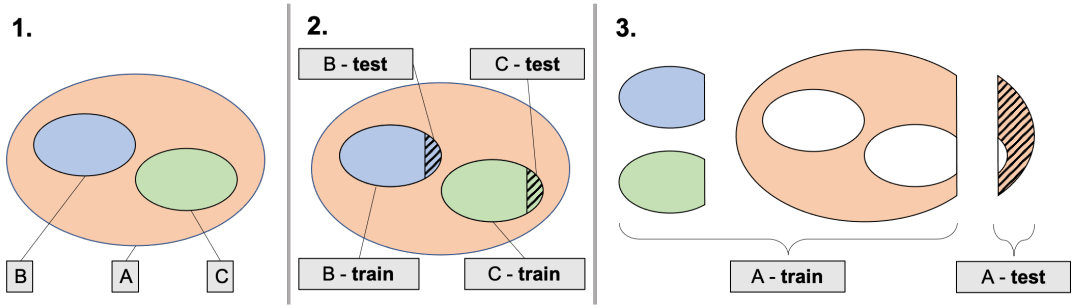


FIGURE 6.2: Sequence of data split when diagnostic labels (B and C) are subgroups of another diagnostic label (A). (1) Participants of each group are identified. (2) B and C subgroups are split between train and test data sets. (3) The rest of the parent group is split between train and test, and the train set of each subgroup is added to the parent train set.

6.2.3 Reproducibility

In the same way as "deep learning", "reproducibility" is an-ill defined concept, often mentioned using similar words (repeatability, reproducibility, replicability) whose

meanings may vary across articles. This is why we chose in this article to work with the definitions of Goodman et al., 2016, in which three different levels of reproducibility are defined:

- **method reproducibility** (sometimes called repeatability) is the ability to repeat the same experiment using the same tools and data to obtain the same results,
- **result reproducibility** (sometimes called replicability) is the corroboration of results by other studies using the same experimental methods,
- **inferential reproducibility** (often not discussed) exists when different scientists deduce the same knowledge claims from a similar study or the re-analysis of the study.

As explained by the authors, inferential reproducibility may be impossible to guarantee as the analysis of results is peculiar to each scientist (and this is also what drives science forward). However, none of these levels are guaranteed if the research is not transparent. What will mostly be discussed here is thus the way to achieve transparency, which consists in closely describing the different steps linking the prior hypothesis to the final claim.

The initial step to achieve transparency is to share usable code. This way Collberg and Proebsting, 2016 tried to locate and build the source code of studies of eight ACM conferences. They first noticed that only half of the source codes could be located. Among these codes, half of them could be built easily (i.e. in less than 30 minutes without the authors' help), others needed more time, the help of the authors, or could even fail (in rare cases). To prevent this pitfall, the source code of ClinicaDL is available on GitHub. Moreover, tests are run at each commit to ensure that the code can be correctly installed and that the main functionalities can be run (see Section 6.3.1 for more information on tests).

However, sharing code is not enough to be fully transparent. For non-deterministic models such as deep learning, method reproducibility can only be achieved by setting a random seed (Beam et al., 2020; Crane, 2018). Crane, 2018 also evaluated the impact of the computational setup, and explained that the software versions of all the system used, the GPU version and even threading should be explicit to allow method reproducibility. All these variables can be easily set and retrieved when using ClinicaDL. First, the code and documentation of ClinicaDL are versioned to allow the user to retrieve the version needed for method reproducibility. Then, as explained in Section 6.2.1, two files at the root of the experiment folder identify the software and dependencies' versions (`environment.txt`) and variables such as threading, GPU usage and random seed (`maps.json`). Moreover, the function `clinicadl train --config_file` was designed to repeat experiments based on a configuration file (see Section 6.3.2). However, we remind that it is still the users' responsibility to describe their GPU system.

As explained by Beam et al., 2020 and Baker, 2016, documentation is also a crucial point to ensure transparency and code usability by other teams, which then allows result reproducibility. This is why ClinicaDL comes with different documentation supports, including tutorials (see Section 6.3.1).

Finally, Goodman et al., 2016 and Stodden et al., 2016 encourage others to report all the explored paths and negative results to be more transparent and to avoid potential bias in reporting. This process may also avoid unfair claims (inferential non-reproducibility) based on the comparisons to weak baselines (Crane, 2018). Indeed, the performance of deep learning systems highly depends on the time spent on their design. This is why it could be interesting to report all the architectures trained to find the final system compared with the ones trained for the baseline one. Again, ClinicaDL allows easily compiling this information as the (hyper)parameters of all the networks trained are saved in their MAPS.

6.3 ClinicaDL overview

ClinicaDL is an open-source software platform entirely written in Python. It uses the PyTorch library as backbone. ClinicaDL extends PyTorch features for neuroimaging applications where the data set structure plays a key role in the organisation of the data and metadata. The software is publicly distributed as an easy-to-install package and is referenced in the Pypi package index[28]. Releases are done on a periodic basis and the code follows the most standard current practices for software development.

ClinicaDL has been designed to be used via the command line interface, with separate sub-commands performing the main tasks, as defined in a classical machine learning pipeline: extract, train, predict. Other sub-commands are available in order to allow the user to structure the data sets, create synthetic data, look for hyperparameters and interpret trained networks. These features are also available through the command line (tsvtool, generate, random-search, interpret).

6.3.1 Development Practices

ClinicaDL has adopted standard practices for software development and distribution of the software with the aim to facilitate the reproduction of experiments. The main features of the software, the management of its inputs/outputs, the data flow and the way the program is used were designed with the objective of staying as close as possible to the definition of reproducibility, as previously given in Section 6.2.3. The main development practices are described below.

Distribution and Installation

The source code is hosted on Github[29]. It uses a version control system and the releases are strictly labeled with the version number. As consequence, the source code used in a specific experiment can be easily retrieved. Labeled versions of the code are released as Python packages that are permanently stored in the official Python Package Index. Good practices related to the version control system include atomic committing, clear commit messages and peer-reviewed contributions.

[28] https://pypi.org/project/clinicadl
[29] https://github.com/aramis-lab/ClinicaDL

The installation of the released packages is done with a single command (`pip install clinicadl`). As often when installing Python packages, users are advised to install it into a virtual environment to avoid requirement conflicts. Instructions for developer installation are also available in the `README` of the repository.

Continuous Integration and Deployment

Each contribution is peer-reviewed by a developer different from the original author. The resulting code is only integrated to the development branch if the post commit actions are executed in a satisfactory way. The ensemble of these actions is described in the Continuous Integration pipeline. This includes:

- **Environment and dependencies verification**: The creation of an environment with all the dependencies necessary to install the package is performed in this step.
- **User interface tests**: The command line interface is tested using the Pytest library. This library allows combining several sets of possible commands used in the user interface. These are systematically tested to avoid errors in the main interface of ClinicaDL.
- **Functional tests**: A different kind of tests is executed before the integration of new code. These tests are called functional tests and are designed to check for the proper operation of the different tasks proposed by the software: e.g. "Train", "Transfer Learning", "Interpretation" and "Random Search" tests use a truncated data set to verify that these tasks run properly on a GPU machine. Other functionalities such as "Predict" to perform inference, "Generate" to create custom data sets or "TSV Tools" to generate files adapted to the task / data set are also checked.
- **Documentation build**: New contributions and/or modifications to the code are expected to be accompanied by the respective documentation. For that reason, documentation is built during the continuous integration pipeline. More details are given in Section 6.3.1.
- **Deployment**: This step is only executed on labeled commits. Indeed, if a commit has a label to reference a version, a Python package is built and uploaded to the Python Package Index and a new version is published.

Model distribution

The work described in Chapter 3 used a preliminary version of ClinicaDL. Several pretrained models generated from the methods described in this work are available to download via Zenodo[30]. These models are also publicly available via a classical https server[31] to facilitate interactive downloading. New versions of the software may induce changes on the organisation of the available models and the way they are loaded and processed by

[30] https://zenodo.org/record/3491003
[31] https://aramislab.paris.inria.fr/files/data/models/dl/

ClinicaDL. For these reasons, new models are trained regularly and stored in folders named with the corresponding software version, e.g. `models_v020` corresponds to the models trained with the version 0.2.0 of ClinicaDL.

Documentation

The documentation of ClinicaDL is available online at https://clinicadl.readthedocs.io. It is automatically built after each commit by Read the Docs[32]. It can also be built locally by running the command `mkdocs serve` with mkdocs-material installed[33].

The documentation is versioned in the same way as the source code. All previous tags are easily accessible online with the version panel in the bottom right corner of any page.

In addition to the user documentation, some tutorials have been created to help users in their first steps with ClinicaDL. These tutorials are designed with Jupyter Books[34], a tool that mixes Markdown content with interactive notebooks. They are referenced in the documentation and GitHub main pages and are accessible online at https://aramislab.paris.inria.fr/clinicadl/tuto. The first sections introduce the clinical context of Alzheimer's disease and basics on deep learning classification. The rest of the book is made of interactive notebooks that present the main functionalities of ClinicaDL and can be easily run locally or on Google Colab (which provides free GPU environments).

Finally a discussion forum is available at https://groups.google.com/g/clinica-user. This tool will be replaced soon by GitHub discussions https://github.com/aramis-lab/clinicadl/discussions.

6.3.2 Main functionalities

The main functionalities of ClinicaDL cover all the steps needed for deep learning experiments, from data set management to the evaluation of results and network interpretation. In addition to pre-implemented features, the source code aims to be modular and the documentation helps users to implement easily their custom experiments[35]. Technical details for each command can be found in the user documentation.

Preprocessing images

ClinicaDL works preferably with images that had been previously preprocessed but one can also perform experiments with unprocessed images, the only requirement is to convert these images to the right format (see Section 6.2.1). Preprocessed images can be obtained using Clinica for different imaging modalities. This software provides, in its current version (0.5.0), light preprocessing pipelines for T1w and PET images that output images suited for further deep learning. For example, the `t1-linear` pipeline mainly performs bias

[32] https://readthedocs.org/
[33] https://squidfunk.github.io/mkdocs-material/getting-started
[34] https://jupyterbook.org/intro.html
[35] https://clinicadl.readthedocs.io/en/latest/Contribute/Custom/

field correction and spatial normalization to the MNI space of T1w MR images, while the `pet-linear` pipeline mainly performs spatial normalization to the MNI space and intensity normalization of PET images. As ClinicaDL and Clinica are fully compatible, outputs of the formerly mentioned pipelines can be introduced easily into a train or classification function of ClinicaDL.

ClinicaDL proposes a simple tool to transform NIfTI images into PyTorch format. The objective is to facilitate the training phase by decompressing the images beforehand (the NIfTI format usually provides compressed images). This functionality writes future input images for neural network training or inference formatted as tensors. The number and shape of these tensors depend on the mode chosen: `image`, `patch`, `roi` or `slice` (see Section 6.2.1).

The tool will run through the entire CAPS/BIDS folder searching for an imaging modality specified by the user and will apply the conversion and extraction of corresponding images. It will also produce a configuration file summarizing all the characteristics of the extraction procedure. The training procedure will then rely on this file to find the images needed for network training.

Generation of toy data sets

ClinicaDL facilitates the generation of semi-synthetic data for evaluation and verification purposes. The new data can be used to test a binary classification task, and it is already organized in the CAPS format (see Section 6.2.1). Two types of data can be created:

- **Trivial data**: A mask is used to create incomplete images. By default, a mask based on a neuroanatomical atlas is used to create images where only half of the brain is present (half-left or half-right). Other kinds of distortions can be created by supplying a customized mask. The final result is the suppression of the region present in the mask.
- **Random data**: All the images belonging to this type of data are obtained from a single image, adding random white noise. The standard deviation of the noise is a parameter chosen by the user. Resulting images are then randomly distributed between two possible labels.

Preparing metadata

To use the train and inference functionalities of the software or to analyse the data, inputs must be organized in the right way. A collection of tools to handle metadata of BIDS-formatted data sets is proposed with ClinicaDL. These tools are intended to provide the correct organisation of the data: get the labels used in classification tasks, split the data to define test, validation and train subsets, and analyze the population of interest. This set of commands is available through the command `clinicadl tsvtool`. Some of them are still specific to the study of Alzheimer's disease:

- Generation of TSV files including only participants with particular restrictions on two Alzheimer's disease data sets (AIBL and OASIS).
- Extraction of labels specific to a particular diagnosis trajectory (e.g. participants labeled with an Alzheimer's disease diagnosis for all their sessions).

Other commands are more generic, and may be applied to any label list, even if they were not generated with ClinicaDL:

- Splitting labels to produce similar distributions from a specific population using as parameters sex and age.
- Splitting labels to perform k-fold cross validation.
- Writing reports to summarize the demographics and clinical distributions of a specific label.

Random search

Random search (Bergstra and Bengio, 2012) is a procedure to find automatically the hyperparameters (architecture and other training hyperparameters) of a framework. It consists in randomly generating sets of hyperparameters to select the best set of hyperparameters as a result. This random generation is based on a hyperparameter space from which hyperparameter sets are sampled. In ClinicaDL, this hyperparameter space is described by a configuration file created by the user.

The main advantage of the random search is its easy parallelization, contrary to other optimization methods that may require successive runs and be time consuming. On the other hand, it is computationally costly and it requires minimum knowledge regarding the subspace of hyperparameters that may work to limit the search and find satisfying results. Moreover, although it can significantly improve the performance of a framework, it will not lead to the optimum, which is very hard to find.

It is also possible to improve the results of a random search by using its results to initialize another technique (genetic algorithm, Bayesian optimization, etc.).

Training networks

The main functionality of ClinicaDL is to train neural networks to learn a task. These tasks can be:

1. **Classification** (of a categorical label, for example the diagnosis),
2. **Regression** (of a continuous label, for example the age),
3. **Image reconstruction**.

These tasks are highly dependent from the architecture. All tasks take as input the image or part of it (see Section 6.2.1); but for classification the output is a flattened array of size equal to the number of classes, for regression it is a single node, whereas for image reconstruction

it has the same size as the input. When the user chooses a task and an architecture, the software will check if the task is compatible with the wanted architecture and will raise an error if it is not the case.

Some pre-built architectures are already available in ClinicaDL, but an objective of the library is to allow the users to add their custom architectures easily. The procedure of such addition is detailed in the documentation.

The models produced by ClinicaDL correspond to the ones that obtained the best performance on the validation set according to metrics chosen by the user. ClinicaDL saves at the end of each epoch the state of the network and of the optimizer. For each selection metric given in input, it replaces the corresponding current best model by the current state if the performance on the validation set is better than the current best value. To minimize the size of the produced MAPS, the checkpoints are removed at the end of the training procedure. They are only used to resume a stopped job, thanks to the dedicated command `resume`.

The command line interface of ClinicaDL offers many options, as there is a large number of training parameters. This is why we tend to a parametrization by configuration files only. Currently, it is already possible to train a network parametrized by a configuration file instead of entering each parameter individually in the command line using `clinicadl train --config_file FILENAME`.

Performance evaluation

ClinicaDL provides specific functions to easily perform inference with models previously trained with the tool. This functionality is available in a specific sub menu of the command line (`clinicadl predict`). For example, one may want to evaluate the performance of a trained model on a set of new samples. In this case, the command will load the best model, the input images (in a BIDS/CAPS-like format) and the list of subjects of the data group. Trained models are available within the MAPS produced during the training and the other information can be either integrated into this structure or proposed as a command line option. The results are written in the MAPS as pre-formatted reports with the metric values at different levels (e.g. image-level and patch-level) and the output values computed for each input image of the data group.

Interpretation

The most critical issue of deep learning methods is their lack of transparency. This is why some interpretability methods have been developed specifically for the field. These methods allow better understanding which patterns or zones of the images have been linked to the result produced by the network. Currently, only the gradient back-propagation method of (Simonyan et al., 2014) is implemented in ClinicaDL. We plan to strengthen the content of this command in future releases.

6.4 Discussion

In this chapter we presented ClinicaDL, a Python open-source software for neuroimaging data processing with deep learning. This software includes many functionalities, such as neuroimaging preprocessing, synthetic dataset generation, label definition, data split with similar demographics, architecture search, network training, performance evaluation and trained network interpretation. The three main objectives of ClinicaDL are to (1) help manipulate neuroimaging data sets, (2) prevent data leakage from biasing results and (3) reproduce deep learning experiments.

First, ClinicaDL relies on BIDS and CAPS formats to organize raw and processed data, respectively. Though these formats were first introduced for neuroimaging data management, they can be easily extended to any kind of medical imaging data, as it would only require renaming and formatting files of a data set.

Secondly, ClinicaDL prevents data leakage as train and validation data characteristics are saved when the output structure (MAPS) is created. Then, when evaluating the performance of a trained model on a new data group, ClinicaDL checks that this data group is independent from the training and validation groups. However, this only works under the assumption that participants are always named in the same way across data groups. For example, the cohorts OASIS-1, OASIS-2 and OASIS-3 comprise common participants anonymized with different names depending on the cohort. If OASIS-3 is used for training and OASIS-1 for test evaluation, ClinicaDL will not detect that there is an intersection between training and test data. Then it is the responsibility of the users to check the independence of their data sets.

Thirdly, ClinicaDL improves deep learning experiment reproducibility by sharing usable and tagged code, saving all parameters of the training set and data groups used for evaluation, and providing extensive documentation. However, though all these elements improve method reproducibility, reproducibility can still be easily broken. For example Crane, 2018 explained that using another GPU system may make the results irreproducible. Then it may not be possible for two different users to obtain the same results on different machines. However, one user may be interested in having a deterministic setting to correctly evaluate the impact of one particular property to improve their performance. Moreover, result reproducibility may also be broken by manual architecture search and the overuse of the same data set (Thompson et al., 2020). Indeed, research studies may be globally overfitting this data set and if one day another data set is released, performance of previous studies may collapse. This is why we implemented the random search method, although its very high computational cost may limit its reproducibility power. In conclusion, as reproducibility is a property which may be broken by many aspects of a study, we advise data scientists to refer to reproducibility checklists made available online[36] to ensure that their work is (largely) reproducible.

Among the three main issues tackled by ClinicaDL, the one which is the most often addressed is the first (data management). For example, TorchIO and Monai ease the use

[36] https://miccai2021.org/files/downloads/MICCAI2021-Reproducibility-Checklist.pdf

of some public data sets (MedMNIST, medical segmentation decathlon challenge, IXI and EPISURF), then other data sets can be plugged to the library components by specifying individually the paths to images and labels (Nobrainer only offers this second option). This way, newcomers can easily begin to handle the libraries by running examples based on integrated data sets, and then try to use their own. The main default of this system is the lack of reproducibility: the list of participants used must be saved by the user independently, and the preprocessing information may be lost. This is not the case with ClinicaDL as the characteristics of each group are saved in the MAPS and the preprocessing is fully described in the configuration file. TorchIO and Monai also deal with reproducibility: TorchIO guarantees that transforms with a random factor can be reproduced as one can get the transforms' history, and Monai allows setting a random seed to compute a deterministic training. However, they do not propose any system similar to the MAPS, thus experiment settings and environment versions may be lost by the user. Finally, none of the libraries reviewed proposed systems to avoid data leakage, though it is a crucial issue in our domain.

As exposed in Section 6.3.2, the association of Clinica and ClinicaDL covers a large variety of procedures needed in deep learning experiments, that starts from the raw data format and ends with network interpretability. On the contrary, TorchIO is more specialized as it focuses on medical imaging transforms, particularly for data augmentation. This focus results in a larger amount of options in this particular domain. This is why we consider integrating modules from TorchIO for data augmentation in the future. Monai is more complete with the possibility to transform images, and train, evaluate and interpret networks. They also provide a large amount of options for many features which are not customizable yet in ClinicaDL (for example the loss and the optimizer). We would also like to enable the parametrization of more features, whose options could be easily added by advanced users.

ClinicaDL aims at being flexible, thus a section of the documentation is dedicated to the addition of new options of the main features. At the present time it is possible to customize:

- the architecture of the network (dependent from the task learnt),
- the mode extracted from the 3D image (current options: image, patch, roi, slice),
- the task learnt by the network (current options: classification, regression, image reconstruction),
- the metrics used for evaluation and best weights selection (dependent from the task learnt).

Even if some options are not already integrated to the framework, we hope that advanced users will be prone to propose new pipelines to the repository.

Conclusion

Summary

The goal of this PhD was to characterize Alzheimer's disease heterogeneity to define clinically relevant subtypes. I wanted to resume the work done during my research internship, during which I defined subtypes based on the gray matter probabilities in different brain regions of demented patients using hierarchical clustering (unpublished work). Though this work led to interesting conclusions, its main limitation was the use of highly-engineered features, which we decided to address by using deep learning methods. But as deep learning methods are non-transparent models, I also planned to spend a great deal of time searching for an appropriate way to interpret their result to better characterize the obtained subtypes.

However, a new objective became central in this work: to ensure the reproducibility and methodological relevance of the scientific contributions. Indeed, we discovered during the first literature review performed at the beginning of the PhD that many studies were contaminated by methodological flaws, and thus could not be considered to design our own framework (chapter 1). Moreover, an extensive literature search on interpretability methods applied to deep learning showed that the field is not mature yet: a large amount of methods exists, and their limitations are not always correctly assessed before being used by the community, leading again to irrelevant results. This is why interpretability metrics were designed to allow a better comparison of methods. Unfortunately, again, this field is not mature enough to allow fair and robust comparisons, and it was already demonstrated that some of these metrics were unreliable (chapter 2).

The conclusions of these two literature reviews led to two experimental studies. First, to better assess the relative performance of commonly used deep learning frameworks, a benchmark was conducted (chapter 3). The diversity of deep learning frameworks comes from the way the inputs are extracted from magnetic resonance images: they could be 2D slices, 3D patches browsing the whole image, a selection of regions, or the whole image. We observed that a conventional machine learning system (support vector machine) had a similar performance to those of deep learning methods, though a more extensive preprocessing procedure was needed for the former. Moreover, this did not help us much to choose the most relevant method because 3D patches, regions and image-based frameworks had similar results (only 2D slices could be excluded). However, to minimize the number of prior hypotheses made on the whole framework, we chose to work with image-based models for further studies as for patches and regions we needed to choose hyperparameters (patch size and stride, or the regions to include). But this observation also

raised an issue for the next step of the PhD: if networks obtained the same results based on the hippocampi only and the whole image, this could mean that the network only learns information from the hippocampi and neglects other parts of the image.

This intuition was confirmed in the second experimental study, which primary goal was to assess the robustness of convolutional neural networks (CNNs) in detecting patterns for Alzheimer's disease detection (chapter 4). In this study, we decided to use the optimized perturbation method. Indeed the paradigm of perturbation methods is very human-friendly as we can reproduce it ourselves: if the part of the image that is needed to find the good output is hidden, we are also not able to predict correctly, whereas we are able to correctly predict from an image in which parts uncorrelated to the task are perturbed. Moreover, the regularization of this method allowed producing smoother attribution maps than many methods based on back-propagation. The first part of the study consisted in evaluating the robustness of the method by evaluating the similarity of masks produced for images of the same participant, images of different participants, different hyperparameters and different groups of data (ADNI and AIBL). We concluded from these experiments that the method is reliable and that results are consistent using different sets of hyperparameters and similar groups of data. Then, we could use this method to assess whether CNNs consistently focused on the same set of regions when retraining the same framework and when using different train / validation splits. Unfortunately, we observed that even though the regions highlighted are relatively consistent (mostly in the medial temporal lobe), the similarity between two CNN retrainings was lower than the similarity of the maps of two different individuals. It means that the regions found by a CNN are mainly due to random factors, and also that it is not able to find the atrophy pattern specific to a participant.

We tried to find a solution to better characterize individual patterns by capturing all the regions relevant for Alzheimer's disease detection. The first option that we explored consisted in training successive CNNs to learn different patterns by masking regions already identified as salient. CNNs were always trained to distinguish demented patients from cognitively normal participants. At the end of a training phase, the attribution map was computed with the perturbation method. This attribution map was then applied to mask all inputs during the next training phase, to ensure that the network learns other patterns than the already found ones. We repeat this process of CNN training / masking when the CNN would not be able to classify anymore from the masked input. This would mean that all the regions encoding information would be found by the attribution maps. Unfortunately, this framework did not succeed due to a limitation of the optimized perturbation method: the formation of artifacts. Instead of masking information relevant to the network, the mask was optimized to generate gray matter outside of the training distribution to perturb the network. We were unable to deal with this issue, though we tried several solutions, such as training a denoising autoencoder to detect these artifacts and avoid their creation by adding a new loss term based on autoencoder reconstruction. This method is briefly described in Appendix E.

Though these artifacts only appeared in this new context, after these experiments we decided to abandon the optimized perturbation method to use gradient back-propagation

instead. This choice was also motivated by the computational cost of the methods: optimized perturbation required an additional learning step (successive forward and backward passes on a whole data set) whereas gradient back-propagation only required a backward pass per image. Then, in the last experimental study of this PhD we tried another approach to solve the problem of individual pattern characterization: data augmentation (chapter 5). Indeed, we first showed thanks to a simulated framework that the detection of patterns specific to a subtype in a heterogeneous class was enhanced when increasing the number of samples. These simulated data sets, in which atrophy patterns were known, also allowed assessing the reliability of the approach. However, the validation of the method on real data using two known diseases to control the accuracy of the subtyping method failed. This is why we collaborated with another team developing a data augmentation based on variational autoencoders to assess whether their method could improve our framework. As the application of their data augmentation was successful on the Alzheimer's disease vs cognitively normal task, we then tried to use it to improve our own setting. Unfortunately, it was still not efficient enough to allow the identification of the two diseases mixed under the same label.

Finally, one major contribution of this PhD is the creation of the Python library ClinicaDL (chapter 6). This library is the deep learning extension of the software already created and maintained by the team: Clinica. At first, the purpose of ClinicaDL was to have an open-source framework allowing the reproduction of the experiments done during this PhD. But the scope of this library is now larger, as external users began to use it as well.

Perspectives

The main limitation remaining at the end of this PhD is the following: the latent space of our CNN trained to differentiate demented patients from cognitively normal participants mixes all patients subtypes, preventing us from retrieving their individual characteristics.

Though data augmentation was a good option according to the simulated data, it did not translate to real cases, though we might see a small improvement between the basic data augmentation procedure and the advanced one based on a variational autoencoder (the attribution maps obtained with the latter are a bit more scattered and less focus on the hippocampus only). This confusion between subtypes may come from the fact that the network brings closer patients when trying to separate the normal from the abnormal distribution. To avoid this effect, one solution could be to consider methods from the "anomaly detection" field. In these methods, an algorithm learns only the normal distribution (in our case cognitively normal participants) to only find if and where a new sample is abnormal. We began to train variational autoencoders on cognitively normal participants, and observed that different patient groups (Alzheimer's disease or fronto-temporal dementia) could be distinguished in its latent space. This could be a valuable option to explore in the future. When the method will be mature enough, it would be of great value to apply it on data sets closest to the clinical routine, with images presenting artifacts and even more heterogeneity, both in terms of population and pathological processes.

Other limitations remain, such as the robustness of CNN training and the creation of artifacts with optimized perturbations. Though we did not investigate it much at the end of the PhD, the patterns detected by the CNN may become more stable thanks to data augmentation, but this remains to be proven. The second point, artifact creation, is much more difficult to deal with. Indeed, it can be easier for the mask to create patterns outside of the training distribution to prevent a trained network from finding the correct label. Though we built an autoencoder able to find if an image is outside of the training distribution, this new component was not robust enough to prevent the mask from creating artifacts during its optimization. This is why other setups should be considered, such as adversarial training, during which the autoencoder could learn to remove artifacts at the same time the mask is optimized. Moreover, we concluded from the results of Appendix D that intensities in an attribution map could not be easily compared to find which regions are more important to the network. A good initiative could be to run a benchmark of interpretability methods on simulated data with different geometrical shapes and check whether some of them are less receptive to the shape of regions of interest than gradient back-propagation.

Finally, a major issue encountered during this PhD is the lack of reproducibility and reliability of studies in the deep learning field. We contributed at our scale with the development of ClinicaDL, which was thought to help deep learning users unfamiliar with the neuroimaging field and avoid common pitfalls that may bias their results. However, though this issue is more and more discussed by scientists (Hutson, 2018), and initiatives launched[37], it will only be solved by applying also more global solutions, such as bettering the peer-review process or allocating more resources to scientists.

[37] http://slow-science.org/

Appendix A

Field strength bias in ADNI cohort

In this appendix, experiments were conducted after the publication of Chapter 3 to check that the classification tasks we performed in this article are not biased towards MRI field strength. Indeed, the ADNI data set we used in the corresponding experiments includes different cohorts (ADNI-1, ADNI-GO and ADNI-2) in which different imaging protocols were used. In particular, the 1.5 T magnetic resonance imaging (MRI) machines were progressively phased out (replaced by 3 T MRI), and recruitment targeted different MCI groups. In detail:

- in ADNI-1, 1.5 T machines were used and the MCI patients recruited were late MCI,
- in ADNI-GO, 1.5 T machines were replaced by 3 T machines for new participants included in the study and the MCI patients recruited were early MCI.

The definitions of early and late MCI appeared with the ADNI-GO cohort: the goal of this cohort was to characterize the stage that precedes MCI patients already enrolled in ADNI-1. Levels of MCI (early or late) are determined using the Wechsler Memory Scale Logical Memory II, quantifying memory impairment. Because of this recruitment criteria, early MCI patients are more likely to stay stable, whereas late MCI patients will convert sooner, leading to sMCI and pMCI populations with different field strength distributions. This change at the same time in the recruitment process and the machines used for MRI may induce a bias in the sMCI vs pMCI classification task performed on ADNI.

Moreover, several deep learning studies (including our previous work) evaluating the evolution of MCI status mixed the different cohorts of ADNI to create their population (Basaia et al., 2019; El-Sappagh et al., 2020; Wen et al., 2020) or did not mention this issue and used ADNI as a homogeneous data set (Gao et al., 2020; Lee et al., 2019c; Shmulev et al., 2018; Zhang and Shi, 2020). This is why we attempted to assess the risk of bias in such studies, with two different experiments. First we evaluated the ability of a CNN to find the field strength of MR images and which performance this CNN had when directly applied to the clinical task sMCI vs pMCI. Then, we reused the networks trained in our previous study to show the existence of bias in some of our previously published results.

							ADNI cohorts			
Label	Subjects	Sessions	Age	% Female	MMSE	% 1.5T	1	GO	2	3
sMCI	266	1 105	72.4 (7.3)	40.0%	27.9 (1.7)	33.1%	88	61	114	3
pMCI	328	918	74.4 (7.1)	41.4 %	26.7 (1.9)	64.8%	193	11	111	13

TABLE A.1: Summary of participant demographics, mini-mental state examination (MMSE) score, field strength and number of sessions of ADNI cohorts at baseline. Values are presented as mean (SD).

A.1 Materials

We included all recruitment phases of ADNI: ADNI-1, ADNI-GO, ADNI-2 and ADNI-3 (data released before January 26, 2021). Some participants may be followed across several phases, then they are not independent. Two diagnosis groups were considered:

- pMCI: sessions of subjects who were diagnosed as MCI, and progressed to AD during the 36 months following the current visit;
- sMCI: sessions of subjects who were diagnosed as MCI, and neither progress nor regress to AD during the 36 months following the current visit.

Table A.1 summarizes the demographics, clinical scores, MRI field strength and distribution in ADNI cohorts of the participants. We observe in this table that there is a difference between sMCI and pMCI field strength distributions. Then an algorithm learning the sMCI vs pMCI classification task may take advantage of the field strength distribution.

A.2 Methods

A.2.1 Field strength classification task

MRI preprocessing

We used the N4ITK method (Tustison et al., 2010) for bias field correction. Then, T1w-MR images were linearly registered (Avants et al., 2008) to the MNI space (ICBM 2009c nonlinear symmetric template) and cropped to remove all rows and columns containing background voxels only. Finally we rescaled intensity between 0 and 1. All data mangement and preprocessing was carried out using the Clinica software (Routier et al., 2021).

Data split

We considered all the ADNI protocols (1, GO, 2 and 3), which we split into training/validation and test sets. Our test set consisted of 100 subject chosen to be a representative subset (according to age, sex and field strength distributions) of each diagnostic class. We used the rest of the ADNI data set as training/validation set. We trained the models using the training/validation dataset. Training and validation sets were generated with a 5-fold cross-validation stratified according to the field strength value (to ensure that the field strength distribution is equivalent in all folds), which resulted in one fold (20%) of the data for validation and the rest for training. As we used longitudinal data,

all splits were performed at the participant level to ensure no data leakage between the training, validation or test sets.

CNN architecture & training - 1.5 T vs 3 T

We trained the network to differentiate 1.5 T from 3 T MRI by optimizing the cross-entropy loss during ten epochs. We used 3D subject-level architecture of Chapter 3. This CNN consists of five convolutional blocks and three fully-connected layers. Each convolutional block is sequentially made of one convolutional layer, one batch normalization layer, one ReLU and one max pooling layer.

The final model was the one that obtained the highest validation balanced accuracy during training. The balanced accuracy of the model was evaluated at the end of each epoch. Network training and inference was performed with ClinicaDL (see Chapter 6).

Evaluation procedure

After training to differentiate 1.5 T from 3 T MRI, the network was applied to two binary classification tasks: 1.5 T vs 3 T and sMCI vs pMCI. We present the mean balanced accuracy of the models applied to the test set, followed by the five mean balanced accuracies of each model obtained on a fold of the 5-fold cross-validation between squared brackets.

A.2.2 Quantifying bias in previously published results

In Chapter 3, we trained networks to differentiate AD patients from cognitively normal (CN) participants and sMCI from pMCI patients on the first three cohorts of ADNI (ADNI-1, GO and 2), using MRI preprocessed with the procedure described in section A.2.1. We used different types of inputs by extracting sub-parts of the MRI:

- **image** corresponds to the whole 3D MRI (this is the input which was used in the previous section of this study),
- **patch** corresponds to 36 patches of size $50 \times 50 \times 50$ voxels with no overlapping covering the whole MRI,
- **roi** corresponds to two cubic patches encompassing the left and right hippocampi,

We then used the following method: the balanced accuracy was evaluated separately for 1.5 T and 3 T images on the test set. If these balanced accuracies are both lower than the original one (on the whole test set) then results are biased towards the field strength. To evaluate the difference between the balanced accuracy on the whole test or only one field strength we performed a paired t-test on the two series of five folds for each experiment. If the balanced accuracies on the 1.5 T and 3 T subsets are significantly lower than on the whole set, then we can conclude that the network partly learned the field strength.

Input	Training data	1.5 T	3 T	1.5 T+3 T	p-values (1)	p-values (2)	p-values (3)
image	Baseline	**0.80** [0.79, 0.87, 0.80, 0.75, 0.81]	**0.84** [0.79, 0.82, 0.84, 0.86, 0.90]	**0.82** [0.79, 0.84, 0.82, 0.80, 0.85]	0.21	0.21	0.21
	Longitudinal	**0.86** [0.88, 0.85, 0.86, 0.88, 0.83]	**0.83** [0.89, 0.82, 0.82, 0.79, 0.84]	**0.85** [0.88, 0.83, 0.84, 0.84, 0.83]	0.28	0.28	0.28
roi	Baseline	**0.86** [0.85, 0.86, 0.88, 0.88, 0.86]	**0.91** [0.89, 0.91, 0.91, 0.94, 0.92]	**0.89** [0.86, 0.88, 0.90, 0.91, 0.89]	<0.01	<0.01	<0.01
	Longitudinal	**0.84** [0.86, 0.8, 0.85, 0.88, 0.83]	**0.87** [0.88, 0.84, 0.89, 0.83, 0.91]	**0.85** [0.86, 0.82, 0.86, 0.85, 0.86]	0.23	0.23	0.23
patch	Baseline	**0.78** [0.76, 0.76, 0.82, 0.78, 0.76]	**0.86** [0.89, 0.86, 0.86, 0.83, 0.83]	**0.81** [0.82, 0.81, 0.84, 0.80, 0.79]	<0.01	<0.01	<0.01
	Longitudinal	**0.86** [0.87, 0.88, 0.84, 0.86, 0.86]	**0.84** [0.84, 0.83, 0.90, 0.83, 0.81]	**0.85** [0.85, 0.86, 0.86, 0.84, 0.83]	0.46	0.46	0.46

TABLE A.2: Comparison of balanced accuracies for task AD vs CN of deep learning methods obtained on 1.5T, 3T and all data available. P-values correspond to the following paired t-tests: (1) 1.5 T vs all, (1) 3 T vs all, (1) 1.5 T vs 3 T.

A.3 Results

A.3.1 Field strength classification task

To assess whether there is a risk that a network learns the field strength instead of the diagnosis status, we trained CNNs to detect the field strength, i.e. 1.5 T vs 3 T, using the T1w-MR images of sMCI and pMCI patients from ADNI.

The CNN perfectly learns to differentiate field strengths in our population by obtaining a balanced accuracy of 0.98 [0.98, 0.96, 0.98, 0.98, 0.98]. Moreover, the direct application of the networks to the sMCI vs pMCI led to a balanced accuracy higher than chance 0.65 [0.65, 0.65, 0.65, 0.64, 0.66] and of similar value as the ones that could be obtained by networks trained on sMCI vs pMCI (0.68 when using the whole image as input, see Table A.3). Then we checked whether our previously published results were contaminated by this bias.

A.3.2 Quantifying bias in published results

We evaluated the presence of bias in our previous work. Results are displayed in Tables A.2 and A.3. The original value always lies between the values obtained for 1.5 T and 3 T for AD vs CN, then we cannot conclude to the learning of the field strength by the network. However, we note that for **roi** and **patch** the results are much better on 3 T images than 1.5 T images when using only baseline data, whereas all balanced accuracies are equivalent when using longitudinal data.

Input	Training data	1.5 T	3 T	1.5 T+3 T	p-values (1)	(2)	(3)
image	Baseline	**0.63** [0.61, 0.73, 0.49, 0.69, 0.66]	**0.60** [0.63, 0.62, 0.51, 0.66, 0.60]	**0.68** [0.68, 0.71, 0.64, 0.73, 0.67]	0.16	<0.01	0.27
	Longitudinal	**0.71** [0.73, 0.70, 0.68, 0.73, 0.69]	**0.67** [0.66, 0.65, 0.69, 0.68, 0.67]	**0.73** [0.74, 0.71, 0.72, 0.74, 0.72]	0.02	<0.01	0.07
roi	Baseline	**0.70** [0.67, 0.70, 0.68, 0.72, 0.72]	**0.70** [0.74, 0.64, 0.74, 0.70, 0.70]	**0.74** [0.75, 0.72, 0.76, 0.74, 0.75]	0.02	0.03	0.82
	Longitudinal	**0.70** [0.66, 0.69, 0.70, 0.67, 0.76]	**0.70** [0.65, 0.67, 0.65, 0.74, 0.78]	**0.74** [0.70, 0.72, 0.72, 0.75, 0.80]	<0.01	0.16	0.96
patch	Baseline	**0.56** [0.50, 0.51, 0.60, 0.59, 0.60]	**0.58** [0.68, 0.50, 0.54, 0.61, 0.58]	**0.68** [0.71, 0.64, 0.64, 0.71, 0.69]	0.01	<0.01	0.64
	Longitudinal	**0.62** [0.64, 0.63, 0.60, 0.63, 0.60]	**0.58** [0.58, 0.60, 0.66, 0.54, 0.53]	**0.70** [0.70, 0.71, 0.69, 0.71, 0.69]	<0.01	<0.01	0.20

TABLE A.3: Comparison of balanced accuracies of deep learning methods for task sMCI vs pMCI obtained on 1.5 T, 3 T and all data available. P-values correspond to the following paired t-tests: (1) 1.5 T vs all, (1) 3 T vs all, (1) 1.5 T vs 3 T.

On the contrary, bias was detected on sMCI vs pMCI results. Indeed, each time the 1.5 T and 3 T results are both significantly different from the original values (except for **image** where only the 3 T series is significantly different from the original values). Then we observe a significant drop in balanced accuracies of 1.5 T and 3 T compared to the original one for **patch** CNN (between 12 and 8 percent points). The **image** and **roi** CNNs are also affected by this bias, but not to the same extent, with drops between 8 and 2 percent points for **image** and drops of 4 percent points for **roi**. We guess that the **patch** experiments are more affected than **image** or **roi** ones as in some patches at the edge of the brain no information relevant to the diagnosis can be found, then the only useful information is the field strength. This behaviour may also be found when using **slice** inputs.

A.4 Conclusion

This study started from the observation the sMCI/pMCI status was associated with MRI field strength because of a recruitment bias in ADNI. We showed that CNNs could successfully learn to differentiate 1.5 T from 3 T MRI, and that a field strength predictor would achieve a 65% balanced accuracy in ADNI. We further observed that sMCI/pMCI predictors would learn the data structure, leading to inflated prediction accuracy. Our case example demonstrates how field strength acts as a confounder on sMCI vs pMCI results. We showed that previous results (including a previous publication from our group) reported inflated prediction accuracy of the sMCI vs pMCI task. This could partly explain the

low generalisability of the prediction onto other test set, such as the Australian Imaging, Biomarkers and Lifestyle (AIBL).

Beyond this specific example, bias may be present in other studies or dataset, and may cause an overestimation of the performance of machine learning algorithms. In addition to the MRI field strength, several other confounders have also been flagged in the neuroimaging literature. They include age of the participants, sex, site, MRI machine, body size (e.g. height, weight, BMI) or head motion. Recently, a large scale examination has suggested many possible confounders of structural MRI studies (Alfaro-Almagro et al., 2021). Importantly, the presence and effect of the putative confounding factor are dependent on each dataset and trait/disorder of interest, and in some cases several confounding factors can contribute to prediction bias.

To avoid this pitfall we can only recommend future studies to more systematically take into account putative confounders. Several approaches may be used, such as the post-hoc ones we implemented here, which consists in evaluating prediction accuracy in subsets of the sample, or evaluating generalisability of the prediction into specific subsets of participants (e.g. into 1.5 T images). Another approach consists in controlling for known counfounders when evaluating the prediction accuracy. For example, one may use a generalised linear regression framework, with confounders fitted as covariates. In practice, this framework is powerful and versatile, in that it allows controlling for all confounders at once and even their interactions. On the other hand, confounders may also be dealt with during the training of algorithms. For example one could over-sample or put more weight on rare samples (here sMCI patients with 1.5 T images and pMCI patients with 3 T images).

Appendix B

Supplementary results of Chapter 3

B.1 Architectures hyperparameters

TABLE B.1: Architecture hyperparameters for 3D subject-level CNN.

As the architecture depends on the size of the input, it slightly differs between the two types of preprocessing (i.e. “Minimal” or “Extensive”). This difference only affects the size of the input of the first FC layer (FC1). The output size of each layer is reported depending on the preprocessing used in the last two columns.

The padding size in convolutional layers has been set to 1 not to decrease the size of the convolutional layer outputs. Without any padding, the number of nodes at the end of the last convolutional layer is too small to reconstruct the image correctly using an autoencoder for the Extensive preprocessing.

The padding size in pooling layers depends on the input: columns of zeros are added along a dimension until the size along this dimension is a multiple of the stride size.

BN: batch normalization; Conv: convolutional layer; FC: fully connected; MaxPool: max pooling.

Layer	Filter size	Number of filters / neurons	Stride size	Padding size	Dropout rate	Output size (Minimal)	Output size (Extensive)
Conv1+BN+ReLU	3x3x3	8	1	1	–	8x169x208x179	8x121x145x121
MaxPool1	2x2x2	–	2	adaptive	–	8x85x104x90	8x61x73x61
Conv2+BN+ReLU	3x3x3	16	1	1	–	16x85x104x90	16x61x73x61
MaxPool2	2x2x2	–	2	adaptive	–	16x43x52x45	16x31x37x31
Conv3+BN+ReLU	3x3x3	32	1	1	–	32x43x52x45	32x31x37x31
MaxPool3	2x2x2	–	2	adaptive	–	32x22x26x23	32x16x19x16
Conv4+BN+ReLU	3x3x3	64	1	1	–	64x22x26x23	64x16x19x16
MaxPool4	2x2x2	–	2	adaptive	–	64x11x13x12	64x8x10x8
Conv5+BN+ReLU	3x3x3	128	1	1	–	128x11x13x12	128x8x10x8
MaxPool5	2x2x2	–	2	adaptive	–	128x6x7x6	128x4x5x4
Dropout	–	–	–	–	0.5	128x6x7x6	128x4x5x4
FC1	–	1300	–	–	–	1300	1300
FC2	–	50	–	–	–	50	50
FC3	–	2	–	–	–	2	2
Softmax	–	–	–	–	–	–	2

TABLE B.2: Architecture hyperparameters for 3D ROI-based and patch-level CNN.

The padding size in pooling layers depends on the input: columns of zeros are added along a dimension until the size along this dimension is a multiple of the stride size.

BN: batch normalization; Conv: convolutional layer; FC: fully connected; MaxPool: max pooling.

Layer	Filter size	Number of filters / neurons	Stride size	Padding size	Dropout rate	Output size
Conv1+BN+ReLU	3x3x3	15	1	0	–	15x48x48x48
MaxPool1	2x2x2	–	2	adaptive	–	15x24x24x24
Conv2+BN+ReLU	3x3x3	25	1	0	–	25x22x22x22
MaxPool2	2x2x2	–	2	adaptive	–	25x11x11x11
Conv3+BN+ReLU	3x3x3	50	1	0	–	50x9x9x9
MaxPool3	2x2x2	–	2	adaptive	–	50x5x5x5
Conv4+BN+ReLU	3x3x3	50	1	0	–	50x3x3x3
MaxPool4	2x2x2	–	2	adaptive	–	50x2x2x2
Dropout1	–	–	–	–	0.5	50x2x2x2
FC1	–	50	–	–	–	50
Dropout2	–	–	–	–	0.5	50
FC2	–	40	–	–	–	40
FC3	–	2	–	–	–	2
Softmax	–	–	–	–	–	2

TABLE B.3: Architecture hyperparameters for 2D slice-level CNN.

The main layers are described in subtable (A). If skip connections are connecting two feature maps of different sizes, an intermediate downsampling layer (described in the subtable (B)) is used to resize the largest feature map.

BN: batch normalization; Conv: convolutional layer; FC: fully connected; MaxPool: max pooling.

Layer	Filter size	Number of filters / neurons	Stride size	Padding size	Dropout rate	Skip connection	Output size
Conv1+BN+ReLU	7x7	64	2	3	–	–	64x112x112
MaxPool1	3x3	–	2	1	–	–	64x56x56
Conv2+BN+ReLU	3x3	64	1	1	–	–	64x56x56
Conv3+BN	3x3	64	1	1	–	MaxPool1	64x56x56
Conv4+BN+ReLU	3x3	64	1	1	–	–	64x56x56
Conv5+BN	3x3	64	1	1	–	Conv3	64x56x56
Conv6+BN+ReLU	3x3	128	2	1	–	–	128x28x28
Conv7+BN	3x3	128	1	1	–	Conv5	128x28x28
Conv8+BN+ReLU	3x3	128	1	1	–	–	128x28x28
Conv9+BN	3x3	128	1	1	–	Conv7	128x28x28
Conv10+BN+ReLU	3x3	256	2	1	–	–	256x14x14
Conv11+BN	3x3	256	1	1	–	Conv9	256x14x14
Conv12+BN+ReLU	3x3	256	1	1	–	–	256x14x14
Conv13+BN	3x3	256	1	1	–	Conv11	256x14x14
Conv14+BN+ReLU	3x3	512	2	1	–	–	512x7x7
Conv15+BN	3x3	512	1	1	–	Conv13	512x7x7
Conv16+BN+ReLU	3x3	512	1	1	–	–	512x7x7
Conv17+BN	3x3	512	1	1	–	Conv15	512x7x7
AveragePool1	7x7	–	1	0	–	–	512x1x1
FC1	–	1000	–	–	–	–	1000
Dropout	–	–	–	–	0.8	–	1000
FC2	–	2	–	–	–	–	2
Softmax	–	–	–	–	–	–	2

(A) Architecture of the 2D slice-level CNN (adaptation of the ResNet-18)

Layer	Filter size	Number of filters / neurons	Stride size	Padding size	Input connection	Output connection
DownConv1	1x1	128	2	0	Conv5	Conv7
DownConv2	1x1	256	2	0	Conv9	Conv11
DownConv3	1x1	512	2	0	Conv13	Conv15

(B) Characteristics of the downsampling layers

B.2 Training hyperparameters

TABLE B.4: Training hyperparameters for autoencoder pretraining experiments.

A summary of the experiments can be found in the first table. The corresponding hyperparameters are listed in the second Table using the same experiments numbers.

Common hyperparameters for all experiments: intensity rescaling: MinMax, optimizer: Adam; Adam parameters: betas=(0.9, 0.999), epsilon=1e-8; loss: mean squared entropy loss; training data: AD + MCI + CN; data split: subject-level. The stopping criterion is the maximal number of epochs.

Experiment number	Classification architectures	Training data	Image preprocessing	Training approach
1	3D subject-level CNN	Baseline	Minimal	single-CNN
2			Extensive	
3	3D ROI-based CNN	Baseline	Minimal	single-CNN
4		Longitudinal		
5	3D patch-level CNN	Baseline	Minimal	single-CNN
6		Longitudinal		
7		Baseline		multi-CNN
8		Longitudinal		

(A) Summary of autoencoder pretraining experiments performed/

Approach	Experiment	Number of epochs	Learning rate	Batch size	Weight decay
3D subject-level CNN	1	50	1e-4	12	0
	2	30	1e-4	12	0
3D ROI-based CNN	3	200	1e-5	32	0
	4	100	1e-5	32	0
3D patch-level CNN	5	20	1e-5	32	0
	6	15	1e-5	32	0
	7	20	1e-5	32	0
	8	15	1e-5	32	0

(B) Hyperparameters corresponding to autoencoder pretraining experiments

TABLE B.5: Training hyperparameters for classification experiments.

A summary of the experiments can be found in the first table. The corresponding hyperparameters are listed in the second table indicated by the experiments numbers.

Common hyperparameters for all experiments: optimizer: Adam; Adam parameters: betas=(0.9, 0.999), epsilon=1e-8; loss: cross entropy. When transfer learning is applied, the corresponding experiment number is given between brackets and can be found in B.4 for AE pretraining (AE) and this table for cross-task transfer learning (CTT).

Experiment number	Classification architectures	Training data	Image preprocessing	Intensity rescaling	Data split	Training approach	Transfer learning	Task
1	3D subject-level CNN	Baseline	Minimal	None	subject-level	single-CNN	None	AD vs CN
2				MinMax			AE (1)	
3								
4		Longitudinal	Extensive				AE (2)	
5							CTT (4)	sMCI vs pMCI
6			Minimal				CTT (3)	
7		Baseline						
8	3D ROI-based CNN	Baseline	Minimal	MinMax	subject-level	single-CNN	AE (3)	AD vs CN
9							CTT (8)	sMCI vs pMCI
10		Longitudinal					AE (4)	AD vs CN
11							CTT (10)	sMCI vs pMCI
12	3D patch-level CNN	Baseline	Minimal	MinMax	subject-level	single-CNN	AE (5)	AD vs CN
13		Longitudinal					AE (6)	
14		Baseline				multi-CNN	AE (7)	AD vs CN
15							CTT (14)	sMCI vs pMCI
16		Longitudinal					AE (8)	AD vs CN
17							CTT (16)	sMCI vs pMCI
18	2D slice-level CNN	Baseline	Minimal	MinMax	subject-level	single-CNN	ImageNet pre-train	AD vs CN
19		Longitudinal						
20		Baseline			slice-level			

(A) Summary of experiments performed

Approach	Experiment	Number of epochs	Learning rate	Batch size	Dropout rate	Weight decay	Patience
3D subject-level CNN	1	50	1e-4	12	0.5	0	10
	2	50	1e-4	12	0.5	0	10
	3	50	1e-4	12	0.5	0	10
	4	50	1e-4	12	0.5	0	5
	5	50	1e-4	12	0.5	0	5
	6	50	1e-5	12	0.5	0	10
	7	50	1e-5	12	0.5	0	20
3D ROI-based CNN	8	200	1e-5	32	0.5	1e-4	10
	9	200	1e-5	32	0.5	1e-3	20
	10	200	1e-5	32	0.5	1e-4	10
	11	200	1e-5	32	0.5	1e-3	20
3D patch-level CNN	12	200	1e-5	32	0.5	1e-3	20
	13	200	1e-5	32	0.5	1e-3	20
	14	200	1e-5	32	0.5	1e-4	15
	15	200	1e-5	32	0.5	1e-3	20
	16	200	1e-5	32	0.5	1e-4	15
	17	200	1e-5	32	0.5	1e-3	20
2D slice-level CNN	18	50	1e-6	32	0.8	1e-4	15
	19	100	1e-6	32	0.8	1e-4	15
	20	50	1e-6	32	0.8	1e-4	15

(B) Hyperparameters corresponding to experiments

B.3 Additional experiments

TABLE B.6: Experiments performed with the single-CNN using thresholding.

MinMax: for CNNs, intensity rescaling was done based on min and max values, resulting all values to be in the range of [0, 1]; AE: autoencoder. The less informative patches (balanced accuracy < 0.7) were not included in the soft voting with the consideration that the labels' probabilities of these patches could harm the majority voting system.

Classification architectures	Training data	Image preprocessing	Intensity rescaling	Data split	Training approach	Transfer learning	Task	Validation balanced accuracy
3D patch-level CNN	Baseline	Minimal	MinMax	subject-level	single-CNN	AE pre-train	AD vs CN	0.79 ± 0.03 [0.81, 0.81, 0.82, 0.80, 0.72]
	Longitudinal							0.83 ± 0.03 [0.86, 0.85, 0.82, 0.84, 0.77]

TABLE B.7: Experiments performed with the multi-CNN without thresholding. MinMax: for CNNs, intensity rescaling was done based on min and max values, resulting all values to be in the range of [0, 1]; AE: autoencoder. sMCI vs pMCI tasks were done with as follows: the weights and biases of the model learnt on the source task (AD vs CN) were transferred to a new model fine-tuned on the target task (sMCI vs pMCI).

All the classifiers composing the multi-CNN were included in the soft voting.

Classification architectures	Training data	Image preprocessing	Intensity rescaling	Data split	Training approach	Transfer learning	Task	Validation balanced accuracy
3D patch-level CNN	Baseline	Minimal	MinMax	subject-level	single-CNN	AE pre-train	AD vs CN	0.76 ± 0.05 [0.74, 0.85, 0.73, 0.77, 0.69]
							sMCI vs pMCI	0.72 ± 0.07 [0.78, 0.65, 0.61, 0.78, 0.76]
	Longitudinal						AD vs CN	0.72 ± 0.04 [0.74, 0.78, 0.72, 0.69, 0.66]
							sMCI vs pMCI	0.70 ± 0.07 [0.73, 0.66, 0.61, 0.81, 0.69]

Appendix C

Description of brain disorders

This appendix aims at shortly presenting the diseases considered by the studies reviewed in Chapter 2.

The majority of the studies focused on the classification of Alzheimer's disease, a neurodegenerative disease of the elderly defined by the presence of three biomarkers: senile plaques formed by amyloid-β protein, neurofibrillary tangles, and atrophy of gray and white matter. The greater affection of the hippocampi can be linked to the memory loss, though other clinical signs may be found. The following diagnosis statuses are often used:

- **AD** refers to demented patients,
- **CN** refers to cognitively normal participants,
- **MCI** refers to patients in prodromal state with mild cognitive impairment,
- **stable MCI** refers to patients in prodromal state who stayed stable during a defined period (often three years),
- **progressive MCI** refers to patients in prodromal state who progressed to Alzheimer's disease after a defined period (often three years).

Most of the studies characterized the disease based on the last biomarker only thanks to T1w MRI, except in (Tang et al., 2019) where the patterns of amyloid-β in the brain are studied.

Fronto-temporal dementia is another neurodegenerative disease of the elderly in which the neuronal loss occurs in the frontal and temporal lobes. Behavior and language are the most affected cognitive functions.

Parkinson's disease is also a neurodegenerative disease of the elderly. It affects dopaminergic neurons in the subtantia nigra. A commonly used neuroimaging technique to detect this loss of dopaminergic neurons is the SPECT, as it uses a ligand that binds to dopamine transporters. Patients are affected by different symptoms linked to motor faculties such as tremor, slowed movements and gait disorder, but also sleep disorder, depression and other symptoms.

Multiple sclerosis is a neurodegenerative disease affecting younger people (it begins between the ages of 20 and 50). It causes demyelination of the white matter in the brain

(brain stem, basal ganglia, tracts near the ventricles), optic nerve and spinal cord. This demyelination results in autonomic, visual, motor and sensory problems.

Intracranial hemorrhage may result from a physical trauma or nontraumatic causes such as a ruptured aneurysm. Different subtypes exist depending on the location of the hemorrhage. Most of the time, it results in the death of the patient.

Autism is a neurodevelopmental disorder affecting social interation and communication. Diagnosis is done based on clinical signs (behavior) and the patterns that may exist in the brain are not reliably described as they overlap with the neurotypical population.

Some brain characteristics which may be related to brain disorders and detected in CT scans were considered in the data set CQ500:

- **Midline Shift** is a shift of the center of the brain past the center of the skull.
- **Mass Effect** is caused by the presence of an intracranial lesion (for example a tumor) which is compressing nearby tissues.
- **Calvarial Fractures** are fractures of the skull.

Finally, one study (Ball et al., 2021) learned to predict the age of cognitively normal patients. Such algorithm can provide a biomarker of brain disorders as patients will have a greater brain age than their chronological age, then it is a general biomarker to establish that a participant is not in the normal distribution.

Appendix D

Supplementary results of Chapter 5

As seen in the results of Chapter 5, the pattern of the typical subtype is always better captured than the atypical one ($b > b^{\dagger}$). Two reasons were explored to explain this difference between subtypes: the proportion of atypical subtypes and the shape of the regions.

D.1 Proportion of subtypes

We explored different proportions of subtype 2 & 3 in the *Atrophied* label. In the following experiments, there are no errors in the *Atrophied* label, and 500 samples were generated per label for each data set.

We compare the images (Figure D.1) and b and $b^{\dagger}$ criteria (Table D.1) of the maps obtained with different proportion of the atypical subtypes (15%, 50% and 85%) and concluded that the proportion of the subtypes could not explain the difference between b and $b^{\dagger}$.

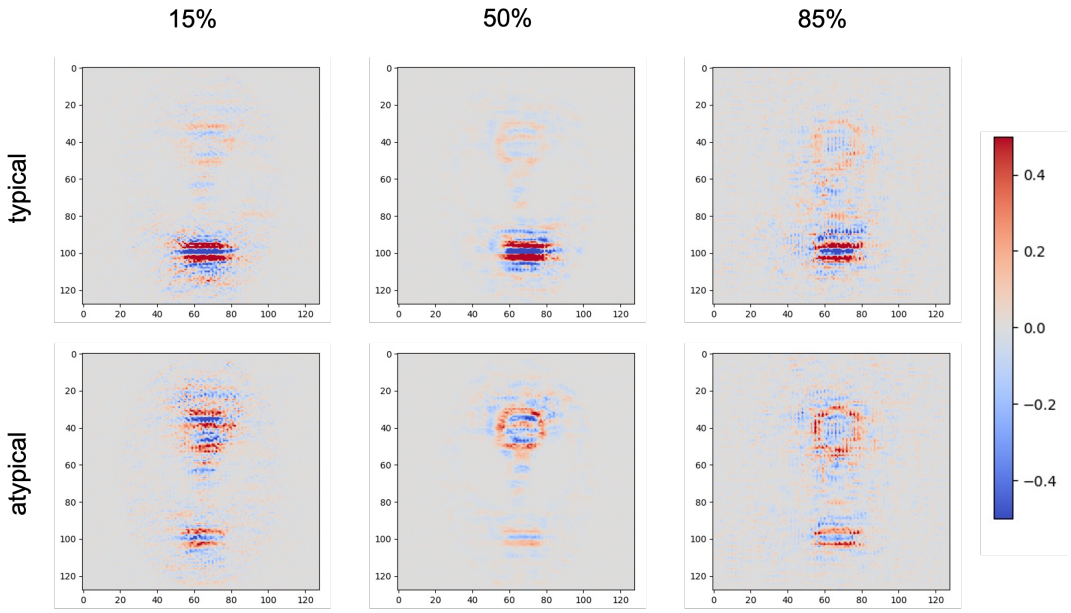


FIGURE D.1: She saliency maps obtained obtained with different proportion of the atypical subtype (first fold).

Atypical proportion	b	$b^\dagger$
15%	5.72	1.39
50%	6.83	1.63
85%	3.33	1.42

TABLE D.1: Values of b and $b^\dagger$ criteria obtained depending on the proportion of the atypical subtype.

D.2 Shape of subtypes

In this experiment a symmetric version of the synthetic data was generated (see Figure D.2). As in the previous version, the regions Top and Bottom allow the identification of three subtypes: subtype 1 has two large regions, subtype 2 has a large Top region and a small Bottom region, subtype 3 has a small Top region and a large Bottom region. We refer to the previous version of the data set as the asymmetric data set. In this setup, both data sets were generated with an *Atrophied* label including 50% of typical subtype, 50% of atypical subtype and no errors. 500 samples were generated per label for each data set.

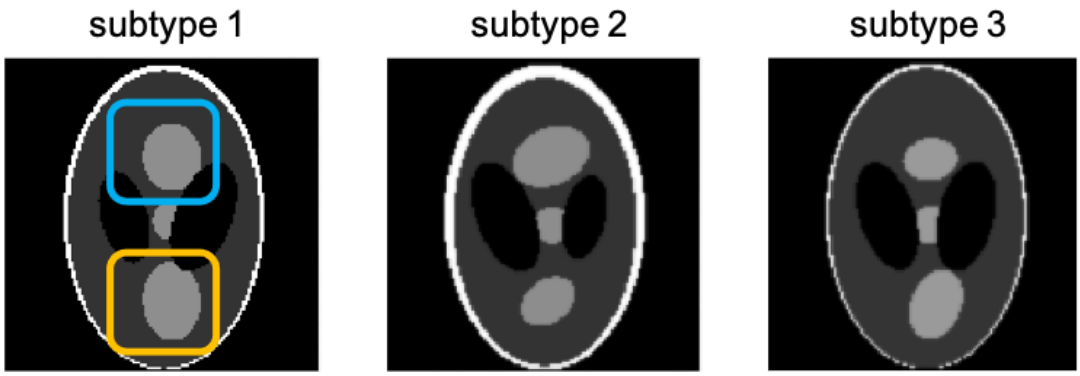


FIGURE D.2: Display of the three synthetic symmetric subtypes. The regions Top and Bottom are highlighted by blue and orange frames, respectively.

We compare the images (Figure D.3) and b and $b^\dagger$ criteria (Table D.2) of the maps obtained with the symmetric and asymmetric data sets. This time we observe that the difference between b and $b^\dagger$ differs between the two versions of the dataset. We thus concluded that the shape of the regions influences the ability of the saliency maps to highlight the correct region of interest.

Dataset	b	b†
asymmetric	6.83	1.62
symmetric	1.50	1.78

TABLE D.2: Values of b and $b^\dagger$ criteria obtained on symmetric and asymmetric data sets.

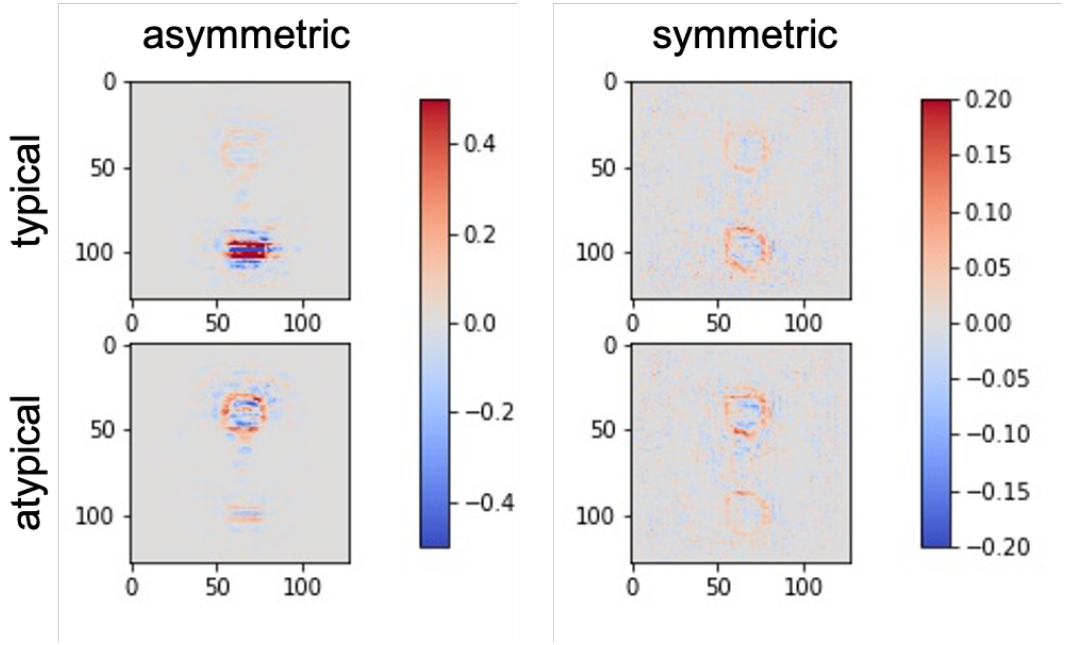


FIGURE D.3: Saliency maps obtained on symmetric and asymmetric data sets (first fold).

Appendix E

Adversarial framework to prevent the occurrence of artifacts when interpreting CNNs: Application to Alzheimer's disease classification

E.1 Introduction

In this appendix, we tried to improve the interpretability method used in chapter 4 to avoid the creation of artifacts in attribution maps. The interpretability method consists in optimizing a mask that learns to perturb a trained convolutional neural network (CNN) so it will classify perturbed data samples in the wrong class, hence it can be seen as being a mix of perturbation and gradient visualization methods. The main limitation of this method is the possible occurrence of artifacts in the mask training process. The masks comprising artifacts are not perturbing images in the regions that are relevant for classification but create new patterns to disturb the CNN without focusing on these regions.

We used as inputs gray matter probability maps, a proxy for atrophy, extracted from T1-weighted (T1w) magnetic resonance (MR) images. The process includes two distinct parts: first a CNN is trained to classify patients with Alzheimer's disease (AD) from control subjects, then the weights of the network are fixed and a mask is trained to prevent the network from classifying correctly all the subjects it has correctly classified after training. The goal of this work was to limit the presence of artifacts and more generally to improve the quality of the attribution maps obtained by introducing a new penalty in the mask loss based on the reconstruction error of an autoencoder that has learned the brain T1w MR image distribution.

E.2 Materials and Methods

E.2.1 Data description and preprocessing

Data used in the preparation of this chapter were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database. The ADNI data set used in our experiments

TABLE E.1: Summary of participant demographics, mini-mental state examination (MMSE) and global clinical dementia rating (CDR) scores at baseline. Values are presented as mean (SD).

	Subjects	Sessions	Age	Gender	MMSE	CDR
CN	330	1 830	74.4 (5.8)	160 M / 170 F	29.1 (1.1)	0:330
AD	336	1 106	75.0 (7.8)	185 M / 151 F	23.2 (2.1)	0.5: 160, 1: 175, 2:1

comprises 1455 participants for whom a T1w MRI was available for at least one visit. Two diagnosis groups were considered:

- CN: sessions of subjects who were cognitively normal at baseline and stayed stable during the follow-up;
- AD: sessions of subjects who were diagnosed as AD at baseline and stayed stable during the follow-up.

The population is described in Table E.1.

Preprocessing of T1w MR images was performed with the Clinica software platform (`www.clinica.run`) (Routier et al., 2021). First the ADNI data set was converted to the BIDS format, then the t1-volume preprocessing pipeline of Clinica was applied (Samper-González et al., 2018). This pipeline performs bias field correction, non-linear registration and tissue segmentation using the Unified Segmentation approach (Ashburner and Friston, 2005) available in SPM12. The gray matter maps in MNI space were retrieved for the image analysis.

E.2.2 CNN classification

The following sections describe the evaluation procedure, the hyperparameters selection and implementation details that are linked to the classification of AD vs CN subjects with CNNs. During training, the weights and biases of the network are optimized to maximize the score function f on a set of images X.

Evaluation procedure

The ADNI data set was split into training/validation, external validation and test sets. The ADNI external validation and test set consisted each of 75 randomly chosen age- and sex-matched subjects for each diagnostic class (i.e. 75 CN subjects, 75 AD patients). The rest of the ADNI data set was used as training/validation set. We ensured that age and sex distributions between training/validation, external validation and test sets were not significantly different. The model selection procedure, including model architecture selection and training hyperparameter fine-tuning, was performed using only the training/validation data set. For that purpose, a 5-fold cross-validation (CV) was performed, which resulted in one fold (20%) of the data for validation and the rest for

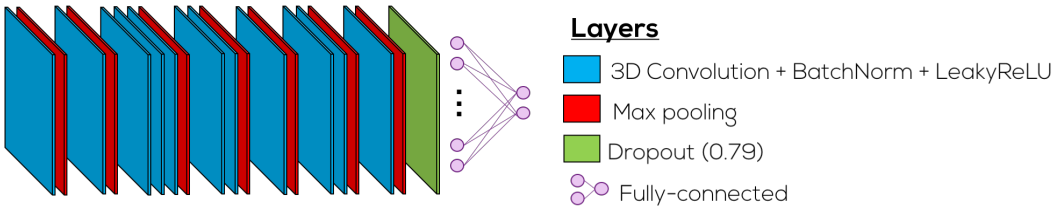


FIGURE E.1: Architecture of the CNN classifier determined with a random search procedure.

training. Note that the 5-fold data split was performed only once for all the experiments with a fixed seed number ($random\_state = 2$), thus guaranteeing that all the experiments used exactly the same subjects during CV.

Hyperparameter selection

We performed a random search (Bergstra and Bengio, 2012) to select the architecture and optimization hyperparameters of our CNN. The hyperparameters explored for the architecture were the number of convolutional blocks, of filters in the first layer and of convolutional layers in a block, the dimension reduction strategy, the number of fully-connected layers and the dropout rate. Other hyperparameters such as the learning rate, the weight decay, the batch size, the data preprocessing and the intensity normalization were also part of the search.

The chosen architecture was the one that obtained the best mean balanced accuracy on the 5-fold CV. This architecture (displayed in Fig. E.1) is composed of seven convolutional blocks followed by a dropout layer and a fully-connected layer. Each convolutional block is made of one to three sub-blocks and a max pooling layer with a kernel size and a stride of 2. Each sub-block is composed of a convolutional layer with kernel size of 3 and padding of 1, a batch-normalization layer and a LeakyReLU activation. The label of the input image is the index of the output node having the maximum activation.

CNN training

The weights of the convolutional and fully connected layers were initialized as described in (He et al., 2015), which corresponds to the default initialization method in PyTorch. Weights were updated based on the cross-entropy loss. We applied the following early stopping strategy for all the classification experiments: the training procedure does not stop until the validation loss is continuously higher than the lowest validation loss for N epochs (N=5). Otherwise, the training continues to the end of a pre-defined number of epochs (30). The final model was the one that obtained the highest validation balanced accuracy during training. The accuracy of the model is evaluated at the end of each epoch.

E.2.3 Visualization method

The proposed visualization method extends the framework of Fong and Vedaldi, 2017. Once the classification network has been trained, its parameters are fixed to the best value

found, denoted as w^*. The method consists in computing a mask that will overlay the most meaningful parts of an image to prevent the network from classifying it correctly. In the following, the goal is to mask AD images that were correctly classified by the CNN so that it systematically classifies them with the CN label. The mask m is a 3D volume of the same size as the input image and hides parts of the image in a voxel-wise manner. In this application, each voxel u of the input image X will be masked by a constant value μ according to the value of the mask for this voxel. The mask values are included in $[0,1]$. The masked input image X'_m at voxel u is defined as:

$$X^m(u) = m(u)X(u) + (1 - m(u))\mu \tag{E.1}$$

As AD patients suffer from gray matter atrophy, the goal of the masking method would be to artificially simulate gray matter restoration in a minimal number of brain regions to make them look like CN subjects. By setting $\mu = 1$, the mask was trained to artificially increase the probability of gray matter for the minimum set of voxels that would lead to the maximum decrease of the performance of the CNN. The optimal mask m^* is the mask for which the following loss function is minimized:

$$m^* = \underset{m}{\operatorname{argmin}}\, f(X^m) + R(m) \quad . \tag{E.2}$$

$R(m)$ regularizes the mask to make it focus only on relevant regions of the image. In the original paper (Fong and Vedaldi, 2017), it was composed of two terms: the first one ensures that a minimum set of voxels is selected while the second allows voxels to be different from 1 only if there are enough voxels in their neighbourhood that are also relevant for classification

$$R(m) = \lambda_1 \|1 - m\|_{\beta_1}^{\beta_1} + \lambda_2 \sum_u \|\nabla m(u)\|_{\beta_2}^{\beta_2} \quad . \tag{E.3}$$

As in chapter 4, once the mask training is finished, values above 0.95 are set to 1. This ensures that the CNN is only perturbed by the zones identified by the mask, and not by the small gradients that can be found on all the surface of the mask.

Artifact occurrence

As already discussed in (Fong and Vedaldi, 2017), this regularization does not prevent from the occurrence of artifacts. Instead of focusing on the relevant regions, the mask will innovate and create new patterns that will disturb the CNN as they may not exist in the original data distribution. Parts of the background may be highlighted, with patterns that are not in the brain MR image distribution (rectangles of gray matter in and outside the brain).

Adversarial regularization term

To avoid this phenomenon, the second term of the regularization has been replaced by an adversarial term. This latter is based on the reconstruction loss of a trained denoising

autoencoder *AE* that ensures that the masked images still remain in the brain MR image distribution

$$R(X,m) = \lambda_1 \|1-m\|_{\beta_1}^{\beta_1} + \lambda_{AE} \left\| AE(X^m) - X^m \right\|_2^2 \quad . \tag{E.4}$$

The architecture of the autoencoder was chosen based on that of the CNN classifier. The encoder part is made of all the convolutional blocks of the CNN before the dropout layer. The decoder part is then the transposed version of the encoder, with convolutions and max pooling being replaced by transposed convolutions and max unpooling respectively. The autoencoder training is similar to the CNN training, except that the weights are updated based on the mean squared error loss and the best model is chosen based on the best validation loss. Moreover, the autoencoder learns to remove noise from images, then Gaussian noise of standard deviation 0.1 or 0.5 may be randomly added to training images.

Metrics of evaluation

The densities of masks according to the 120 regions-of-interest (ROIs) of the AAL2 atlas (Rolls et al., 2015) was computed to assess the coherence with previous knowledge of the disease. For each ROI, 1 minus the sum of the values in the ROI is computed, resulting in a ROI-vector of size 120 for each mask. Each value in the ROI-vector represents the density of the mask in the associated ROI. We expect an non-artifacted mask to have a high density in the hippocampi and none in the background.

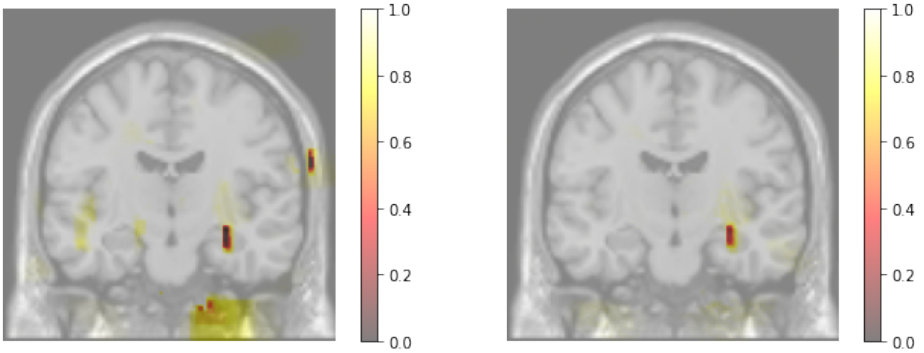
E.3 Results

Once the architecture was chosen, the CNN was trained ten times on the whole training/validation set and the external validation set was used as the validation set. We selected among these runs the only run that produced artifacts when optimizing the perturbation mask. On this run, the validation balanced accuracy was 0.84, and the test balanced accuracy 0.87.

E.3.1 Baseline method

A group mask for AD patients with the same parameters as in chapter 4 was computed ($\lambda_1 = 0.0001$, $\lambda_2 = 0.01$, $\beta_1 = 0.1$, $\beta_2 = 1$). As artifacts were found in this mask, as shown in Figure E.2, the values of the regularization terms were increased to the values used for individual mask computations ($\lambda_1 = 0.01$, $\lambda_2 = 1$, $\beta_1 = 0.1$, $\beta_2 = 1$). With these new values, masked regions outside of the brain are less intense.

Though increasing the values of the regularization allows removing some artifacts, there are still regions highlighted outside of the brain. Moreover the patterns are unnatural (rectangles of gray matter in the hippocampus). This is why we tried to improve this result by using the adversarial regularization with an autoencoder loss.



(A) Mask obtained with light regularization ($\lambda_1 = 0.0001$, $\lambda_2 = 0.01$)

(B) Mask obtained with high regularization ($\lambda_1 = 0.01$, $\lambda_2 = 1$)

FIGURE E.2: Group masks on AD patients optimized from a CNN causing artifacts

E.3.2 Adversarial regularization

The denoising autoencoder successfully learned to remove noise from images (best mean validation loss 0.0017). We performed a grid search on the new set of hyperparameters λ_1 and λ_{AE} to find a couple of values that would allow removing the artifacts in the optimized mask. As shown in Figure E.3, we did not succeed in preventing the formation of artifacts during the optimization of the mask with any values. Even worse, increasing too much the value of λ_{AE} strongly deteriorates the mask (see image obtained from $\lambda_1 = 0.01$ and $\lambda_{AE} = 100$).

E.4 Conclusion

We did not solve the creation of artifacts in mask optimization with our adversarial optimization. We guess that during its optimization, the mask learns to fool the autoencoder if the regularization term λ_{AE} is too high (leading to a new type of artifacts). A way to prevent this could be to train the autoencoder at the same time as the mask to re-enforce the autoencoder and avoid this failure mode. However, as such, adversarial frameworks are difficult to train, this option was thus abandoned.

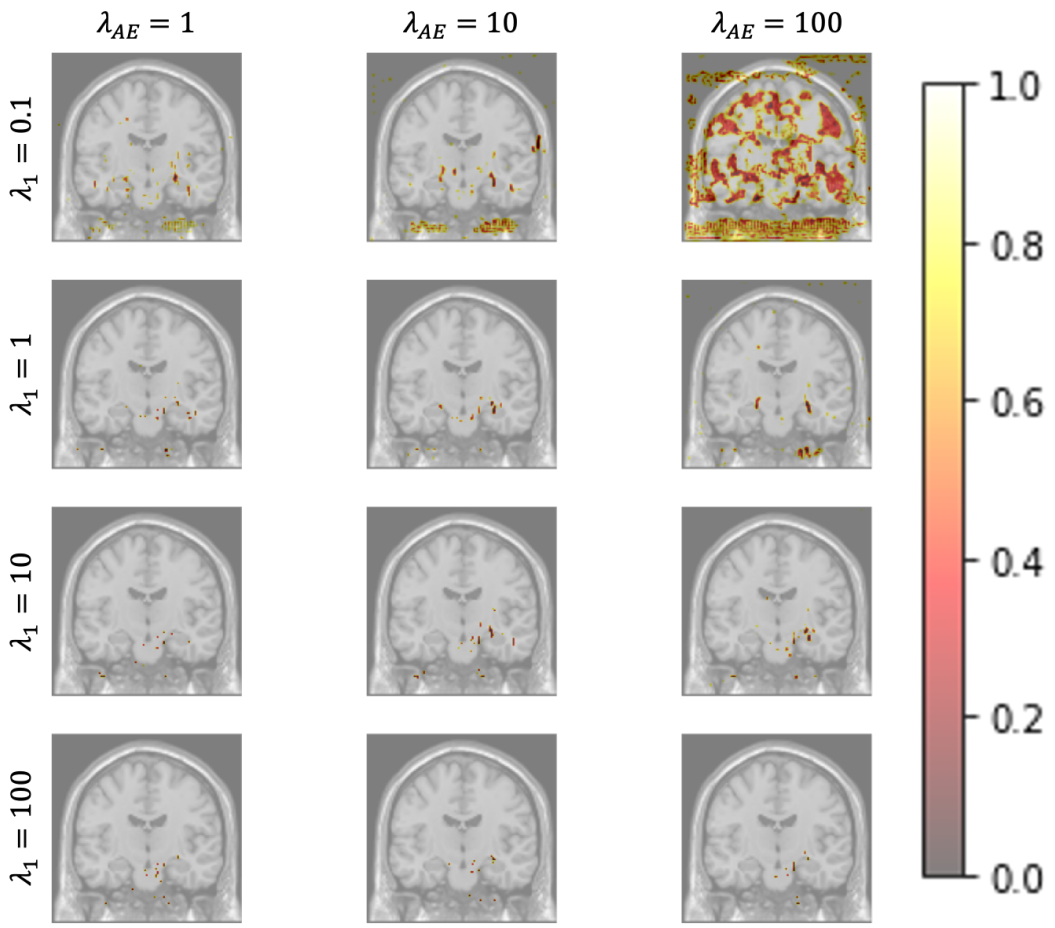


FIGURE E.3: Grid search on λ_1 and λ_{AE} hyperparameters.

Appendix F

Computing resources & Data access

F.1 Computing resources

This work was granted access to the HPC resources of IDRIS under the allocation 2019-100963 made by GENCI (Grand Équipement National de Calcul Intensif) in the context of the Jean Zay "Grands Challenges" (2019). Some experiments were also performed on the cluster of the Paris Brain Institute, which is equipped with 4 NVIDIA P100 GPU cards (64 GB shared memory) and 24 CPUs (120 GB shared memory).

F.2 Data access

Data collection and sharing for this project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: AbbVie, Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Araclon Biotech; BioClinica, Inc.; Biogen; Bristol-Myers Squibb Company; CereSpir, Inc.; Cogstate; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and Company; EuroImmun; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; Fujirebio; GE Healthcare; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Lumosity; Lundbeck; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Neurotrack Technologies; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Takeda Pharmaceutical Company; and Transition Therapeutics. The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (www.fnih.org). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Therapeutic Research Institute at the University of Southern California. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of Southern California.

Data used in the preparation of this thesis was obtained from the Australian Imaging Biomarkers and Lifestyle flagship study of ageing (AIBL) funded by the Commonwealth

Scientific and Industrial Research Organisation (CSIRO) which was made available at the ADNI database (www.loni.usc.edu/ADNI). The AIBL researchers contributed data but did not participate in analysis or writing of this report. AIBL researchers are listed at www.aibl.csiro.au.

The OASIS Cross-Sectional project (Principal Investigators: D. Marcus, R, Buckner, J, Csernansky J. Morris) was supported by the following grants: P50 AG05681, P01 AG03991, P01 AG026276, R01 AG021910, P20 MH071616, and U24 RR021382.

Data used in preparation of this thesis was obtained from the Frontotemporal Lobar Degeneration Neuroimaging Initiative (FTLDNI) database (http://4rtni-ftldni.ini.usc.edu/). The investigators at NIFD/FTLDNI contributed to the design and implementation of FTLDNI and/or provided data, but did not participate in analysis or writing of this report (unless otherwise listed).

Bibliography

Abrol, A. et al. (2020). "Deep Residual Learning for Neuroimaging: An Application to Predict Progression to Alzheimer's Disease". In: *Journal of Neuroscience Methods* 339, p. 108701. DOI: 10.1016/j.jneumeth.2020.108701.

Adebayo, J. et al. (2018). "Sanity Checks for Saliency Maps". In: *Advances in Neural Information Processing Systems*, pp. 9505–9515.

Aderghal, K et al. (2018). "Classification of Alzheimer Disease on Imaging Modalities with Deep CNNs Using Cross-Modal Transfer Learning". In: *IEEE 31st International Symposium on Computer-Based Medical Systems (CBMS)*, pp. 345–350.

Aderghal, K. et al. (2017a). "Classification of sMRI for AD Diagnosis with Convolutional Neuronal Networks: A Pilot 2-D+ϵ Study on ADNI". In: *International Conference on Multimedia Modeling*, pp. 690–701.

Aderghal, K. et al. (2017b). "FuseMe: Classification of sMRI images by fusion of Deep CNNs in 2D+ϵ projections". In: *15th International Workshop on Content-Based Multimedia Indexing*. ACM, p. 34.

Aisen, P. S. et al. (2010). "Clinical Core of the Alzheimer's Disease Neuroimaging Initiative: Progress and Plans". In: *Alzheimer's & Dementia: The Journal of the Alzheimer's Association* 6.3, pp. 239–246. DOI: 10.1016/j.jalz.2010.03.006.

Alfaro-Almagro, F. et al. (2021). "Confound Modelling in UK Biobank Brain Imaging". In: *NeuroImage* 224, p. 117002. DOI: 10.1016/j.neuroimage.2020.117002.

Amoroso, N. et al. (2018). "Deep learning reveals Alzheimer's disease onset in MCI subjects: Results from an international challenge". In: *J. Neurosci. Methods* 302, pp. 3–9.

Andersson, J. L., M. Jenkinson, and S. Smith (2007). "Non-Linear Registration, Aka Spatial Normalisation FMRIB Technical Report TR07JA2". In: *FMRIB Analysis Group of the University of Oxford* 2.1, e21.

Ansart, M. et al. (2021). "Predicting the Progression of Mild Cognitive Impairment Using Machine Learning: A Systematic, Quantitative and Critical Review". In: *Medical Image Analysis* 67, p. 101848. DOI: 10.1016/j.media.2020.101848.

Antonelli, M. et al. (2021). "The Medical Segmentation Decathlon". In: *arXiv:2106.05735 [cs, eess]*.

Ashburner, J. (2007). "A fast diffeomorphic image registration algorithm". In: *Neuroimage* 38.1, pp. 95–113.

Ashburner, J. and K. J. Friston (2000). "Voxel-Based Morphometry—The Methods". In: *NeuroImage* 11.6, pp. 805–821. DOI: 10.1006/nimg.2000.0582.

— (2005). "Unified Segmentation". In: *NeuroImage* 26.3, pp. 839–851. DOI: 10.1016/j.neuroimage.2005.02.018.

Au, R., R. J. Piers, and L. Lancashire (2015). "Back to the Future: Alzheimer's Disease Heterogeneity Revisited". In: *Alzheimer's & Dementia : Diagnosis, Assessment & Disease Monitoring* 1.3, pp. 368–370. DOI: 10.1016/j.dadm.2015.05.006.

Avants, B. B. et al. (2008). "Symmetric Diffeomorphic Image Registration with Cross-Correlation: Evaluating Automated Labeling of Elderly and Neurodegenerative Brain". In: *Medical image analysis* 12.1, pp. 26–41. DOI: 10.1016/j.media.2007.06.004.

Avants, B. B. et al. (2014). "The Insight ToolKit Image Registration Framework". In: *Frontiers in Neuroinformatics* 8. DOI: 10.3389/fninf.2014.00044.

Ba, J., V. Mnih, and K. Kavukcuoglu (2015). "Multiple Object Recognition with Visual Attention". In: *ICLR (Poster)*. URL: http://arxiv.org/abs/1412.7755.

Bach, S. et al. (2015). "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation". In: *PLOS ONE* 10.7, e0130140. DOI: 10.1371/journal.pone.0130140.

Bäckström, K. et al. (2018). "An Efficient 3D Deep Convolutional Network for Alzheimer's Disease Diagnosis Using MR Images". In: *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pp. 149–153. DOI: 10.1109/ISBI.2018.8363543.

Bae, J. et al. (2019). "Transfer Learning for Predicting Conversion from Mild Cognitive Impairment to Dementia of Alzheimer's Type Based on 3D-Convolutional Neural Network". In: *bioRxiv*. DOI: 10.1101/2019.12.20.884932.

Baker, M. (2016). "1,500 Scientists Lift the Lid on Reproducibility". In: *Nature News* 533.7604, p. 452. DOI: 10.1038/533452a.

Ball, G. et al. (2021). "Individual Variation Underlying Brain Age Estimates in Typical Development". In: *NeuroImage* 235, p. 118036. DOI: 10.1016/j.neuroimage.2021.118036.

Basaia, S. et al. (2019). "Automated classification of Alzheimer's disease and mild cognitive impairment using a single MRI and deep neural networks". In: *Neuroimage Clin* 21, p. 101645.

Baskar, D, V. S. Jayanthi, and A. N. Jayanthi (2018). "An efficient classification approach for detection of Alzheimer's disease from biomedical imaging modalities". In: *Multimed. Tools Appl.*, pp. 1–33.

Beam, A. L., A. K. Manrai, and M. Ghassemi (2020). "Challenges to the Reproducibility of Machine Learning Models in Health Care". In: *JAMA* 323.4, pp. 305–306. DOI: 10.1001/jama.2019.20866.

Bergstra, J. and Y. Bengio (2012). "Random Search for Hyper-Parameter Optimization". In: *Journal of Machine Learning Research* 13.Feb, pp. 281–305.

Bernal, J. et al. (2018). "Deep convolutional neural networks for brain image analysis on magnetic resonance imaging: a review". In: *Artif. Intell. Med.*

Bhagwat, N. et al. (2018). "Modeling and prediction of clinical symptom trajectories in Alzheimer's disease using longitudinal data". In: *PLoS Comput. Biol.* 14.9, e1006376.

Biffi, C. et al. (2020). "Explainable Anatomical Shape Analysis through Deep Hierarchical Generative Models". In: *IEEE Transactions on Medical Imaging* 39.6, pp. 2088–2099. DOI: 10.1109/TMI.2020.2964499.

Burduja, M., R. T. Ionescu, and N. Verga (2020). "Accurate and Efficient Intracranial Hemorrhage Detection and Subtype Classification in 3D CT Scans with Convolutional and Long Short-Term Memory Neural Networks". In: *Sensors* 20.19, p. 5611. DOI: 10.3390/s20195611.

Burgos, N. et al. (2020). "Deep Learning for Brain Disorders: From Data Processing to Disease Treatment". In: *Briefings in Bioinformatics* bbaa310. DOI: 10.1093/bib/bbaa310.

Bussola, N. et al. (2021). "AI Slipping on Tiles: Data Leakage in Digital Pathology". In: *Pattern Recognition. ICPR International Workshops and Challenges*. Ed. by A. Del Bimbo et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 167–182. DOI: 10.1007/978-3-030-68763-2_13.

Böhle, M. et al. (2019). "Layer-Wise Relevance Propagation for Explaining Deep Neural Network Decisions in MRI-Based Alzheimer's Disease Classification". In: *Frontiers in Aging Neuroscience* 10.JUL. DOI: 10.3389/fnagi.2019.00194.

Cárdenas-Peña, D., D. Collazos-Huertas, and G. Castellanos-Dominguez (2016). "Centered Kernel Alignment Enhancing Neural Network Pretraining for MRI-Based Dementia Diagnosis". In: *Comput. Math. Methods Med.* 2016, p. 9523849.

— (2017). "Enhanced Data Representation by Kernel Metric Learning for Dementia Diagnosis". In: *Front. Neurosci.* 11, p. 413.

Cecotti, H. and A. Gräser (2011). "Convolutional Neural Networks for P300 Detection with Application to Brain-Computer Interfaces". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.3, pp. 433–445. DOI: 10.1109/TPAMI.2010.125.

Chaddad, A., C. Desrosiers, and T. Niazi (2018). "Deep Radiomic Analysis of MRI Related to Alzheimer's Disease". In: *IEEE Access* 6, pp. 58213–58221. DOI: 10.1109/ACCESS.2018.2871977.

Chadebec, C. et al. (2021). "Data Augmentation in High Dimensional Low Sample Size Setting Using a Geometry-Based Variational Autoencoder".

Cheng, D and M Liu (2017). "CNNs based multi-modality classification for AD diagnosis". In: *10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 1–5.

Cheng, D. et al. (2017). "Classification of MR brain images by combination of multi-CNNs for AD diagnosis". In: *Ninth International Conference on Digital Image Processing (ICDIP)*. Vol. 10420. International Society for Optics and Photonics, p. 1042042.

Çitak-ER, F., D. Goularas, and B. Ormeci (2017). "A novel Convolutional Neural Network Model Based on Voxel-based Morphometry of Imaging Data in Predicting the Prognosis of Patients with Mild Cognitive Impairment". In: *J. Neurol. Sci. Turk.* 34.1.

Collberg, C. and T. A. Proebsting (2016). "Repeatability in Computer Systems Research". In: *Communications of the ACM* 59.3, pp. 62–69. DOI: 10.1145/2812803.

Couvy-Duchesne, B. et al. (2020). "Ensemble Learning of Convolutional Neural Network, Support Vector Machine, and Best Linear Unbiased Predictor for Brain Age Prediction: ARAMIS Contribution to the Predictive Analytics Competition 2019 Challenge". In: *Frontiers in Psychiatry* 11. DOI: 10.3389/fpsyt.2020.593336.

Crane, M. (2018). "Questionable Answers in Question Answering Research: Reproducibility and Variability of Published Results". In: *Transactions of the Association for Computational Linguistics* 6, pp. 241–252. DOI: 10.1162/tacl_a_00018.

Cui, R, M Liu, and G Li (2018). "Longitudinal analysis for Alzheimer's disease diagnosis using RNN". In: *IEEE 15th International Symposium on Biomedical Imaging (ISBI)*, pp. 1398–1401.

Cuingnet, R. et al. (2011). "Automatic Classification of Patients with Alzheimer's Disease from Structural MRI: A Comparison of Ten Methods Using the ADNI Database". In: *NeuroImage*. Multivariate Decoding and Brain Reading 56.2, pp. 766–781. DOI: 10.1016/j.neuroimage.2010.06.013.

DeGrave, A. J., J. D. Janizek, and S.-I. Lee (2021). "AI for Radiographic COVID-19 Detection Selects Shortcuts over Signal". In: *Nature Machine Intelligence* 3.7, pp. 610–619. DOI: 10.1038/s42256-021-00338-7.

Deng, J. et al. (2009). "ImageNet: A Large-Scale Hierarchical Image Database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

Dickerson, B. C. et al. (2001). "MRI-derived entorhinal and hippocampal atrophy in incipient and very mild Alzheimer's disease". In: *Neurobiol. Aging* 22.5, pp. 747–754.

Dolph, C. V. et al. (2017). "Deep learning of texture and structural features for multiclass Alzheimer's disease classification". In: *International Joint Conference on Neural Networks (IJCNN)*, pp. 2259–2266.

Dong, A. et al. (2016a). "CHIMERA: Clustering of Heterogeneous Disease Effects via Distribution Matching of Imaging Patterns". In: *IEEE transactions on medical imaging* 35.2, pp. 612–621. DOI: 10.1109/TMI.2015.2487423.

Dong, A. et al. (2016b). "Heterogeneity of Neuroanatomical Patterns in Prodromal Alzheimer's Disease: Links to Cognition, Progression and Biomarkers". In: *Brain*, aww319. DOI: 10.1093/brain/aww319.

Duraisamy, B., J. V. Shanmugam, and J. Annamalai (2019). "Alzheimer disease detection from structural MR images using FCM based weighted probabilistic neural network". In: *Brain Imaging Behav.* 13.1, pp. 87–110.

Dyrba, M., A. H. Pallath, and E. N. Marzban (2020). "Comparison of CNN Visualization Methods to Aid Model Interpretability for Detecting Alzheimer's Disease". In: *Bildverarbeitung für die Medizin 2020*. Ed. by T. Tolxdorff et al. Informatik aktuell. Wiesbaden: Springer Fachmedien, pp. 307–312. DOI: 10.1007/978-3-658-29267-6_68.

Eitel, F. and K. Ritter (2019). "Testing the Robustness of Attribution Methods for Convolutional Neural Networks in MRI-Based Alzheimer's Disease Classification". In: *Interpretability of Machine Intelligence in Medical Image Computing and Multimodal Learning for Clinical Decision Support*. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 3–11. DOI: 10.1007/978-3-030-33850-3_1.

Eitel, F. et al. (2019). "Uncovering Convolutional Neural Network Decisions for Diagnosing Multiple Sclerosis on Conventional MRI Using Layer-Wise Relevance Propagation". In: *NeuroImage: Clinical* 24, p. 102003. DOI: 10.1016/j.nicl.2019.102003.

El-Sappagh, S. et al. (2020). "Multimodal Multitask Deep Learning Model for Alzheimer's Disease Progression Detection Based on Time Series Data". In: *Neurocomputing* 412, pp. 197–215. DOI: 10.1016/j.neucom.2020.05.087.

Ellis, K. A. et al. (2009). "The Australian Imaging, Biomarkers and Lifestyle (AIBL) Study of Aging: Methodology and Baseline Characteristics of 1112 Individuals Recruited for a Longitudinal Study of Alzheimer's Disease". In: *International Psychogeriatrics* 21.4, pp. 672–687. DOI: 10.1017/S1041610209009405.

Ellis, K. A. et al. (2010). "Addressing Population Aging and Alzheimer's Disease through the Australian Imaging Biomarkers and Lifestyle Study: Collaboration with the Alzheimer's Disease Neuroimaging Initiative". In: *Alzheimer's & Dementia: The Journal of the Alzheimer's Association* 6.3, pp. 291–296. DOI: 10.1016/j.jalz.2010.03.009.

Esmaeilzadeh, S. et al. (2018). "End-To-End Alzheimer's Disease Diagnosis and Biomarker Identification: 9th International Workshop, MLMI 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Proceedings". In: *Machine Learning in Medical Imaging*. Ed. by Y. Shi, H.-I. Suk, and M. Liu. Vol. 11046. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 337–345.

Falahati, F., E. Westman, and A. Simmons (2014). "Multivariate data analysis and machine learning in Alzheimer's disease with a focus on structural magnetic resonance imaging". In: *J. Alzheimers. Dis.* 41.3, pp. 685–708.

Farooq, A et al. (2017). "A deep CNN based multi-class classification of Alzheimer's disease using MRI". In: *2017 IEEE International Conference on Imaging Systems and Techniques (IST)*, pp. 1–6.

Fong, R. C. and A. Vedaldi (2017). "Interpretable Explanations of Black Boxes by Meaningful Perturbation". In: *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 3449–3457. DOI: 10.1109/ICCV.2017.371.

Fonov, V. S. et al. (2009). "Unbiased nonlinear average age-appropriate brain templates from birth to adulthood". In: *Neuroimage* Supplement 1.47, S102.

Fonov, V. et al. (2011). "Unbiased average age-appropriate atlases for pediatric studies". In: *Neuroimage* 54.1, pp. 313–327.

Fonov, V. et al. (2018). "Deep learning of quality control for stereotaxic registration of human brain MRI".

Friston, K. J. et al. (1995). "Analysis of fMRI Time-Series Revisited". In: *NeuroImage* 2.1, pp. 45–53. DOI: 10.1006/nimg.1995.1007.

Fu, G. et al. (2021). "Attention-Based Full Slice Brain CT Image Diagnosis with Explanations". In: *Neurocomputing* 452, pp. 263–274. DOI: 10.1016/j.neucom.2021.04.044.

Gao, F. et al. (2020). "AD-NET: Age-Adjust Neural Network for Improved MCI to AD Conversion Prediction". In: *NeuroImage: Clinical* 27, p. 102290. DOI: 10.1016/j.nicl.2020.102290.

Gibson, E. et al. (2018). "NiftyNet: a deep-learning platform for medical imaging". In: *Computer Methods and Programs in Biomedicine* 158, pp. 113–122. ISSN: 0169-2607. DOI: 10.1016/j.cmpb.2018.01.025.

Goodman, S. N., D. Fanelli, and J. P. A. Ioannidis (2016). "What Does Research Reproducibility Mean?" In: *Science Translational Medicine* 8.341, 341ps12–341ps12. DOI: 10.1126/scitranslmed.aaf5027.

Gorgolewski, K. et al. (2011). "Nipype: a flexible, lightweight and extensible neuroimaging data processing framework in python". In: *Front. Neuroinform.* 5, p. 13.

Gorgolewski, K. J. et al. (2016). "The Brain Imaging Data Structure, a Format for Organizing and Describing Outputs of Neuroimaging Experiments". In: *Scientific Data* 3, p. 160044. DOI: 10.1038/sdata.2016.44.

Gorgolewski, K. J. et al. (2017). "BIDS Apps: Improving Ease of Use, Accessibility, and Reproducibility of Neuroimaging Data Analysis Methods". In: *PLOS Computational Biology* 13.3, e1005209. DOI: 10.1371/journal.pcbi.1005209.

Gorji, H. T. and J Haddadnia (2015). "A novel method for early diagnosis of Alzheimer's disease based on pseudo Zernike moment from structural MRI". In: *Neuroscience* 305, pp. 361–371.

Greve, D. N. and B. Fischl (2009). "Accurate and Robust Brain Image Alignment Using Boundary-Based Registration". In: *NeuroImage* 48.1, pp. 63–72. DOI: 10.1016/j.neuroimage.2009.06.060.

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

Guo, C. et al. (2017). "On Calibration of Modern Neural Networks". In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML'17. Sydney, NSW, Australia: JMLR.org, pp. 1321–1330.

Gutiérrez-Becker, B. and C. Wachinger (2018). "Deep Multi-Structural Shape Analysis: Application to Neuroanatomy". In: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 11072 LNCS, pp. 523–531. DOI: 10.1007/978-3-030-00931-1_60.

Habes, M. et al. (2020). "Disentangling Heterogeneity in Alzheimer's Disease and Related Dementias Using Data-Driven Methods". In: *Biological Psychiatry*. DOI: 10.1016/j.biopsych.2020.01.016.

Haller, S., K. O. Lovblad, and P. Giannakopoulos (2011). "Principles of classification analyses in mild cognitive impairment (MCI) and Alzheimer disease". In: *J. Alzheimers. Dis.* 26 Suppl 3, pp. 389–394.

He, K. et al. (2015). "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. Santiago, Chile: IEEE, pp. 1026–1034. DOI: 10.1109/ICCV.2015.123.

— (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Hon, M and N. M. Khan (2017). "Towards Alzheimer's disease classification through transfer learning". In: *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 1166–1169.

Hosseini-Asl, E, R Keynton, and A El-Baz (2016). "Alzheimer's disease diagnostics by adaptation of 3D convolutional network". In: *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 126–130.

Hosseini Asl, E. et al. (2018). "Alzheimer's disease diagnostics by a 3D deeply supervised adaptable convolutional network". In: *Front. Biosci.* 23.2, pp. 584–596.

Hu, J. et al. (2021). "Deep Learning-Based Classification and Voxel-Based Visualization of Frontotemporal Dementia and Alzheimer's Disease". In: *Frontiers in Neuroscience* 14. DOI: 10.3389/fnins.2020.626154.

Hutson, M. (2018). "Artificial Intelligence Faces Reproducibility Crisis". In: *Science* 359.6377, pp. 725–726. DOI: 10.1126/science.359.6377.725.

Islam, J. and Y. Zhang (2017). "A Novel Deep Learning Based Multi-class Classification Method for Alzheimer's Disease Detection Using Brain MRI Data". In: *Brain Informatics*. Ed. by Y. Zeng et al. Vol. 10654. Lecture Notes in Computer Science. Springer International Publishing, pp. 213–222.

— (2018). "Brain MRI analysis for Alzheimer's disease diagnosis using an ensemble system of deep convolutional neural networks". In: *Brain Inform* 5.2, p. 2.

Jack, C. R. et al. (2016). "A/T/N: An Unbiased Descriptive Classification Scheme for Alzheimer Disease Biomarkers". In: *Neurology* 87.5, pp. 539–547. DOI: 10.1212/WNL.0000000000002923.

Jenkinson, M. and S. Smith (2001). "A Global Optimisation Method for Robust Affine Registration of Brain Images". In: *Medical Image Analysis* 5.2, pp. 143–156. DOI: 10.1016/s1361-8415(01)00036-6.

Jenkinson, M. et al. (2002). "Improved Optimization for the Robust and Accurate Linear Registration and Motion Correction of Brain Images". In: *NeuroImage* 17.2, pp. 825–841. DOI: 10.1006/nimg.2002.1132.

Jha, D., J.-I. Kim, and G.-R. Kwon (2017). "Diagnosis of Alzheimer's Disease Using Dual-Tree Complex Wavelet Transform, PCA, and Feed-Forward Neural Network". In: *J. Healthc. Eng.* 2017, p. 9060124.

Jin, D. et al. (2020). "Generalizable, Reproducible, and Neuroscientifically Interpretable Imaging Biomarkers for Alzheimer's Disease". In: *Advanced Science* 7.14, p. 2000675. DOI: 10.1002/advs.202000675.

Jungo, A. et al. (2021). "Pymia: A Python Package for Data Handling and Evaluation in Deep Learning-Based Medical Image Analysis". In: *Computer Methods and Programs in Biomedicine* 198, p. 105796. DOI: 10.1016/j.cmpb.2020.105796.

Kalavathi, P. and V. B. S. Prasath (2016). "Methods on Skull Stripping of MRI Head Scan Images—a Review". In: *Journal of Digital Imaging* 29.3, pp. 365–379. DOI: 10.1007/s10278-015-9847-8.

Kaufman, S. et al. (2012). "Leakage in data mining: Formulation, detection, and avoidance". In: *ACM Transactions on Knowledge Discovery from Data* 6.4, 15:1–15:21. ISSN: 1556-4681. DOI: 10.1145/2382577.2382579.

Korolev, S et al. (2017). "Residual and plain convolutional neural networks for 3D brain MRI classification". In: *IEEE 14th International Symposium on Biomedical Imaging (ISBI)*, pp. 835–838.

Kriegeskorte, N. et al. (2009). "Circular analysis in systems neuroscience: the dangers of double dipping". In: *Nat. Neurosci.* 12.5, pp. 535–540.

Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). "Imagenet Classification with Deep Convolutional Neural Networks". In: *Advances in neural information processing systems* 25, pp. 1097–1105.

Kuhn, M. and K. Johnson (2013). *Applied Predictive Modeling*. Springer, New York, NY.

Lakhani, P. et al. (2018). "Hello World Deep Learning in Medical Imaging". In: *Journal of Digital Imaging* 31.3, pp. 283–289. ISSN: 1618-727X. DOI: 10.1007/s10278-018-0079-6.

Lam, B. et al. (2013). "Clinical, Imaging, and Pathological Heterogeneity of the Alzheimer's Disease Syndrome". In: *Alzheimer's Research & Therapy* 5.1, p. 1. DOI: 10.1186/alzrt155.

LeCun, Y., Y. Bengio, and G. Hinton (2015). "Deep learning". In: *Nature* 521.7553, pp. 436–444.

Ledig, C. et al. (2015). "Robust whole-brain segmentation: application to traumatic brain injury". In: *Med. Image Anal.* 21.1, pp. 40–58.

Lee, E. et al. (2019a). "Toward an Interpretable Alzheimer's Disease Diagnostic Model with Regional Abnormality Representation via Deep Learning". In: *NeuroImage* 202, p. 116113. DOI: 10.1016/j.neuroimage.2019.116113.

Lee, G. et al. (2019b). "MildInt: Deep Learning-Based Multimodal Longitudinal Data Integration Framework". In: *Frontiers in Genetics* 10. DOI: 10.3389/fgene.2019.00617.

Lee, G. et al. (2019c). "Predicting Alzheimer's Disease Progression Using Multi-Modal Deep Learning Approach". In: *Scientific Reports* 9.1, p. 1952. DOI: 10.1038/s41598-018-37769-z.

Leming, M., J. M. Górriz, and J. Suckling (2020). "Ensemble Deep Learning on Large, Mixed-Site fMRI Datasets in Autism and Other Tasks". In: *International Journal of Neural Systems*, p. 2050012. DOI: 10.1142/S0129065720500124.

Li, F., D. Cheng, and M. Liu (2017). "Alzheimer's Disease Classification Based on Combination of Multi-Model Convolutional Networks". In: *2017 IEEE International Conference on Imaging Systems and Techniques (IST)*, pp. 1–5. DOI: 10.1109/IST.2017.8261566.

Li, F., M. Liu, and Alzheimer's Disease Neuroimaging Initiative (2018). "Alzheimer's disease diagnosis based on multiple cluster dense convolutional networks". In: *Comput. Med. Imaging Graph.* 70, pp. 101–110.

Li, F. et al. (2015). "A Robust Deep Model for Improved Classification of AD/MCI Patients". In: *IEEE J Biomed Health Inform* 19.5, pp. 1610–1616.

Lian, C. et al. (2018). "Hierarchical Fully Convolutional Network for Joint Atrophy Localization and Alzheimer's Disease Diagnosis using Structural MRI". In: *IEEE Trans. Pattern Anal. Mach. Intell.*

Lin, W. et al. (2018). "Convolutional Neural Networks-Based MRI Image Analysis for the Alzheimer's Disease Prediction From Mild Cognitive Impairment". In: *Front. Neurosci.* 12, p. 777.

Lipton, Z. C. (2018). "The Mythos of Model Interpretability". In: *Communications of the ACM* 61.10, pp. 36–43. DOI: 10.1145/3233231.

Liu, J. et al. (2016). "Multi-view ensemble learning for dementia diagnosis from neuroimaging: An artificial neural network approach". In: *Neurocomputing* 195, pp. 112–116.

Liu, J. et al. (2018a). "Applications of Deep Learning to MRI Images: A Survey". In: *Big Data Mining and Analytics* 1.1, pp. 1–18. DOI: 10.26599/BDMA.2018.9020001.

Liu, M. et al. (2018b). "Multi-Modality Cascaded Convolutional Neural Networks for Alzheimer's Disease Diagnosis". In: *Neuroinformatics* 16.3-4, pp. 295–308.

Liu, M. et al. (2018c). "Anatomical Landmark Based Deep Feature Representation for MR Images in Brain Disease Diagnosis". In: *IEEE J Biomed Health Inform* 22.5, pp. 1476–1485.

Liu, M. et al. (2018d). "Joint Classification and Regression via Deep Multi-Task Multi-Channel Learning for Alzheimer's Disease Diagnosis". In: *IEEE Trans. Biomed. Eng.*

— (2018e). "Landmark-based deep multi-instance learning for brain disease diagnosis". In: *Med. Image Anal.* 43, pp. 157–168.

Liu, S. et al. (2015). "Multimodal neuroimaging feature learning for multiclass diagnosis of Alzheimer's disease". In: *IEEE Trans. Biomed. Eng.* 62.4, pp. 1132–1140.

Lu, D. and Q. Weng (2007). "A Survey of Image Classification Methods and Techniques for Improving Classification Performance". In: *International journal of Remote sensing* 28.5, pp. 823–870.

Lu, D. et al. (2018). "Multimodal and Multiscale Deep Neural Networks for the Early Diagnosis of Alzheimer's Disease using structural MR and FDG-PET images". In: *Sci. Rep.* 8.1, p. 5697.

Lundberg, S. M. and S.-I. Lee (2017). "A Unified Approach to Interpreting Model Predictions". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., pp. 4768–4777.

Lundervold, A. S. and A. Lundervold (2018). "An overview of deep learning in medical imaging focusing on MRI". In: *Z. Med. Phys.*

Madabhushi, A. and J. Udupa (2005). "Interplay between Intensity Standardization and Inhomogeneity Correction in MR Image Processing". In: *IEEE Transactions on Medical Imaging* 24.5, pp. 561–576. DOI: 10.1109/TMI.2004.843256.

Magesh, P. R., R. D. Myloth, and R. J. Tom (2020). "An Explainable Machine Learning Model for Early Detection of Parkinson's Disease Using LIME on DaTSCAN Imagery". In:

Computers in Biology and Medicine 126, p. 104041. DOI: 10.1016/j.compbiomed.2020.104041.

Mahanand, B. S. et al. (2012). "Identification of brain regions responsible for Alzheimer's disease using a Self-adaptive Resource Allocation Network". In: *Neural Netw.* 32, pp. 313–322.

Maitra, M. and A. Chatterjee (2006). "A Slantlet transform based intelligent system for magnetic resonance brain image classification". In: *Biomed. Signal Process. Control* 1.4, pp. 299–306.

Marcus, D. S. et al. (2007). "Open Access Series of Imaging Studies (OASIS): Cross-Sectional MRI Data in Young, Middle Aged, Nondemented, and Demented Older Adults". In: *Journal of Cognitive Neuroscience* 19.9, pp. 1498–1507. DOI: 10.1162/jocn.2007.19.9.1498.

Martinez-Murcia, F. J. et al. (2020). "Studying the Manifold Structure of Alzheimer's Disease: A Deep Learning Approach Using Convolutional Autoencoders". In: *IEEE Journal of Biomedical and Health Informatics* 24.1, pp. 17–26. DOI: 10.1109/JBHI.2019.2914970.

Mathew, N. A., R. S. Vivek, and P. R. Anurenjan (2018). "Early Diagnosis of Alzheimer's Disease from MRI Images Using PNN". In: *International CET Conference on Control, Communication, and Computing (IC4)*, pp. 161–164.

Montavon, G., W. Samek, and K.-R. Müller (2018). "Methods for Interpreting and Understanding Deep Neural Networks". In: *Digital Signal Processing* 73, pp. 1–15. DOI: 10.1016/j.dsp.2017.10.011.

Montavon, G. et al. (2017). "Explaining Nonlinear Classification Decisions with Deep Taylor Decomposition". In: *Pattern Recognition* 65, pp. 211–222. DOI: 10.1016/j.patcog.2016.11.008.

Moradi, E. et al. (2015). "Machine Learning Framework for Early MRI-Based Alzheimer's Conversion Prediction in MCI Subjects". In: *NeuroImage* 104, pp. 398–412. DOI: 10.1016/j.neuroimage.2014.10.002.

Mostapha, M. et al. (2018). "Non-Euclidean, convolutional learning on cortical brain surfaces". In: *IEEE 15th International Symposium on Biomedical Imaging (ISBI)* 2018, pp. 527–530.

Murray, M. E. et al. (2011). "Neuropathologically Defined Subtypes of Alzheimer's Disease with Distinct Clinical Characteristics: A Retrospective Study". In: *Lancet neurology* 10.9, pp. 785–796. DOI: 10.1016/S1474-4422(11)70156-9.

Nichols, E. et al. (2019). "Global, Regional, and National Burden of Alzheimer's Disease and Other Dementias, 1990–2016: A Systematic Analysis for the Global Burden of Disease Study 2016". In: *The Lancet Neurology* 18.1, pp. 88–106. DOI: 10.1016/S1474-4422(18)30403-4.

Nigri, E. et al. (2020). "Explainable Deep CNNs for MRI-Based Diagnosis of Alzheimer's Disease". In: *Proceedings of the International Joint Conference on Neural Networks*. DOI: 10.1109/IJCNN48605.2020.9206837.

Ning, K. et al. (2018). "Classifying Alzheimer's disease with brain imaging and genetic data using a neural network framework". In: *Neurobiol. Aging* 68, pp. 151–158.

Oh, K. et al. (2019). "Classification and Visualization of Alzheimer's Disease Using Volumetric Convolutional Neural Network and Transfer Learning". In: *Scientific Reports* 9.1, pp. 1–16. DOI: 10.1038/s41598-019-54548-6.

Olah, C., A. Mordvintsev, and L. Schubert (2017). "Feature Visualization". In: *Distill* 2.11, e7. DOI: 10.23915/distill.00007.

Oliveira, F. P. M. and J. M. R. S. Tavares (2014). "Medical Image Registration: A Review". In: *Computer Methods in Biomechanics and Biomedical Engineering* 17.2, pp. 73–93. DOI: 10.1080/10255842.2012.670855.

Ortiz, A. et al. (2016). "Ensembles of Deep Learning Architectures for the Early Diagnosis of the Alzheimer's Disease". In: *Int. J. Neural Syst.* 26.7, p. 1650025.

Panwar, H. et al. (2020). "Application of Deep Learning for Fast Detection of COVID-19 in X-Rays Using nCOVnet". In: *Chaos, Solitons & Fractals* 138, p. 109944. DOI: 10.1016/j.chaos.2020.109944.

Parisot, S. et al. (2018). "Disease prediction using graph convolutional networks: Application to Autism Spectrum Disorder and Alzheimer's disease". In: *Med. Image Anal.* 48, pp. 117–130.

Paszke, A. et al. (2017). "Automatic differentiation in PyTorch".

Pawlowski, N. et al. (2017). "DLTK: State of the Art Reference Implementations for Deep Learning on Medical Images". In: *arXiv:1711.06853 [cs]*.

Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python". In: *J. Mach. Learn. Res.* 12.Oct, pp. 2825–2830.

Pérez-García, F., R. Sparks, and S. Ourselin (2021). "TorchIO: A Python Library for Efficient Loading, Preprocessing, Augmentation and Patch-Based Sampling of Medical Images in Deep Learning". In: *Computer Methods and Programs in Biomedicine* 208, p. 106236. DOI: 10.1016/j.cmpb.2021.106236.

Petersen, R. C. et al. (2010). "Alzheimer's Disease Neuroimaging Initiative (ADNI): clinical characterization". In: *Neurology* 74.3, pp. 201–209.

Pérez-García, F. et al. (2020). *EPISURG: a dataset of postoperative magnetic resonance images (MRI) for quantitative analysis of resection neurosurgery for refractory epilepsy*. DOI: 10.5522/04/9996158.v1. URL: /articles/dataset/EPISURG_a_dataset_of_postoperative_magnetic_resonance_images_MRI_for_quantitative_analysis_of_resection_neurosurgery_for_refractory_epilepsy/9996158/1.

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

Qiu, S. et al. (2020). "Development and Validation of an Interpretable Deep Learning Framework for Alzheimer's Disease Classification". In: *Brain: A Journal of Neurology* 143.6, pp. 1920–1933. DOI: 10.1093/brain/awaa137.

Raschka, S. (2015). *Python Machine Learning*. Packt Publishing Ltd.

Rathore, S. et al. (2017). “A review on neuroimaging-based classification studies and associated feature extraction methods for Alzheimer’s disease and its prodromal stages”. In: *Neuroimage* 155, pp. 530–548.

Raut, A. and V. Dalal (2017). “A Machine Learning Based Approach for Detection of Alzheimer’s Disease Using Analysis of Hippocampus Region from MRI Scan”. In: *2017 International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 236–242. DOI: 10.1109/ICCMC.2017.8282683.

Ravi, D., D. C. Alexander, and N. P. Oxtoby (2019). “Degenerative Adversarial NeuroImage Nets: Generating Images That Mimic Disease Progression”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*. Ed. by D. Shen et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 164–172. DOI: 10.1007/978-3-030-32248-9_19.

Razzak, M. I., S. Naz, and A. Zaib (2018). “Deep Learning for Medical Image Processing: Overview, Challenges and the Future”. In: *Classification in BioApps: Automation of Decision Making*. Ed. by N. Dey, A. S. Ashour, and S. Borra. Cham: Springer International Publishing, pp. 323–350.

Reitz, C. (2012). “Alzheimer’s Disease and the Amyloid Cascade Hypothesis: A Critical Review”. In: *International Journal of Alzheimer’s Disease* 2012, pp. 1–11. DOI: 10.1155/2012/369808.

Ribeiro, M. T., S. Singh, and C. Guestrin (2016). “"Why Should I Trust You?": Explaining the Predictions of Any Classifier”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD ’16*. San Francisco, California, USA: ACM Press, pp. 1135–1144. DOI: 10.1145/2939672.2939778.

Rieke, J. et al. (2018). “Visualizing Convolutional Networks for MRI-Based Diagnosis of Alzheimer’s Disease”. In: *Understanding and Interpreting Machine Learning in Medical Image Computing Applications*. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 24–31. DOI: 10.1007/978-3-030-02628-8_3.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks by Brian D. Ripley*. Cambridge University Press.

Rolls, E. T., M. Joliot, and N. Tzourio-Mazoyer (2015). “Implementation of a New Parcellation of the Orbitofrontal Cortex in the Automated Anatomical Labeling Atlas”. In: *NeuroImage* 122, pp. 1–5. DOI: 10.1016/j.neuroimage.2015.07.075.

Routier, A. et al. (2021). “Clinica: An Open-Source Software Platform for Reproducible Clinical Neuroscience Studies”. In: *Frontiers in Neuroinformatics* 15, p. 39. DOI: 10.3389/fninf.2021.689675.

Salvatore, C. et al. (2015). “Magnetic resonance imaging biomarkers for the early diagnosis of Alzheimer’s disease: a machine learning approach”. In: *Front. Neurosci.* 9, p. 307.

Samala, R. K. et al. (2020). “Hazards of Data Leakage in Machine Learning: A Study on Classification of Breast Cancer Using Deep Neural Networks”. In: *Medical Imaging 2020: Computer-Aided Diagnosis*. Vol. 11314. International Society for Optics and Photonics, p. 1131416. DOI: 10.1117/12.2549313.

Samek, W. et al. (2017). "Evaluating the Visualization of What a Deep Neural Network Has Learned". In: *IEEE Transactions on Neural Networks and Learning Systems* 28.11, pp. 2660–2673. DOI: 10.1109/TNNLS.2016.2599820.

Samper-González, J. et al. (2018). "Reproducible Evaluation of Classification Methods in Alzheimer's Disease: Framework and Application to MRI and PET Data". In: *bioRxiv*, p. 274324. DOI: 10.1101/274324.

Sarle, W. S. (1997). "Neural Network FAQ, part 1 of 7". In: *Introduction, periodic posting to the Usenet newsgroup comp. ai. neural-nets URL: ftp://ftp. sas. com/pub/neural/FAQ. html*.

Schuff, N et al. (2009). "MRI of hippocampal volume loss in early Alzheimer's disease in relation to ApoE genotype and biomarkers". In: *Brain* 132.Pt 4, pp. 1067–1077.

Selvaraju, R. R. et al. (2017). "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626. DOI: 10.1109/ICCV.2017.74.

Senanayake, U, A Sowmya, and L Dawes (2018). "Deep fusion pipeline for mild cognitive impairment diagnosis". In: *IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pp. 1394–1997.

Shams-Baboli, A and M Ezoji (2017). "A Zernike moment based method for classification of Alzheimer's disease from structural MRI". In: *3rd International Conference on Pattern Recognition and Image Analysis (IPRIA)*, pp. 38–43.

Shattuck, D. W. et al. (2001). "Magnetic Resonance Image Tissue Classification Using a Partial Volume Model". In: *NeuroImage* 13.5, pp. 856–876. DOI: 10.1006/nimg.2000.0730.

Shen, T et al. (2018). "Decision Supporting Model for One-year Conversion Probability from MCI to AD using CNN and SVM". In: *40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 738–741.

Shepp, L. A. and B. F. Logan (1974). "The Fourier reconstruction of a head section". In: *IEEE Transactions on Nuclear Science* 21.3, pp. 21–43. DOI: 10.1109/TNS.1974.6499235.

Shi, J. et al. (2018). "Multimodal Neuroimaging Feature Learning With Multimodal Stacked Deep Polynomial Networks for Diagnosis of Alzheimer's Disease". In: *IEEE J Biomed Health Inform* 22.1, pp. 173–183.

Shmulev, Y., M. Belyaev, and The Alzheimer's Disease Neuroimaging Initiative (2018). "Predicting Conversion of Mild Cognitive Impairments to Alzheimer's Disease and Exploring Impact of Neuroimaging: Second International Workshop, GRAIL 2018 and First International Workshop, Beyond MIC 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings". In: *Graphs in Biomedical Image Analysis and Integrating Medical Imaging and Non-Imaging Modalities*. Ed. by D. Stoyanov et al. Vol. 11044. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 83–91.

Shrikumar, A. et al. (2017). "Not Just a Black Box: Learning Important Features Through Propagating Activation Differences". In: *arXiv:1605.01713 [cs]*.

Simonyan, K. and A. Zisserman (2015). “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In:

Simonyan, K., A. Vedaldi, and A. Zisserman (2014). “Deep inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps”. In: *In Workshop at International Conference on Learning Representations*.

Simpson, A. L. et al. (2019). “A Large Annotated Medical Image Dataset for the Development and Evaluation of Segmentation Algorithms”. In: *arXiv:1902.09063 [cs, eess]*.

Sled, J. G., A. P. Zijdenbos, and A. C. Evans (1998). “A Nonparametric Method for Automatic Correction of Intensity Nonuniformity in MRI Data”. In: *IEEE Transactions on Medical Imaging* 17.1, pp. 87–97. DOI: 10.1109/42.668698.

Smith, S. M. (2002). “Fast Robust Automated Brain Extraction”. In: *Human Brain Mapping* 17.3, pp. 143–155. DOI: 10.1002/hbm.10062.

Spasov, S. E. et al. (2018). “A Multi-modal Convolutional Neural Network Framework for the Prediction of Alzheimer’s Disease”. In: *Conf. Proc. IEEE Eng. Med. Biol. Soc.* 2018, pp. 1271–1274.

Springenberg, J. et al. (2015). “Striving for Simplicity: The All Convolutional Net”. In: *ICLR (workshop track)*.

Stodden, V. et al. (2016). “Enhancing Reproducibility for Computational Methods”. In: *Science* 354.6317, pp. 1240–1241. DOI: 10.1126/science.aah6168.

Suk, H.-I. et al. (2014). “Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis”. In: *Neuroimage* 101, pp. 569–582.

— (2015). “Latent feature representation with stacked auto-encoder for AD/MCI diagnosis”. In: *Brain Struct. Funct.* 220.2, pp. 841–859.

— (2017). “Deep ensemble learning of sparse regression models for brain disease diagnosis”. In: *Med. Image Anal.* 37, pp. 101–113.

Tang, Z. et al. (2019). “Interpretable Classification of Alzheimer’s Disease Pathologies with a Convolutional Neural Network Pipeline”. In: *Nature Communications* 10.1, pp. 1–14. DOI: 10.1038/s41467-019-10212-1.

Taqi, A. M. et al. (2018). “The Impact of Multi-Optimizers and Data Augmentation on TensorFlow Convolutional Neural Network Performance”. In: *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pp. 140–145.

Thibeau-Sutre, E. et al. (2020). “Visualization Approach to Assess the Robustness of Neural Networks for Medical Image Classification”. In: *Medical Imaging 2020: Image Processing*. Vol. 11313. International Society for Optics and Photonics, 113131J. DOI: 10.1117/12.2548952.

Thompson, W. H. et al. (2020). “Dataset Decay and the Problem of Sequential Analyses on Open Datasets”. In: *eLife* 9, e53498. DOI: 10.7554/eLife.53498.

Thung, K.-H., P.-T. Yap, and D. Shen (2017). “Multi-stage Diagnosis of Alzheimer’s Disease with Incomplete Multimodal Data via Multi-task Deep Learning”. In: *Deep Learn Med Image Anal Multimodal Learn Clin Decis Support* 10553, pp. 160–168.

Tomsett, R. et al. (2020). "Sanity Checks for Saliency Metrics". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.04, pp. 6021–6029. DOI: 10.1609/aaai.v34i04.6064.

Tustison, N. J. et al. (2010). "N4ITK: Improved N3 Bias Correction". In: *IEEE Transactions on Medical Imaging* 29.6, pp. 1310–1320. DOI: 10.1109/TMI.2010.2046908.

Uchida, S. (2013). "Image Processing and Recognition for Biological Images". In: *Development, growth & differentiation* 55.4, pp. 523–549.

Valliani, A. and A. Soni (2017). "Deep Residual Nets for Improved Alzheimer's Diagnosis". In: *8th ACM International Conference on Bioinformatics, Computational Biology,and Health Informatics*. ACM, pp. 615–615.

Voss, C. et al. (2021). "Visualizing Weights". In: *Distill* 6.2, e00024.007. DOI: 10.23915/distill.00024.007.

Vovk, U., F. Pernus, and B. Likar (2007). "A Review of Methods for Correction of Intensity Inhomogeneity in MRI". In: *IEEE transactions on medical imaging* 26.3, pp. 405–421.

Vu, T. D. et al. (2017). "Multimodal learning using Convolution Neural Network and Sparse Autoencoder". In: *IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 309–312.

Vu, T.-D. et al. (2018). "Non-white matter tissue extraction and deep convolutional neural network for Alzheimer's disease detection". In: *Soft Comput* 22.20, pp. 6825–6833.

Wang, F. et al. (2017a). "Residual Attention Network for Image Classification". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI: IEEE, pp. 6450–6458. DOI: 10.1109/CVPR.2017.683.

Wang, H. et al. (2019). "Ensemble of 3D densely connected convolutional network for diagnosis of mild cognitive impairment and Alzheimer's disease". In: *Neurocomputing* 333, pp. 145–156.

Wang, S.-H. et al. (2018a). "Classification of Alzheimer's Disease Based on Eight-Layer Convolutional Neural Network with Leaky Rectified Linear Unit and Max Pooling". In: *J. Med. Syst.* 42.5, p. 85.

Wang, S. et al. (2017b). "Automatic Recognition of Mild Cognitive Impairment from MRI Images Using Expedited Convolutional Neural Networks". In: *Artificial Neural Networks and Machine Learning – ICANN 2017*. Springer International Publishing, pp. 373–380.

Wang, X. et al. (2018b). "Temporal Correlation Structure Learning for MCI Conversion Prediction: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part III". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. Ed. by A. F. Frangi et al. Vol. 11072. Lecture Notes in Computer Science. Springer International Publishing, pp. 446–454.

Wen, D. et al. (2018). "Deep Learning Methods to Process fMRI Data and Their Application in the Diagnosis of Cognitive Impairment: A Brief Overview and Our Opinion". In: *Front. Neuroinform.* 12, p. 23.

Wen, J. et al. (2020). "Convolutional Neural Networks for Classification of Alzheimer's Disease: Overview and Reproducible Evaluation". In: *Medical Image Analysis* 63, p. 101694. DOI: 10.1016/j.media.2020.101694.

Wen, J. et al. (2021). "Reproducible Evaluation of Diffusion MRI Features for Automatic Classification of Patients with Alzheimer's Disease". In: *Neuroinformatics* 19.1, pp. 57–78. DOI: 10.1007/s12021-020-09469-5.

Whitwell, J. L. et al. (2007). "3D Maps from Multiple MRI Illustrate Changing Atrophy Patterns as Subjects Progress from Mild Cognitive Impairment to Alzheimer's Disease". In: *Brain* 130.7, pp. 1777–1786. DOI: 10.1093/brain/awm112.

Wood, D., J. Cole, and T. Booth (2019). "NEURO-DRAM: A 3D Recurrent Visual Attention Model for Interpretable Neuroimaging Classification". In: *arXiv:1910.04721 [cs, stat]*. arXiv: 1910.04721 [cs, stat].

Wu, C. et al. (2018). "Discrimination and conversion prediction of mild cognitive impairment using convolutional neural networks". In: *Quant. Imaging Med. Surg.* 8.10, pp. 992–1003.

Xie, N. et al. (2020). "Explainable Deep Learning: A Field Guide for the Uninitiated". In: *arXiv:2004.14545 [cs, stat]*.

Xu, K. et al. (2015). "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention". In: *Proceedings of the 32nd International Conference on Machine Learning*. PMLR, pp. 2048–2057.

Yagis, E. et al. (2021). *Deep Learning in Brain MRI: Effect of Data Leakage Due to Slice-Level Split Using 2D Convolutional Neural Networks*. Preprint. In Review. DOI: 10.21203/rs.3.rs-464091/v1.

Yeh, C.-K. et al. (2019). "On the (In)Fidelity and Sensitivity of Explanations". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., pp. 10967–10978.

Zeiler, M. D. and R. Fergus (2014). "Visualizing and Understanding Convolutional Networks". In: *Computer Vision – ECCV 2014*. Ed. by D. Fleet et al. Lecture Notes in Computer Science. Springer International Publishing, pp. 818–833.

Zhang, L. et al. (2020). "A Survey on Deep Learning for Neuroimaging-Based Brain Disorder Analysis". In: *Frontiers in Neuroscience* 14, p. 779. DOI: 10.3389/fnins.2020.00779.

Zhang, T. and M. Shi (2020). "Multi-Modal Neuroimaging Feature Fusion for Diagnosis of Alzheimer's Disease". In: *Journal of Neuroscience Methods* 341, p. 108795. DOI: 10.1016/j.jneumeth.2020.108795.

Zhang, Y. et al. (2018). "Multivariate Approach for Alzheimer's Disease Detection Using Stationary Wavelet Entropy and Predator-Prey Particle Swarm Optimization". In: *J. Alzheimers. Dis.* 65.3, pp. 855–869.

Zhong, Z. et al. (2020). "Random Erasing Data Augmentation". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.07, pp. 13001–13008. DOI: 10.1609/aaai.v34i07.7000.

Zhou, B. et al. (2016). "Learning Deep Features for Discriminative Localization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921–2929.

Zhou, T. et al. (2017). “Feature Learning and Fusion of Multimodality Neuroimaging and Genetic Data for Multi-status Dementia Diagnosis”. In: *Mach Learn Med Imaging* 10541, pp. 132–140.

— (2019). “Effective feature learning and fusion of multimodality data using stage-wise deep neural network for dementia diagnosis”. In: *Hum. Brain Mapp.* 40.3, pp. 1001–1016.