Intelligence artificielle et maladies neurologiques : aider le diagnostic et améliorer la compréhension du comportement des réseaux de neurones convolutifs

Giulia Maria Mattia

THÈSE

En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse 3 - Paul Sabatier

Présentée et soutenue par

Giulia Maria MATTIA

Le 9 décembre 2022

Intelligence Artificielle et Maladies Neurologiques : Aider le Diagnostic et Améliorer la Compréhension du Comportement des Réseaux de Neurones Convolutifs

---

Ecole doctorale : **GEETS - Génie Electrique Electronique,Télécommunications et Santé : du système au nanosystème**

Spécialité : **Radiophysique et Imagerie Médicales**

Unité de recherche :
ToNIC-Toulouse NeuroImaging Center (UMR 1214)

Thèse dirigée par
Patrice PERAN et Xavier FRANCERIES

Jury

M. Vincent LABATUT, Rapporteur
M. Nacim BETROUNI, Rapporteur
M. Stéphane LEHERICY, Examinateur
M. Patrice PERAN, Directeur de thèse
M. Xavier FRANCERIES, Co-directeur de thèse
Mme Isabelle BERRY, Présidente

A dissertation in order to obtain the title of
DOCTOR OF THE UNIVERSITY OF TOULOUSE
Issued by Université Toulouse 3 - Paul Sabatier

Defended by

Giulia Maria MATTIA

On 9 December 2022

Artificial Intelligence for Neurological Disorders: Aiding the Diagnosis and Better Understanding Convolutional Neural Network Behavior

---

PhD School: **GEETS - Génie Electrique Electronique, Télécommunications et Santé : du système au nanosystème**

Department: **Radiophysique et Imagerie Médicales**

Research Laboratory: **ToNIC-Toulouse NeuroImaging Center (UMR 1214)**

Supervised by

Patrice PERAN and Xavier FRANCERIES

The jury members

Mr. Vincent LABATUT, Referee

Mr. Nacim BETROUNI, Referee

Mr. Stéphane LEHERICY, Examiner

Mr. Patrice PERAN, PhD Supervisor

Mr. Xavier FRANCERIES, PhD Co-supervisor

Mrs. Isabelle BERRY, President

Acknowledgments

These three years have gone by in the blink of an eye. It feels like yesterday that I began this journey, yet here we are at its conclusion, which marks a new beginning.
These three years have been dense with some challenging events which have consolidated our strive while fortifying our spirits.

I wish to express my sincere gratitude to all the jury members who accepted to evaluate this work, Mr. Stéphane Lehéricy as examiner and Mrs. Isabelle Berry as jury president, and Mr. Vincent Labatut and Mr. Nacim Betrouni as referees. It has been an honor to receive your feedback and benefit from your expertise, which offered me new leads to explore and insightful points of reflection.
I am also extremely grateful to my supervisors, Patrice Péran and Xavier Franceries, for their advice and patience as they believed in me and endured my many colorful results. Thank you for trusting me with this research project, for your assistance and dedicated involvement, and for listening to my ideas while steering me on the right path. Xavier, you also accompanied me in the teaching experience, which was enriching and extremely valuable. Thank you for your support and precious help. Patrice, you have shown me what it means to have an idea and gradually and concretely develop it, with all the difficulties and satisfactions one can encounter. Thank you for your guidance and constant presence and for sharing your expertise.
I take this opportunity to thank the Université Toulouse III - Paul Sabatier and the GEETS Ph.D. School, particularly Mrs. Marie Estruga, for their availability and promptness of response in all administrative and educational procedures.
I would like to thank the MRI technical platform at Toulouse Neuroimaging Center (ToNIC), Université de Toulouse, Inserm, UPS, France, and all members of the ToNIC Laboratory, especially Maryline, and Déborah. Thanks also to Federico, Edouard, Stein, and Benjamine for their contribution to this research work and constructive scientific discussions.
Thanks to my colleagues, and representatives of the GEETS Ph.D. students, Alice, Valentin, Youssef, François, Pierre, and Maxime, for their cooperation and support in our role as representatives.

These three years would not have been the same without my wonderful Alessandro. You are my soulmate, my best friend, my inspiration, and my eternal companion. Thank you for your endless caring, for always being by my side and empowering me to be the best version of myself, and for making each day count.
It is incredible how much distance can become infinitesimal, even at 2000 km. Thank you to my mother Angela and my grandparents Rosanna and Pino. You have supported me with all my choices, keeping a relentless faith in my abilities, as a constant reminder of your unconditional love. Thanks to my uncles, Paola and Massimo, and my cousins, Marco and Alessia, for their love and support. Sergio, Emilia, Silvia, Nonno Tonino, and Nonna Elvira, you are my extended family for whom I am always grateful. Thank you for your amazing presence, unconditional love, and encouragement throughout these years.
Thanks to all my friends and colleagues, Annagrazia, Désiré, Gerardina, Antonio, Margherita, Irene, Manfredo, Marthe, Perrine, Carla, Wafae, Marie, Sabrina, and all the others I did not mention, who have contributed to making this experience full of memorable events.

I shall conclude with this meaningful quote from John Powell:
“The only real mistake is the one from which we learn nothing.”

Giulia

Contents

List of Tables

List of Figures

Résumé

L'Intelligence artificielle est désormais utilisée pour accomplir les tâches les plus diverses, de la reconnaissance de visage à la traduction de texte. Parmi ces méthodes inspirées du fonctionnement du cerveau humain, l'apprentissage profond (*deep learning*) a montré d'excellentes performances en analyse d'image à l'aide des réseaux de neurones convolutifs (CNN). Le milieu médical est en train de bénéficier de la puissance de ces outils consacrés notamment à l'aide au diagnostic, comme dans la maladie de Parkinson ou d'Alzheimer. L'utilisation des CNN et de l'imagerie par résonance magnétique nucléaire (IRM), qui permet d'étudier le cerveau dans sa structure et son fonctionnement, a montré des résultats très prometteurs. Toutefois, les CNN sont souvent appelés « boites noires » puisque leur fonctionnement n'est pas transparent pour ses utilisateurs.

Ces travaux de thèse visent à mieux comprendre ces méthodes appliquées aux données IRM 3D cérébrales pour aider le diagnostic des maladies neurologiques. En première étape, la manipulation des données d'entrée des CNN, nous a permis d'investiguer leur capacité discriminative. Nous avons ainsi étudié le comportement des CNN en comparant leur capacité à discriminer des images IRM originales et altérées. Les résultats obtenus par les CNN ont été très satisfaisants, ce qui a amené à rechercher quelles sont les zones de l'image les plus discriminantes pour la prédiction.

En deuxième étape, nous avons étudié la pathologie, en nous focalisant sur le nombre des sujets nécessaires au réseau lors de l'apprentissage pour garantir de bonnes performances. Cela est aussi un aspect crucial pour les méthodes de deep learning dont l'apprentissage requiert normalement beaucoup de données. Toutefois, dans le cadre médical nous avons accès à quelques centaines de données dans la plupart des cas. Nous avons démontré qu'un réseau de neurones convolutifs est capable de bien discriminer un sujet sain d'un patient atteint d'atrophie multisystématisée (AMS), malgré un nombre limité de données d'entrée. A l'aide d'une technique récemment développée permettant de visualiser les parties de l'image considérées importantes par les CNN, nous avons montré que les parties discriminantes comprenaient des régions notamment d'intérêt pour la physiopathologie connue de l'AMS.

La puissance discriminante des CNN a aussi été exploitée pour réaliser une discrimination entre sujets sains et patients en état de coma, en utilisant différentes séquences d'IRM. La méthode de visualisation a mis en lumière des régions en lien avec le coma, en confirmant les performances très satisfaisantes du réseau.

Les études présentées dans cette thèse ouvrent la voie pour découvrir comment les informations englobées dans les données d'apprentissage peuvent aider à la recherche des signatures spatiales significatives obtenues par les CNN dans le cas particulier des données de neuroimagerie.
L'application des CNN dans le cadre médical offre la possibilité d'aider le diagnostic de différentes maladies neurologiques en se basant exclusivement sur les données d'entrée. Cependant, la validité de ces résultats se fonde sur notre capacité à expliquer et éclairer ces méthodes pour en favoriser l'acceptation et, par conséquence, l'utilisation dans un contexte clinique.

Abstract

Artificial Intelligence (AI) currently permeates several aspects of our everyday lives. It allows for solving complex tasks such as face recognition or text translation. Among these powerful tools inspired by brain functioning, deep learning methods have recently been gaining ground in image analysis thanks to the rise of Convolutional Neural Networks (CNNs). In the biomedical domain, these methods have found great success in discriminating neurological diseases, such as Parkinson's or Alzheimer's disease. Many applications using Magnetic Resonance Imaging (MRI) in combination with CNNs have shown promising results. Despite these outstanding performances, CNNs are considered "black boxes" as their decision-making process is not always transparent for human operators.

This Ph.D. thesis aims to better understand these methods applied to 3D brain MRI data and aid the diagnosis of neurological diseases. First, we characterized CNN behavior by altering ad hoc brain MRI data. That allowed the investigation of CNN performance to distinguish original from altered brain MRI images. Given the satisfying results, we searched for a spatial signature to discover the most discriminant image parts for CNN prediction.

Secondly, we examined CNN performance by considering a specific pathology and focusing on the number of training data needed to discern patients with Multiple System Atrophy (MSA) from healthy controls. Indeed, another crucial aspect of deep learning is the quantity of data necessary for the network to learn. However, it is not uncommon to deal with a lack of data in the medical domain, with only a few hundreds available. We showed that, even with a limited number of samples, the network could perform the task on new data. Using a recently developed visualization technique, we found that the most discriminant regions were in line with the known MSA physiopathology. Furthermore, using different MRI sequences, we exploited the discriminating power of CNNs to classify normal subjects against comatose patients. We obtained excellent performances supported by the visualization maps, which included regions of interest for patients in coma.

The work presented in this thesis opens the way for discovering how the information enclosed in the input data may aid the diagnosis and provide evidence for CNN decisions. That can lead to finding significant spatial signatures relative to the pathology.
CNN methods for health applications offer the possibility to support the diagnosis of neurological disorders with a data-driven approach. Nevertheless, it is pivotal to demonstrate the validity of these methods to favor their acceptance and use.

1 State of the Art

1.1 Neuroimaging

Over the past few decades, advancements in technology have led to thorough *in vivo* analyses of the structure and functioning of the brain. Biomarkers extracted using imaging techniques such as Positron Emission Tomography (PET), Computed Tomography (CT), and Magnetic Resonance Imaging (MRI) allow for characterization and follow-up regarding a plethora of brain-related pathologies [1–3].
A particular focus on MRI is provided in the following sections with an overview of principles and techniques.

1.1.1 Magnetic Resonance Imaging

MRI is a non-invasive and non-ionizing imaging technique based on the nuclear magnetic resonance phenomenon. The latter occurs when atomic nuclei (e.g. hydrogen protons), subject to an intense static magnetic field, are exposed to a variable magnetic field at a specific frequency, called the Larmor fequency. MRI has proven extremely helpful in gaining insights into anatomical and functional aspects of the human body.

Protected by a thermally isolated container, superconducting coils immersed in liquid helium at 4 K can produce high-intensity magnetic fields. This type of coil avoids energy dissipation due to Joule heating. Nowadays, imaging at 1.5 T, 3 T, and even 7 T is feasible and used for diagnostic purposes.
Only nuclei with non-null spin are eligible to perform MRI [4]. In physics, the spin of an elementary particle was considered at first a property related to magnetic moments, later associated with a pure quantum property, expressed by a number multiple of $\frac{1}{2}$. Let us briefly explain some concepts necessary to understand the MRI phenomenon.

A particle with charge q moving at velocity $\vec{v}$, immersed in a magnetic field $\vec{B}$, undergoes the effect of a magnetic force known as Lorentz force, defined in (1.1):

$$\vec{F} = q\vec{v} \wedge \vec{B} \tag{1.1}$$

We can also define the magnetic moment $\vec{\mu}$ as in (1.2), being the product between γ, the gyromagnetic ratio specific to the type of nuclei, and $\vec{J}$, the angular momentum of the particle.

$$\vec{\mu} = \gamma \vec{J} \tag{1.2}$$

Hydrogen nuclei are eligible for MRI, having spin $I = \frac{1}{2}$. They present two possible orientations for magnetic moments: parallel $\vec{\mu}_p$ and antiparallel $\vec{\mu}_{ap}$, respectively with equal and opposite orientations to the magnetic field. Under a static magnetic field $\vec{B_0}$, the spins of these protons orient around B_0 in a double cone as in Fig. 1.2.
Immersed in a magnetic field $\vec{B_0}$, charged particles with magnetic moment $\vec{\mu}$ are submitted to the moment $\vec{\Gamma}$ due to the magnetic force, defined in (1.3). That makes magnetic moments $\vec{\mu}$ initiate a precession movement around the magnetic field direction.

$$\vec{\Gamma} = \vec{\mu} \wedge \vec{B} \tag{1.3}$$

Proton precession occurs at a specific frequency, known as *Larmor frequency*, identified by ν_0 in (1.4).

$$\nu_0 = \frac{\gamma}{2\pi} B_0 \tag{1.4}$$

As a result of magnetic interactions under the effect of $\vec{B_0}$, the hydrogen nuclei population split into two energy levels. This phenomenon is known as the *Zeeman effect*. The resulting magnetization $\vec{M} = \vec{M_0}$ is directly proportional to the difference between these two energy levels.
Since $\vec{M_0}$ has a very low intensity, measuring it with a classic approach may reveal quite challenging. To tackle this issue, we can perturb the system by applying an electromagnetic wave. The latter presents low energy but the same precession frequency ν_0 (i.e. the Larmor frequency) inducing the resonance phenomenon and must be perpendicular to the direction of $\vec{B_0}$. The Radio Frequency (RF) wave is then produced by the magnetic resonance field $\vec{B_1}$ and the electric field $\vec{E_1}$. $\vec{B_1}$ is needed for the imaging part, whereas $\vec{E_1}$ is just responsible for heat deposition in the body.

The application of $\vec{B_1}$ causes parallel and anti-parallel magnetic moments to get into phase coherence, resulting in the transversal component of the magnetization.
To quantify the energy absorbed per unit mass under the effect of an RF wave, we measure the Specific Absorption Rate (SAR) during image acquisition. According to safety standards, it should not overcome a value of 4 W/kg on the entire body for an acquisition lasting 15 min.
Shortly after the application of $\vec{B_1}$, due to the resonance, the resulting magnetization $\vec{M}$ is directed around $\vec{B_1}$ with angular velocity equal to ω_1 in (1.5).

$$\omega_1 = \gamma B_1 \tag{1.5}$$

When the RF pulse lasting a time interval Δt stops, an angle φ appears and characterizes the two components of the resulting magnetization. This angle is defined in (1.6) and represented in Fig. 1.1. The effect of an RF pulse on magnetic moments can be observed in Fig. 1.2.

$$\varphi\ (rad) = \omega_1 \Delta t = \gamma B_1 \Delta t \tag{1.6}$$

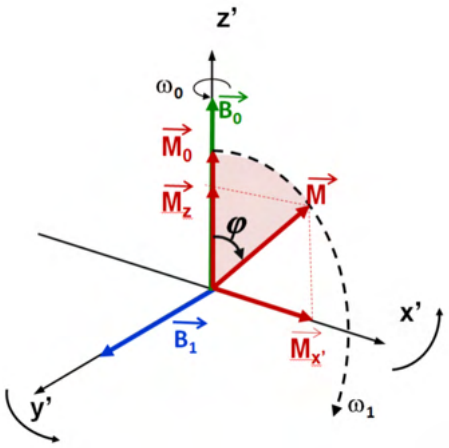


Figure 1.1: Representation of the angle φ resulting from the application of an RF pulse. Reproduced with permission from [4]

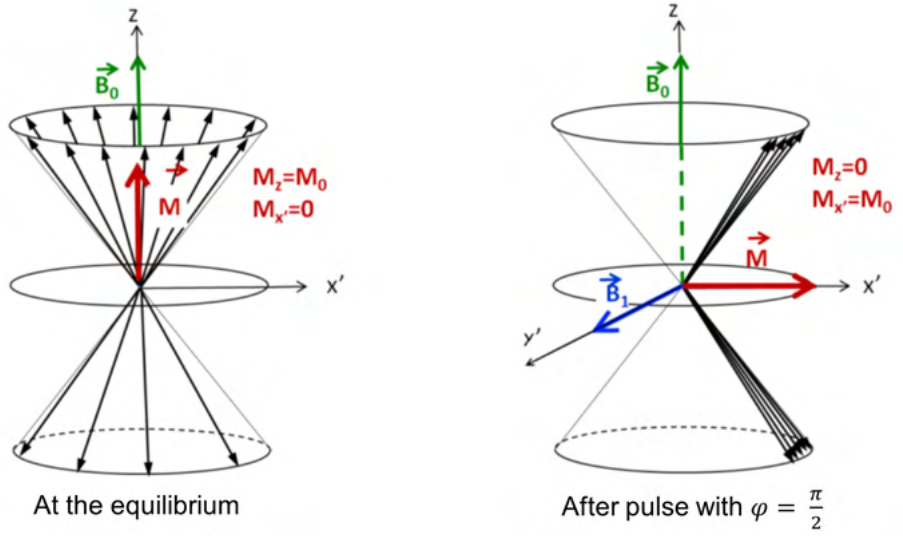


Figure 1.2: Effect of an RF pulse on magnetic moments. At the equilibrium, the resulting magnetization $\vec{M}$ is directed as the static magnetic field $\vec{B_0}$. After a pulse with $\varphi = \pi/2$, $\vec{M}$ is directed perpendicularly to $\vec{B_0}$. Reproduced with permission from [4]

RF sequences are also characterized by the *Repetition Time (TR)*, which is the time interval between two consecutive pulses (application of B_1), and the *Echo Time (TE)*, which is

the time between the RF pulse delivery and the actual receipt of the echo signal.
Once the RF pulse ends, the system returns to the equilibrium state, which is called *relaxation*. The magnetic field variation produces an electric current in the coil, proportional to the variation of the transverse component M_x. This signal is known as *Free Induction Decay (FID)* and represents the Magnetic Resonance (MR) signal to be measured, represented in Fig. 1.3.

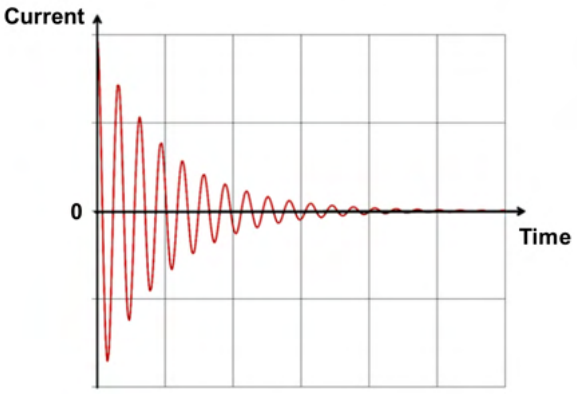


Figure 1.3: Example of free-induction decay signal. Adapted from [4]

According to relaxation time and pulse angle, we can distinguish [5]:

- *T1*, known as *longitudinal relaxation time*, indicating the time at which excited protons return to equilibrium and realign with $\vec{B_0}$;
- *T2*, known as *transverse relaxation time*, indicating the time at which excited protons return to equilibrium or go out of phase. It determines the time necessary to lose phase coherence among the nuclei spinning perpendicularly to $\vec{B_0}$.

1.1.2 MRI Sequences

Tweaking the parameters of FID signals, different MRI sequences (also called modalities or indices) can be defined, each expressing specific properties of the imaged tissues. The most common sequences are briefly described in the following.

- *T1-weighted*, produced by short TR ($\approx$ 500 ms) and TE ($\approx$ 15 ms), with image characteristics (e.g. contrast, brightness) depending on the T1 *spin-lattice relaxation* time of each tissue [5]. In T1-weighted images, the Cerebrospinal Fluid (CSF) is dark, and gray matter appears darker than white matter.

- *T2-weighted*, produced by long TR (≈ 4000 ms) and TE (≈ 90 ms), with image characteristics dictated by T2 *spin-spin relaxation* times [5]. In these images, gray matter appears brighter than white matter, whereas the CSF is bright.

- *Dynamic Contrast-Enhanced Magnetic Resonance Imaging (DCE-MRI)*, injecting a contrast agent with paramagnetic properties (e. g. Gadolinium). It can enhance image quality and analyze vascular structures or lesions, including cancerous ones [6].

- *Diffusion-Weighted Imaging (DWI)* can examine tissue structure on a microscopic scale. By studying the Brownian motion of water molecules, DWI can reveal pathological alterations and physiological details about the brain [7]. Worth mentioning is that the signal characterizing each image volume element (i.e. a *voxel*), at the millimetric resolution, derives from all microscopic movements of water molecules in the considered voxel.
  When applying a magnetic field that varies in space via a gradient determined by the b value measured in s/mm^2, each molecule emits an RF signal with slightly different phases. In a voxel, these randomly distributed phases reflect the trajectory of single molecules. The latter represents the diffusion process, which causes an attenuation of the MRI signal. We can compute the attenuation A as in (1.7), where D is the diffusion coefficient.

$$A = e^{-bD} \tag{1.7}$$

  One currently used measure is the Apparent Diffusion Coefficient (ADC), requiring image acquisition of at least two b values (e.g. 0 and 1000 s/mm^2). The variation between the two acquired signals can be modeled by several functions, but the most frequently used is defined by the following monoexponential equation in 1.8.

$$S_{b_{1000}} = S_{b_0} \cdot e^{-b \cdot ADC} \tag{1.8}$$

  This biomarker can provide quantitative measures of the extracellular fraction and cell density [8].
  Focusing on white matter mapping, *Diffusion Tensor Imaging (DTI)* is widely used to extract different indices sensitive to the displacements of water molecules such as Mean Diffusivity (MD) and Fractional Anisotropy (FA) [9,10]. DTI provides information about diffusion directionality useful to characterize brain tissue architecture [11]. MD maps are particularly informative as they express a quantitative parameter, measured in mm^2/s, corresponding to the mean voxelwise diffusion of water molecules. An increase in MD values can indicate pathophysiological changes observed in neu-

rodegenerative diseases such as Alzheimer's Disease (AD) [12], Parkinson's Disease (PD) [13], and Multiple System Atrophy (MSA) [14].

- *functional Magnetic Resonance Imaging (fMRI)* can give insights into brain functions by using the *Blood-Oxygen-Level-Dependent (BOLD)* signal. This modality considers T2* relaxation to find local changes in hemoglobin concentration. Indeed, hemoglobin presents paramagnetic properties (diamagnetic when oxygenated, paramagnetic when deoxygenated) [15]. When an area of the brain is activated, more oxygenated blood flow is delivered to support this effort. By measuring the ratio between oxygenated and deoxygenated hemoglobin using several acquisitions in time, we can trace the Hemodynamic Response Function (HRF), reflecting the underlying neural activity. Given different tasks (e.g. auditory or visual) alternating with baseline conditions (i.e. resting state), fMRI allows for localization of the activated brain areas, represented in an activation map [16, 17].

For an exhaustive overview of current advances, pitfalls, and clinical applications of MRI, please refer to [11, 18].

1.2 Into the World of Artificial Intelligence

Artificial Intelligence (AI) empowers many aspects of our everyday life in the economy, health, communication, and transportation systems. Its capabilities stem from image recognition to text translation, potentiated by the advent of Deep Learning (DL) [19].
We can define AI as the ability of a machine to try and emulate intelligent behavior through the creation of sophisticated algorithms integrated into electronic devices or computers [20].

The great challenge AI responds to is performing tasks people generally do automatically but find difficult to explain formally, e.g. identifying faces or spoken words [21]. How can machines perform these tasks? They learn from *experience*, in other words, *data*, constructing hierarchical representations, thus freeing human operators from defining the required knowledge. Instead of learning from hard-coded knowledge, AI systems become more efficient when learning their knowledge by extracting patterns directly from raw data. This is the definition of Machine Learning (ML). Building from low- to high-level concepts allows these intelligent systems to understand complex patterns. Representing this procedure with a graph, we might notice it is deep, characterized by several layers of processing units. This approach is known as *deep learning* [21].

It may be challenging to know a priori which is the best set of features suited for the task at

hand. Traditional ML algorithms require a preconceived set of features to function well. For instance, they may receive data to decide whether a patient should undergo surgery. They can infer the relationship between these data and the corresponding outcome, but they cannot change data representation, i.e. the set of features. That may lead to poor performances. *Representation learning* addresses this issue by learning rather rapidly relevant features instead of manually designing them.

Finally, the ultimate goal of AI is to create an Artificial General Intelligence (AGI) and possibly overcome human-level performance [22] by mimicking brain functioning. However, there is still a long way to go.
Fig. 1.4 offers a representation of the principal AI sub-fields. The following sections provide some insights into these different domains.

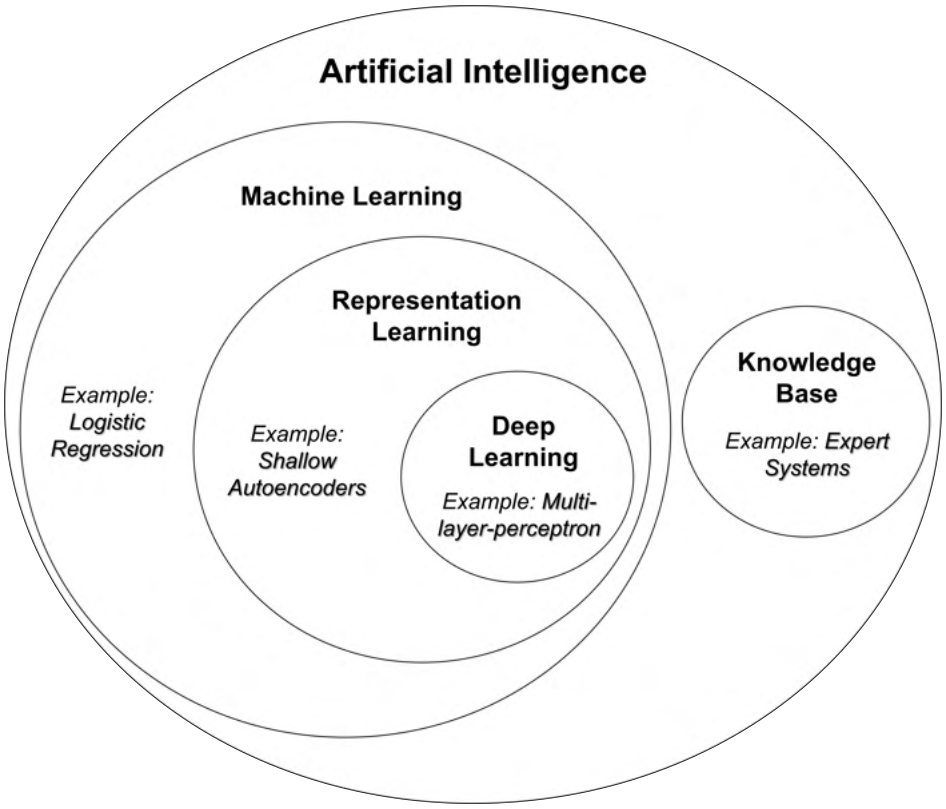


Figure 1.4: Representation of the different AI disciplines. Adapted from [21]

1.2.1 Brief History of AI

Humans have always dreamed of creating a machine able to think and act intelligently. In the 1950s, this did not seem such a utopia after the first robots and computers appeared. However, today this is not yet a reality.

The term "artificial intelligence" was proposed in 1956 by John McCarthy in an act resuming the principles of this newborn field. Indeed, in the summer of the same year, John McCarthy and Marvin Minsky held a conference at Dartmouth College (New Hampshire), gathering other scientists to discuss the emerging domains of cybernetics and informatics [20]. At that time, some researchers thought of intelligent machines solely based on logical rules, e.g. the Expert System (ES). An example is the Logic Theorist, a program able to demonstrate mathematical theorems as a mathematician by using decision trees [23].
Unfortunately, no more funding was granted to pursue AI research until the 1980s. Despite the commercialization of other expert systems, the latter met with little success owing to the difficulty of constraining knowledge in an ensemble of fixed rules. We denote this logic-based current of AI, predominant in the 1970s and 1980s, as Good Old-Fashioned Artificial Intelligence (GOFAI) [20].

In the 1950s, an opposing current to logic-based AI began to arise. It embraced the theories of Donal Hebb, a Canadian psychologist and neurobiologist, who studied the role of neuronal connections in learning. The main idea was to devise machines with a functioning inspired by the human or animal brain, able to train themselves. Inspired by the connections between the biological neurons, these scientists modeled the same architecture into the Artificial Neural Network (ANN), opening a new era in ML. Henceforth, the artificial neuron has become the undisputed protagonist at the core of the ANNs, as the biological neuron constitutes the foundations of the brain.

In 1957, Frank Rosenblatt invented the *perceptron*, the first machine able to learn, inspired by Hebb's cognitive theory [24]. Revolutionary for that time, the perceptron could recognize some forms but remained limited in its capabilities, as composed of a single layer. After the publication of a book by Marvin Minsky and Seymour Papert arguing the limitations of the perceptron [25], the faith in this method completely dropped and culminated with the end of AI research funding from 1969 (a period known as *AI winter*).

Despite these stepbacks, Kunihiko Fukushima proposed the Cognitron in 1975, followed by the Neocognitron in 1981 [26, 27]. His work was motivated by the late advancements in neuroscience carried out by David H. Hubel and Torsen N. Wiesel, who won the Nobel prize in 1981 for their study on the cat's visual cortex [28]. Hubel and Wiesel discovered that every neuron within the primary visual cortex connects to a small part of the visual field, named *receptive field*. These neurons are called *simple cells*. Other neurons in the successive layers, known as *complex cells*, aggregate the information from the previous layers to ensure invariance to small movements of the objects in the visual field and obtain the final image. Following this organization, the Neocognitron was composed of four layers, alternating sim-

ple and complex cells, and a final layer for classification similar to the perceptron [27]. The first four layers were trained without considering the task to solve (i.e. unsupervised learning). The last layer instead specialized in solving the task (i.e. supervised learning).
Even though the Neocognitron was able to recognize simple forms, like numbers, it lacked a learning algorithm to update the parameters of all the layers.

In 1986, the technique based on the backward error propagation proposed by David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams represented the breakthrough needed to train multilayer ANNs efficiently [29]. At the same time, the young scientist Yann Le Cun was working on training multilayer ANNs, when he proposed his algorithm named Hierarchical Learning Machine (HLM). The latter consisted in propagating some desired states for each neuron instead of propagating gradients backwards [30]. This trick was useful to overcome the computational limitations of the time.

In 1988, Yann le Cun proposed the first Convolutional Neural Network (CNN) with only four layers for recognizing written characters, always inspired by the structure of the visual cortex. LeNet5, a later version of the first CNN, was commercialized to automatically read between 10% and 20% of the deposited checks in the United States [20].
Regardless of this success, another gloomy period began in 1995 for ANNs and lasted for about 15 years. CNNs were considered too complex and demanding from a computational point of view. Between 1992 and 1995, the Support Vector Machine (SVM), developed by Isabelle Guyon, Vladimir Vapnik, and Bernhard Boser, became the preferred ML method [31].

Continuous efforts and unremitting trust in ANN's potential finally paid off in 2012, when the revolution brought by deep learning was undeniable. Indeed, training deep networks became feasible, thanks to the massive employment of the Graphical Processing Unit (GPU) coupled with large data sets available after the internet outbreak.
Since 2010, the ImageNet competition has brought together researchers to solve image recognition tasks [32]. In 2012, Geoffrey Hinton and his students scored an error of 16%, outperforming the best performance of the previous year with an error equal to 25% [33]. They used a deep CNN trained with GPUs (see Section 1.2.4.2.3 for more information). In the following years, all participants experimented with variants of the same method to outperform this unbelievable performance.
Henceforth, the scientific community has turned its attention to deep learning, which proved its value in several tasks and today permeates our everyday life.

Fig. 1.5 provides a timeline of AI history with the most remarkable events.

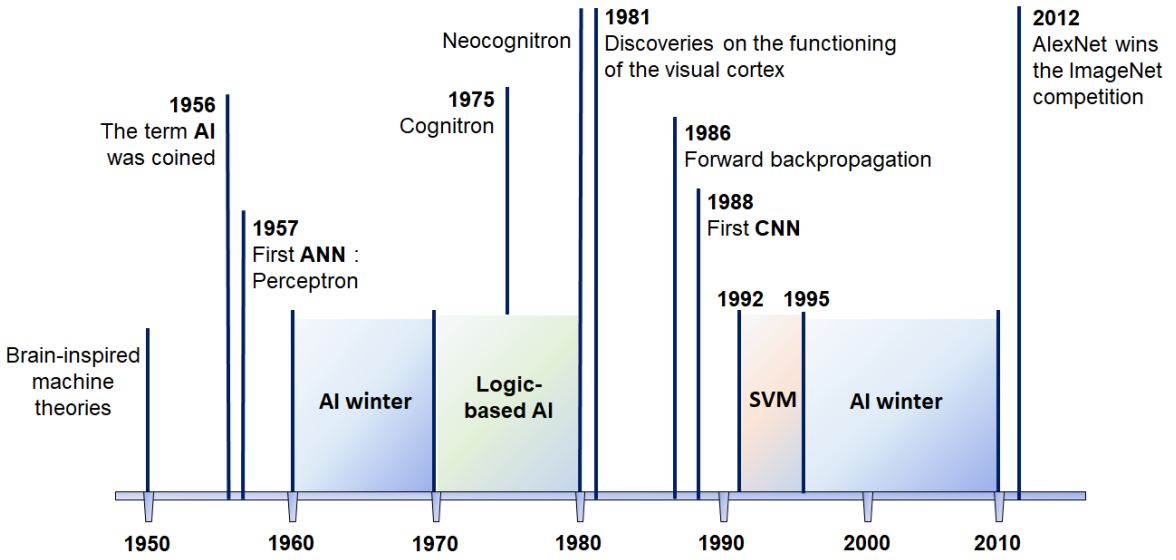


Figure 1.5: Most significant events in the history of artificial intelligence. AI: Artificial Intelligence; ANN: Artificial Neural Network; CNN: Convolutional Neural Network; SVM: Support Vector Machine

1.2.2 Expert Systems

Expert systems are computer programs capable of reasoning. They comprise a set of if-then rules established and inspired by a human expert [34].
An expert system typically consists of the following components [35]:

- *Knowledge Base (KB).* It encloses the knowledge of the system in the form of rules, basic facts, and heuristics;
- *Inference engine.* It allows the inference of new information by using the KB through a reasoning method;
- *Explanation facility.* It explains the decision-making process of the ES.

Despite the evident advantages such as clear explanations and solid expertise, expert systems remain limited in their capacity. Unlike human experts, they cannot reason outside their inference engine, thus failing to solve unseen tasks [36].
Further details about ES are available in [37, 38].

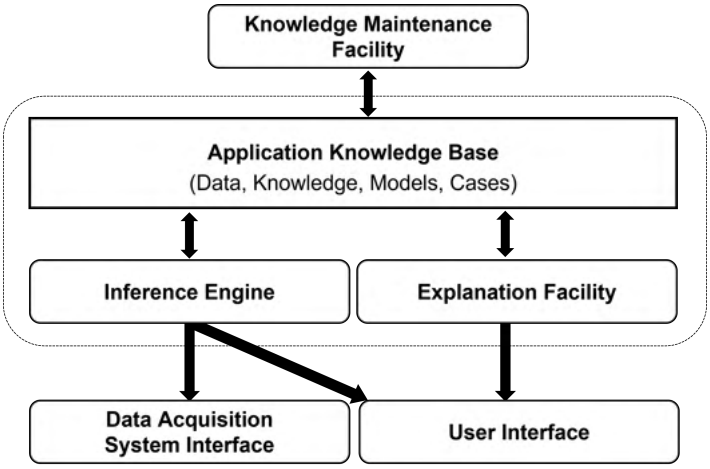


Figure 1.6: Architecture of an expert system. Adapted from [35]

1.2.3 Machine Learning

Machine learning comprises algorithms capable of learning from data [21]. The latter are usually composed of several examples, each represented by an ensemble of *features*, i.e. specific characteristics extracted from data.

We can understand what *learning* means in this context by quoting the definition provided by Tom Mitchell in 1997 [39]: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." As previously mentioned, we should evaluate model performance on different data than those used for training. Based on the knowledge gathered from the training data, the machine should successfully perform the same task on unseen data. This generalization ability is undoubtedly the most crucial ability a machine may develop.
We can list two main types of ML algorithms, *supervised* and *unsupervised*, introduced in the following sections.

1.2.3.1 Supervised Learning

Supervised algorithms require labeled data, with each example associated with a *class* or *label*, i.e. a category. For instance, one famous labeled dataset is the Iris dataset containing precise measurements regarding 150 iris plants [40].
Among the various supervised tasks ML can perform, we can find [21]:

- *Classification*. The goal of the machine consists in specifying to which of the c categories the input belongs. To this end, the learning algorithm should devise a function

as in (1.9).

$$g : \mathbb{R}^n \rightarrow \{1, ..., c\} \tag{1.9}$$

A category h is assigned by the function g to the input vector $\boldsymbol{i}$ as in (1.10).

$$h = g(\mathbf{i}) \tag{1.10}$$

An example of classification tasks is determining if a subject is affected by Parkinson's disease or is instead healthy [41].

- *Regression*. Similar to classification, the machine should produce an output which is a numerical value, provided some input. For instance, predicting a subject's age using brain MRI data represents a regression task [42].

In supervised learning, algorithms are supposed to search for the relationship between each label and the corresponding examples. Let us briefly describe two emblematic algorithms to understand how supervised learning works.

1.2.3.1.1 Linear Regression

To see what an ML algorithm can do, we briefly present *linear regression*, used to solve regression problems through a linear function of the input [21].
Equation (1.11) describes the predicted value $\hat{\mathrm{r}} \in \mathbb{R}^n$ as obtained by the input vector $\mathbf{v} \in \mathbb{R}^n$ weighted by the transpose of the vector of parameters $\mathbf{k} \in \mathbb{R}^n$.

$$\hat{r} = \mathbf{k}^\top \mathbf{v} \tag{1.11}$$

The linear combination between each feature and the corresponding entry of $\mathbf{v}$ determines the output. We can interpret each parameter as a *weight*, representing how much it can affect the prediction.
We evaluate the performance on a set of data left out from training, called *test set*, defining $\mathbf{V}^{te}$ as the inputs associated with $\mathbf{r}^{te}$, the regression targets.
To assess model performance on the test set, we can compute the Mean Squared Error (MSE) in (1.12), denoting by $\hat{r}^{te}$, the predicted regression value.

$$MSE_{te} = \frac{1}{n} \sum_j (\hat{\mathbf{r}}^{te} - \mathbf{r}^{te})_j^2 \tag{1.12}$$

One possible approach to solve this task is to let the algorithm gain experience using the training set ($\mathbf{V}^{tr}$, $\hat{r}^{tr}$) by minimizing the error MSE_{tr} solving for where the gradient is equal

to $\mathbf{0}$ and obtaining $\mathbf{k}$ as in (1.13) (refer to [21] for the complete proof).

$$\mathbf{k} = (\mathbf{V}^{tr\top}\mathbf{V}^{tr})^{-1}\mathbf{V}^{tr\top}\mathbf{r}^{tr} \tag{1.13}$$

A more exhaustive model for linear regression usually comprises a bias term b as detailed in (1.14). In this version, the function maps the prediction from features with an affine transformation.

$$\hat{r} = \mathbf{k}^{\top}\mathbf{v} + b \tag{1.14}$$

One variant of linear regression is *logistic regression*, which instead performs classification and outputs the probability of belonging to a class.

1.2.3.1.2 Support Vector Machine

As one of the most representative and used algorithms in supervised ML, SVM allows for binary classification. It comprises a linear function outputting the sample class [31]. Mapping the non-linear input vectors into a high-dimensional feature space makes them linearly separable. Therefore, the training goal is to find the best hyperplane to split the two classes. This hyperplane should present the furthest distance from the nearest training samples named *support vectors*.

One innovative advance of SVM is the *kernel trick*. It states that several ML algorithms can be defined by using dot products between examples [21]. Let us consider the linear function used by the SVM in analogy with logistic regression:

$$\mathbf{k}^{\top}\mathbf{v} + b = b + \sum_{j=1}^{N}(\alpha_j \mathbf{v}^{tr\top}\mathbf{v}^{(j)}), \tag{1.15}$$

where $\mathbf{v}^{(j)}$ represents a training example, and α is the vector of coefficients. We can substitute $\mathbf{v}$ with the output of a given feature function $\eta(\mathbf{v})$ and the dot product with a function called *kernel* defined as $k(\mathbf{v}, \mathbf{v}^j) = \eta(\mathbf{v}) \cdot \eta(\mathbf{v}^j)$, with $\cdot$ being an inner product equivalent to $\eta(\mathbf{v})^{\top}\eta(\mathbf{v}^j)$. The function which makes the predictions is defined in (1.16).

$$g(\mathbf{v}) = b + \sum_{j=1} \alpha_j k(\mathbf{v}, \mathbf{v}^{(j)}) \tag{1.16}$$

The relationship between α and $g(\mathbf{v})$ as well as $\eta(\mathbf{v})$ and $g(\mathbf{v})$ is linear, although the function in (1.16) is nonlinear with respect to $\mathbf{v}$.

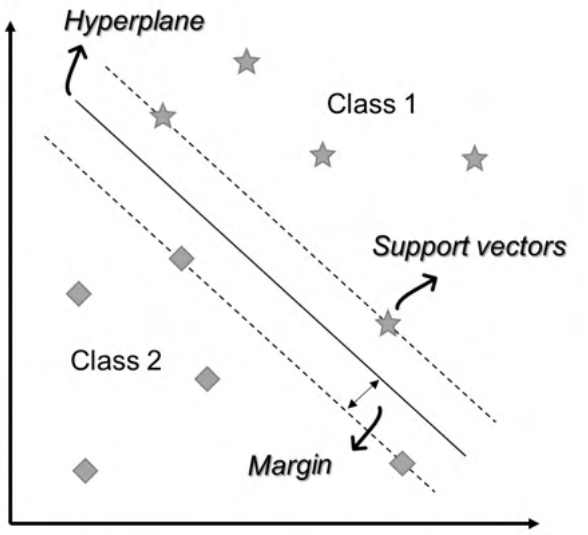


Figure 1.7: Representation of the main components of an SVM

1.2.3.2 Unsupervised Learning

Unsupervised algorithms can learn data structure based on some features. Clustering techniques, such as k-means or hierarchical clustering, aim to divide data into groups with similar characteristics [43, 44].
Unsupervised learning techniques can be divided into two main categories, as illustrated in Fig. 1.8:

- *Partitional* or *flat.* They split data into disjoint clusters, as do k-means and self-organizing maps.
- *Hierarchical.* They produce a hierarchy of nested clusters, such as AGNES (AGglomerative NESting) or DIANA (DIvisive ANAlysis Clustering).

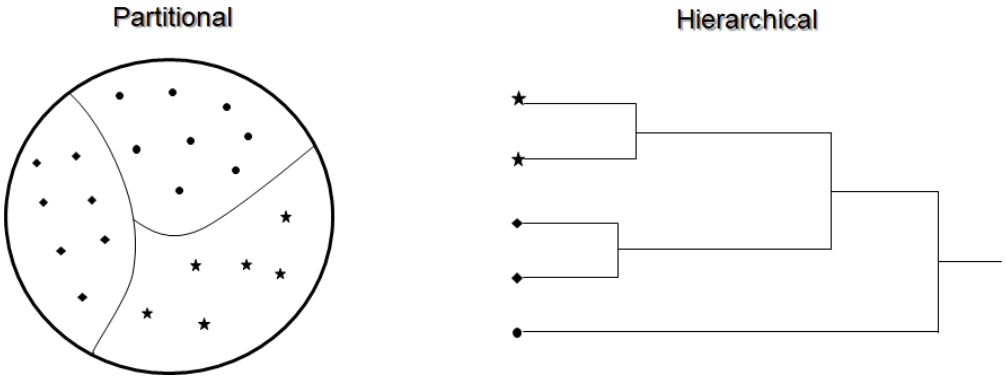


Figure 1.8: Representation of the two categories of unsupervised learning algorithms

1.2.3.2.1 K-Means

K-means aims at minimizing intracluster variability and maximizing intercluster distance with an iterative procedure [43]. We can characterize a typical k-means algorithm with the following:

- *Centroid* (or *prototype*), mean value of the elements belonging to a cluster;
- *Intracluster variability*, defined as the sum of the distance between each element of the cluster and the cluster centroid:
- *Intercluster distance*, obtained as the distance between two centroids.

One disadvantage of this clustering technique is that the user must define the number of clusters k a priori. According to the problem and the number of samples, it is customary to examine different k and choose the most appropriate. There exist some techniques to help find the optimal k, such as:

- *Elbow Method.* It considers the sum of squared distances S of each sample to the closest centroid. The point at which S starts to decrease indicates the best k.
- *Silhouette Coefficient*. It evaluates the similarity between a sample and its cluster compared to the other clusters [45]. The most appropriate k is the one with the highest silhouette coefficient in the range $[0,\ 1]$.

Unfortunately, there is no agreement concerning the best approach to choose k, and the results obtained by the different techniques can vary.
To perform k-means, we can proceed as follows:

1. Choose the number of clusters k;
2. Initialize cluster centroids (e.g. by assigning samples randomly to each cluster or considering a sample as a cluster centroid);
3. Compute cluster centroids;
4. Assign elements to clusters according to the lowest distance measure;
5. Iterate from 3) until a stopping condition is met (e.g. maximum number of iterations or no element changed cluster compared to the previous iteration).

Some extensions to k-means allow for determining k automatically, such as ISODATA [46].

1.2.3.2.2 Hierarchical clustering

Hierarchical clustering creates a tree of nested clusters. First, we need to define a distance measure, such as the euclidean distance, to compute the similarity between clusters. Then, we can group samples according to the lowest similarity measure until we arrive at the partition corresponding to the initial data. If two elements n_1 and n_2 are grouped together at level l_0, they remain together at higher levels too (l_1, l_2, and so on). That is why this type of clustering is called hierarchical. As shown in Fig. 1.9, according to the *cut-point*, we can choose to favor:

- *Intracluster variability*, decreasing with an increasing number of clusters:
- *Intercluster distance*, increasing with decreasing number of clusters.

Another factor is the number of elements in each cluster, which can vary from one to the total. One-sample clusters may identify *outliers*, i.e. samples significantly differing from the data distribution.
This technique may give insights about data structure, despite some drawbacks, such as being highly affected by distance metrics or the difficulty in choosing the number of clusters most suited for the task at hand.
Nested clusters can be represented with a dendrogram, i.e. a tree diagram, as in Fig. 1.9.

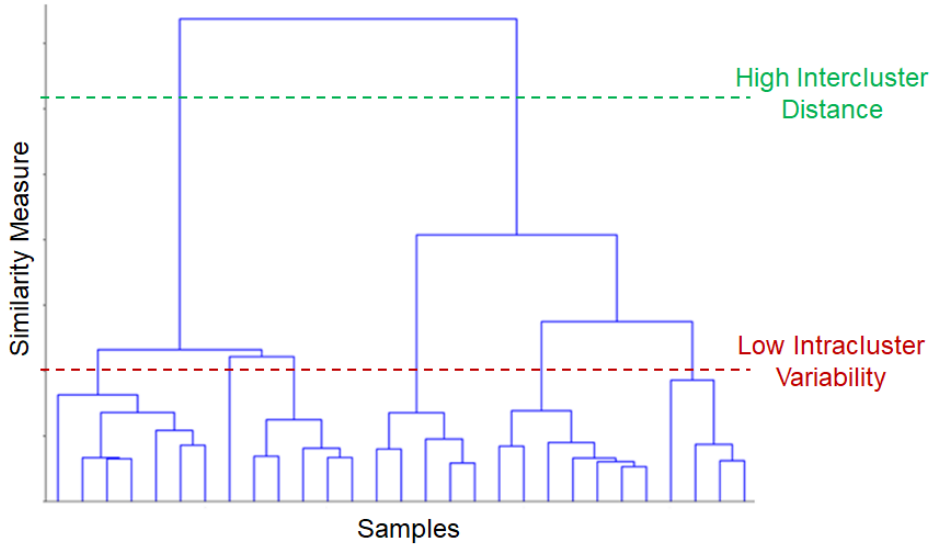


Figure 1.9: Example of a dendrogram. Choosing the cut-point at two clusters maximizes intercluster distance. The higher the number of clusters, the lower the resulting intracluster variability

1.2.3.3 Key Concepts

Capacity As previously mentioned, the ultimate goal of an ML algorithm is to generalize, in other words, perform well on unseen data.
Although we can train an algorithm without using the test (or generalization) set, we can rely on the *i.i.d. assumptions* to be confident about the final generalization performance. According to these assumptions:

- Each example in the train and test sets is independent from the others;
- Train and test sets belong to the same probability distribution.

The generalization ability of ML algorithms can be measured by looking at the training error and the difference between training and test error, hopefully as small as possible. These two evaluations help establish the *underfitting* or *overfitting* conditions.
A model underfits when the training error is not sufficiently low, whereas it overfits when the difference between training and test error is considerable, with the test error usually higher than the training error.

Model capacity is the ability to fit different types of functions. If the capacity is low, the model is more likely to underfit. If the capacity is high, the model is more exposed to memorization of irrelevant characteristics from the training data, thus resulting in overfitting. A concrete example illustrates these concepts in Fig. 1.10, whereas Fig. 1.11 presents the relation between capacity and error.

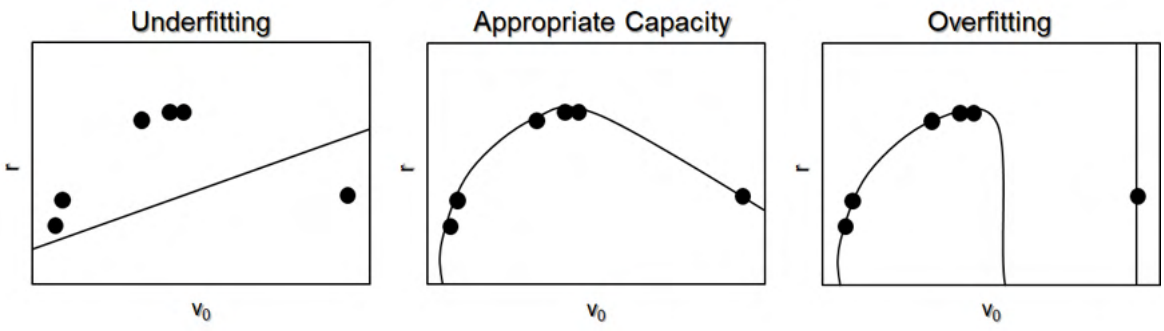


Figure 1.10: *Left*. An example of underfitting with a linear function unable to capture data structure. *Middle*. A function with the appropriate capacity would perform well on new points. *Right*. An example of overfitting: the function passes through all the points without finding their underlying relationship.
Adapted from [21]

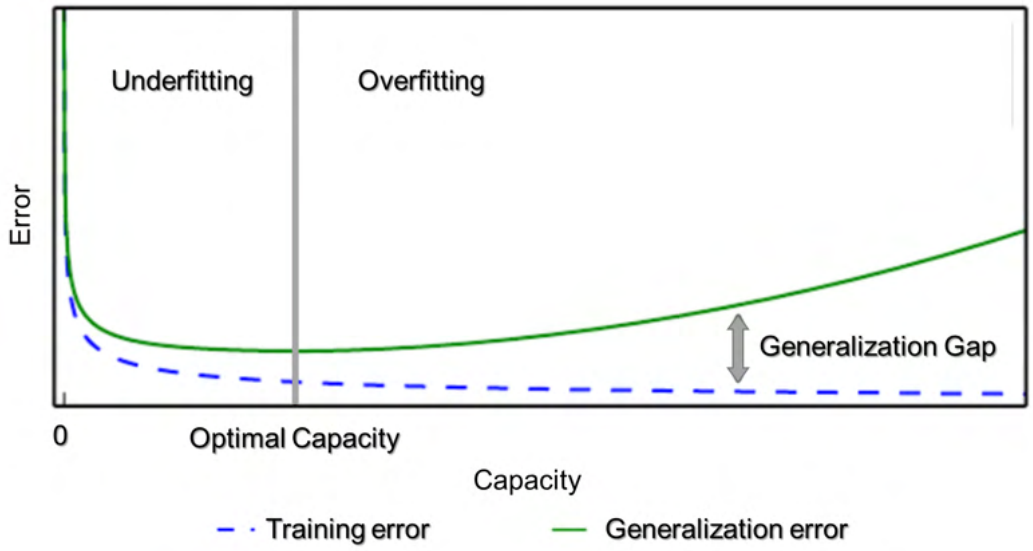


Figure 1.11: Capacity is represented as a function of the error, indicating areas of underfitting and overfitting. Adapted from [21]

However, model capacity depends on aspects other than the number of features or the particular family of functions. We cannot often find the best possible function, but we can settle for one that diminishes the training error. That is the *representational capacity* [21].
We must emphasize a crucial consideration about generalization. Although the fact of inferring general rules from a set of limited samples may seem quite illogical as not supported by all the necessary information, ML wishes to create models that are, quoting from [21], "*probably* correct about *most* members of the set they concern".
The *no free lunch theorem* for ML maintains that every classification algorithm will present the same error rate on new data considering the mean over all possible data generating distributions [47]. However, in practice, we cannot access all data distributions, but we deal with particular kinds, interesting for real-world applications.

Regularization Although the no free lunch theorem ensures no absolute best ML model, we can still try to optimize performances by considering the task at hand. To guide the learning algorithm in the right direction, we could modify the functions it uses and make it prefer some particular solution according to the best fit to training data.
Regularizing a model means modifying its learning algorithm to decrease its generalization error without degrading the performance of the training data. Many forms of regularization have been devised to address specific problems, always keeping in mind the principle of the no free lunch theorem [21]. We discuss regularization techniques in Section 1.2.4.1.3.

Hyperparameters Each model comprises a variable number of parameters determining its structure and behavior, known as *hyperparameters*. The model does not learn hyperparameters, so we must define them a priori.
A separate set from training data, called *validation set*, can be used to establish hyperparameters and track model performance to avoid overfitting. Reasonable data splitting percentages are 80% for training and 20% for validation.
Choosing hyperparameters on training data would result in maximum model capacity and, consequently, overfitting.

Cross Validation If the dataset is small (with a few hundred samples or less), an alternative is to perform k-fold Cross Validation (CV): data are split into k subsets such that k-1 parts serve for training and the remaining one for testing. That allows for averaging the error across k splits, thus giving a hint about model stability regardless of training data.
Fig. 1.12 illustrates a practical example for a 10-fold CV.

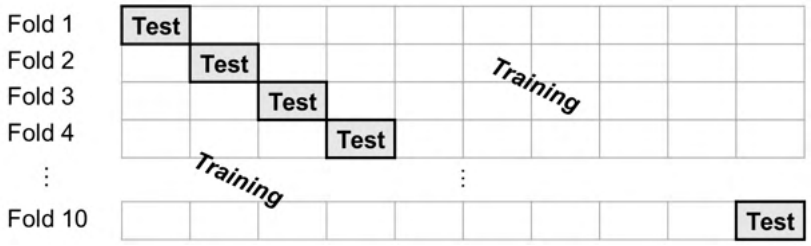


Figure 1.12: Example of data set split for 10-fold cross-validation. Each sample will be used at least once for training and testing the model. Averaging performances across folds can tell us about model stability

Bias and Variance When the model is prone to perform systematic errors, ignoring some aspects of the data, it is said to have a high *bias*. When errors do not present any particular structures and small changes in the data significantly influence the model, the latter is unstable and characterized by high *variance*. Fig. 1.13 provides a representation of these concepts.

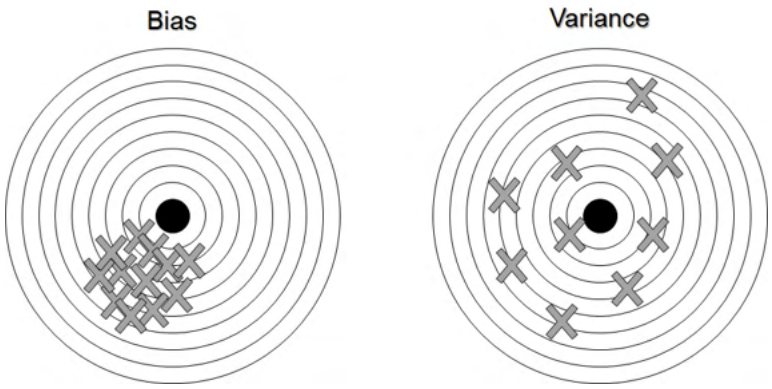


Figure 1.13: When the model presents a high bias, it is affected by systematic errors. In the case of a model with high variance, it is unstable and does not capture the structure of the data

According to a more formal definition, bias expresses the deviation from the true value of the function parameter. Variance refers to the deviation from the expected estimator value according to the considered sampling of data [21]. These concepts are tightly related to overfitting and underfitting: low capacity causes high bias, whereas variance tends to increase with high capacity (see Fig. 1.14).

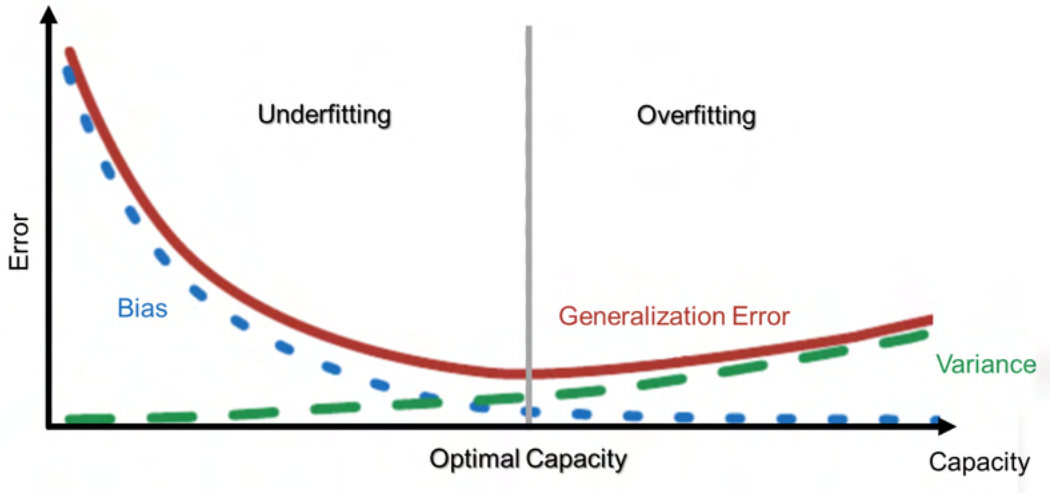


Figure 1.14: Bias and variance as a function of model capacity. Adapted from [21]

1.2.3.4 Performance Evaluation

According to the task, we can adopt different metrics to evaluate and compare the performance of an ML algorithm.
For binary classification, we can use the *confusion matrix*, defining four categories:

- *True Positive (TP)*, samples correctly identified as the positive class;

- *True Negative (TN)*, samples correctly identified as the negative class;
- *False Positive (FP)*, negative samples wrongly predicted as positive;
- *False Negative (FN)*, positive samples wrongly predicted as negative.

For instance, we can associate the negative class with healthy subjects and the positive class with patients.
Based on the confusion matrix illustrated in Fig. 1.15, we can compute the following metrics:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1.17}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{1.18}$$

$$Specificity = \frac{TN}{TN + FP} \tag{1.19}$$

$$PPV = \frac{TP}{TP + FP} \tag{1.20}$$

$$NPV = \frac{TN}{TN + FN} \tag{1.21}$$

The confusion matrix may be easily extended to multiclass problems as well.

Confusion Matrix		True Class	
		Negative	Positive
Predicted Class	Negative	TN	FN
	Positive	FP	TP

Figure 1.15: Confusion matrix for binary classification problems

In the case of imbalanced classes, we can compute the balanced accuracy as follows, to account for the different number of samples per class:

$$Balanced\ Accuracy = \frac{Sens + Spec}{2} \tag{1.22}$$

For the sake of brevity, we will refer to accuracy or balanced accuracy indistinctly according to the considered number of samples per class.

1.2.4 Deep Learning

No matter how powerful ML methods, such as SVM, are when analyzing a small data set with complex relationships, they still present some limitations. For instance, they can be sensitive to data dimension, preventing them from efficiently processing multidimensional data [20], an issue known as the *curse of dimensionality* [21]. Another issue is the difficulty of manually extracting a set of features relevant to a given task, a process that is called *feature engineering*. Indeed, this depends on the complexity of the task at hand. For instance, considering object recognition, we should compute several features independent of position, distortions, and shifts. Imagine the quantity of time and effort this takes with no guarantee of success. That is where *representation learning* comes into play: it can extract meaningful features from raw data without human intervention. Deep learning falls into this category, allowing for feature extraction and successive classification with an ANN. Its name derives from using several successive modules, each with simple yet non-linear functions with higher and higher degrees of abstraction.

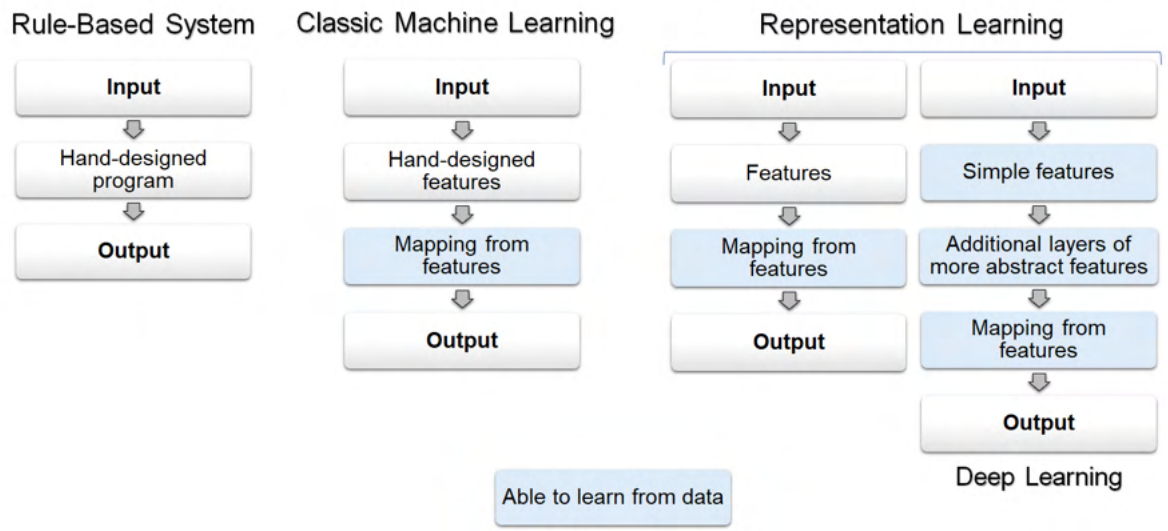


Figure 1.16: Comparison between AI systems. Shaded boxes highlight modules that learn from data. Adapted from [21]

Fig. 1.16 provides a comparison between the different AI systems. We can see that deep learning hierarchically performs feature extraction, from very simple to more abstract levels, automatically learned by the machine to optimize task resolution. By contrast, classic machine learning demands hand-crafted feature extraction, somehow limiting or biasing the successive mapping depending on the feature choice and informative content.
Central to deep learning are artificial neural networks in all their forms. In the following, we

provide an overview of their conception and evolution.

1.2.4.1 Feedforward Neural Networks

The discoveries in neuroscience related to the functioning of the brain and human intelligence opened the way for the creation of artificial neural networks [20].
In the 1950s, neuroscientists focused their attention on the way neurons communicate with each other. A biological neuron is characterized by different branches, the *dendrites*, allowing connections to other neurons. The emitted electric signals are transmitted through contact areas, called *synapses*. After being processed in the cellular body, the emitted signals pass to the neighboring neurons through the *axon*. This output represents an ensemble of electric signals (i.e. action potentials or *spikes*) with a frequency expressing neuron activity.

The first mathematical model of a biological neuron was devised in 1943 by Warren McCulloch and Walter Pitts, two American neuroscientists and cybernetics specialists. This simplified version of the biological neuron is called an *artificial neuron*. It performs a weighted sum of the received inputs (e.g. resulting from the activity of neighboring neurons) and produces a numerical value. Then, it compares this output to a threshold: if the value is inferior, the neuron is inactive, otherwise it is active and the signal propagates through the axon to the downstream neurons.
According to the model of McCulloch and Pitts, we can see the brain as a logic inference machine, in which binary neurons perform logical computations [48]. Fig. 1.17 highlights the similarities between biological and artificial neuron. Inspired by the artificial neuron, Frank Rosenblatt proposed the first machine able to learn in 1957, knows as *perceptron*, and composed of a single artificial neuron. His work was motivated by the theories of Donald Hebb, who suggested that connections between two simultaneously active neurons are strengthened [49]. Hebbian learning was confirmed in the 1960s and exhaustively explained by Eric Kandel in the 1970s.

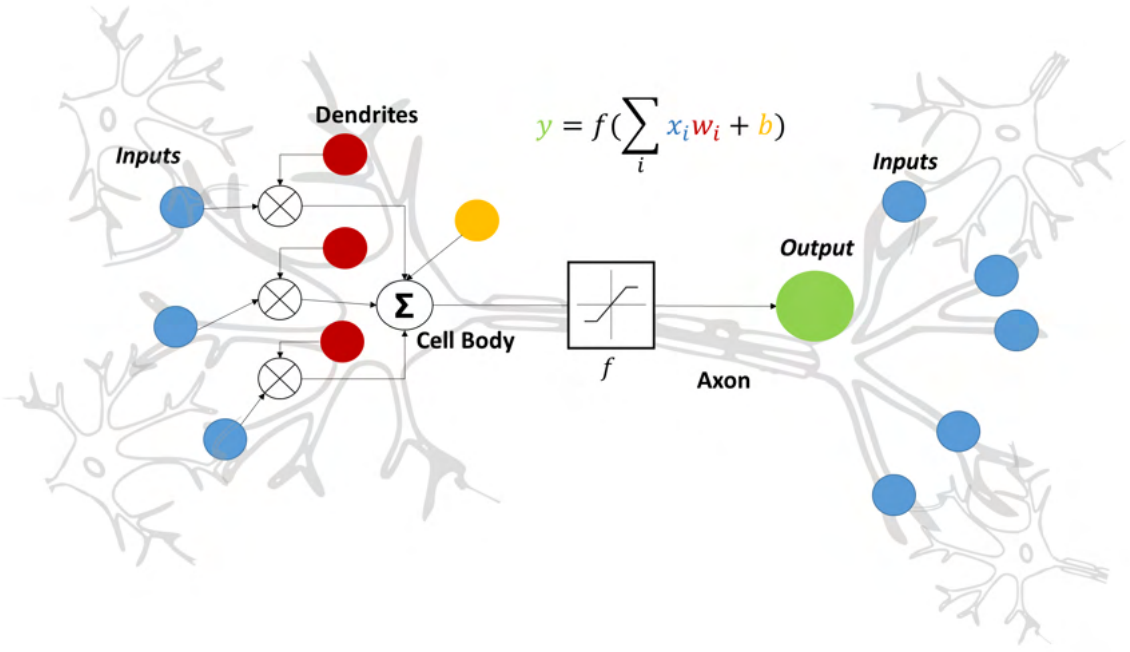


Figure 1.17: Comparison between a biological and an artificial neuron. In biological neurons, inputs are processed in the cell body and transmitted to neighboring neurons through the axon. Similarly, in artificial neurons, the weighted sum between the inputs x and the weights w adds to a bias term b, then passes through an activation function f to produce an output y

Rosenblatt introduced a way for the neuron to learn by modifying its synaptic connections (i.e. the weights) based on the errors between predicted and expected values. This idea was in line with the studies about the synaptic efficiency of Santiago Ramon y Cajal, dating back to the nineteenth century. The concept of adjusting model parameters according to the predictions regarding input data was not novel in statistics, yet it had not been applied to pattern recognition.
The perceptron can be interpreted as a linear classifier that divides its input space into two parts. However, if the input is not linearly separable, it cannot be described with a linear combination of the weights. That happens when considering pattern recognition in one image: if the goal is to recognize some shape, the slightest modification of orientation, position, or dimension could cause the model to fail, as each weight connects to one pixel.

For pattern recognition, one solution was the introduction of an intermediate module, called a *feature extractor*, between the input and the classifier itself. The latter is in charge of detecting specific traits in the image to describe it in a meaningful yet more synthetic way [20]. Let us consider face recognition as an example. In this case, discriminating traits can be the nose, the eyes, or the lips of a person, and we wish to obtain features carrying this

information. We use a vector fed to the classifier to encode the feature presence, absence, or intensity. However, when feature extractors are hand-designed, their development can be very complex and time-consuming.

The discriminative power of perceptrons grew with their multi-layer version, composed by several *hidden* layers of artificial neurons, also called *units*. A Multilayer Perceptron (MLP) was indeed able to analyze even nonlinear relationships between input and outputs. Fig. 1.18 shows a typical architecture. Each unit performs a weighted sum of the inputs, which is then passed through an activation function and transferred to the successive units, fully connected. This mode of functioning is called *feedforward*, as each layer takes the resulting states of preceding layers as input.
If the number of units, and thus of layers, is considerably high (usually more than five layers), we call these algorithms *deep neural networks*.

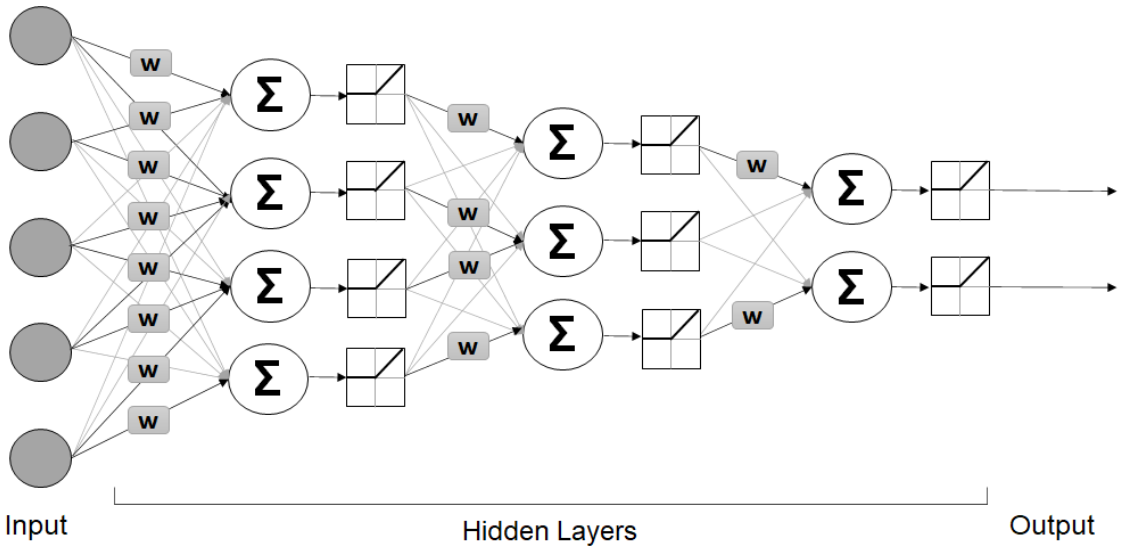


Figure 1.18: Typical structure of a feedforward multilayer perceptron. Note the fully connected scheme between units. For clarity's sake, only few connections are shown. Adapted from [20]

There are two main types of layers in MLPs:

- *Linear*, performing the weighted sum between inputs and weights;
- *Nonlinear*, applying a nonlinear function to the output of each unit. Some examples are available in Fig. 1.19.

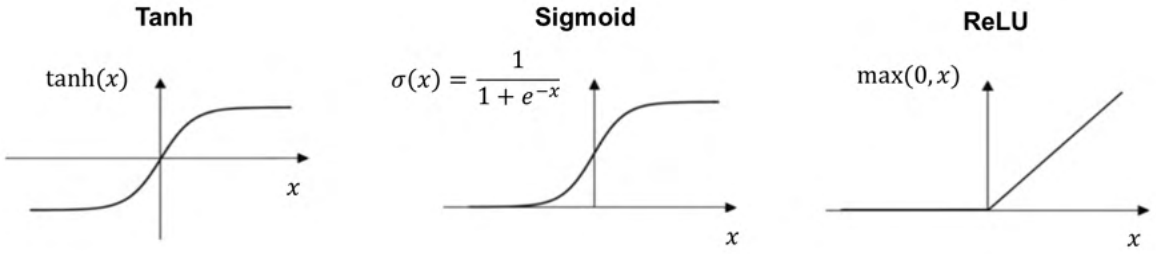


Figure 1.19: Examples of activation functions, used to introduce nonlinearity in feedforward MLPs

However, feature extraction was still performed manually due to the difficulty of efficiently adjusting the weights of several layers. That required high computational power, which the hardware of the time could not provide.
Although the first attempts to completely automatize the feature extraction process led to successful results in the 1980s, we must wait until 2012 for the scientific community to embrace the method revolutionizing image recognition, known as CNN. The latter performs feature extraction by automatically *learning* features during training, so optimized to solve the considered task. At the heart of the present work, we describe CNNs thoroughly in Section 1.2.4.2. Other DL algorithms are gaining considerable attention such as Generative Adversarial Network (GAN) for synthetic data generation [50], and Recurrent Neural Network (RNN) able to deal with sequential data and especially used for text translation.

Let us now focus on *feedforward neural networks*, also known as MLPs, considered the core of deep learning. *Feedforward* refers to the information flowing from the input $\mathbf{x}$ to the output y, without any feedback connections, implemented when the output is reintroduced into the model. Feedback connections are characteristic of RNNs [51].
The goal of an MLP is to approximate some function g to produce a mapping $y = g(\mathbf{x}, \boldsymbol{\theta})$, with $\boldsymbol{\theta}$ the set of parameters learned to optimize the function.

Besides the input layer receiving input data, MLPs are generally composed of a variable number of *hidden layers*, each characterized by a specific function, whereas the final layer is called the *output layer*. The number of layers determines the *depth* of the model, whereas the number of units (i.e. the artificial neurons) in each hidden layer determines its *width*. It is important to note that the relationship between training examples and their output specifies the result expected from the network: a value very close to the output y. On the other hand, the learning algorithm establishes the relationships between hidden layers with no a priori [21]. To find nonlinear relationships, it is the model that learns nonlinear functions, without the need for the user to design them, as before the advent of DL. That is the strategy

adopted for the hidden layers and is equivalent to applying a nonlinear transformation to the input of a linear model, allowing for extending the capacity of linear models.

1.2.4.1.1 Hidden and Output Units

Hidden units are responsible for the non-linearity of neural networks. They usually consist of a nonlinear function q applied to an affine transformation $\mathbf{a}$ of the input vector $\mathbf{x}$, multiplied by the weight matrix transposed $\mathbf{W}^\top$ and summed to a bias term b, as in (1.23) and (1.24).

$$\mathbf{h} = q(\mathbf{a}) \tag{1.23}$$

$$\mathbf{a} = \mathbf{W}^\top \mathbf{x} + b \tag{1.24}$$

Any hidden unit may be an output unit whose choice influences the type of loss function, i.e. the function to be optimized (please refer to Section 1.2.4.1.5 for more details) [21].
In the following, we mention some well-known units with their related activation function.

- *Softmax*. It initiates a competition between units since each expresses a class probability. This behavior resembles the winner-take-all neuron associated with the lateral inhibition hypothesized in the cortex for neighboring neurons [21]. Similarly, in MLPs one unit presents the highest probability and establishes the winning class.
  Softmax activation $\sigma : \mathbb{R}^C \rightarrow (0,1)^C$ is defined as in (1.25), with C the number of classes and $\mathbf{x}$ the input vector. The total sum over the components must add up to 1.

$$\sigma(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{j=1}^{C} e^{x_j}} \tag{1.25}$$

  We widely employ Softmax as an output unit.

- *Sigmoid*. Used extensively in recurrent networks, it is defined in (1.26). One drawback is that saturation can occur for a consistent part of its domain [21].

$$q(x) = \frac{1}{1 + e^{-x}} \tag{1.26}$$

- *Hyperbolic tangent*. Closely related to the sigmoid unit σ, it is defined as follows:

$$q(x) = tanh(x) = 2\sigma(2x) - 1 \tag{1.27}$$

  Fig. 1.19 represents sigmoid and hyperbolic activation functions.

- *Rectified Linear Unit (ReLU).* Given its similarity to a linear activation except for zeroing half of its domain, ReLU is rather easy to implement, with large but consistent gradients. We can characterize it with the formula in (1.28).

$$q(x) = max\{0, x\} \tag{1.28}$$

ReLU activation presents advantages such as computational efficiency, better convergence performance, and reduced vanishing gradient problems compared to more complex units like sigmoid in (1.26) and hyperbolic tangent in (1.27).
However, if a considerable number of units reaches zero activation, this would cause learning to stop. For this reason, variants like Leaky ReLU in (1.29) present certain flexibility for negative numbers.

$$q(x) = \begin{cases} x, & \text{if } x > 0, \\ 0.01x, & \text{otherwise} \end{cases} \tag{1.29}$$

Another possibility is the Exponential Linear Unit (ELU), defined in (1.30), in which $\alpha > 0$. ELU has shown higher classification accuracy than ReLU [52].

$$q(x) = \begin{cases} x, & \text{if } x > 0, \\ \alpha(e^x - 1), & \text{otherwise} \end{cases} \tag{1.30}$$

Fig. 1.20 offers an illustration of ReLU and its variants.

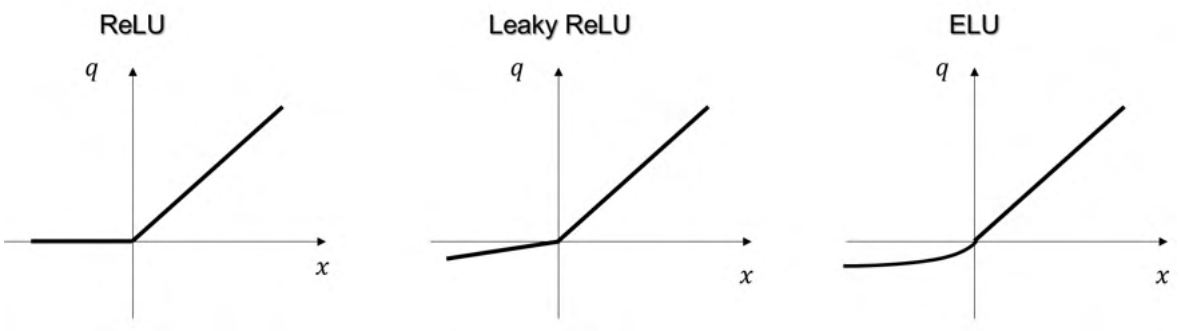


Figure 1.20: Variants of the Rectified Linear Unit (ReLU)

1.2.4.1.2 Back-Propagation

Forward propagation defines the flow of information forward through the network from the input to the output [21]. This behavior can continue until the computation of a scalar cost.

Back-propagation, or *backprop* reintroduces the information from the cost backward into the flow to allow for gradient computation [29]. The learning algorithm for ANNs consists of backprop for computing the gradient and another algorithm for learning via this gradient, e.g. Stochastic Gradient Descent (SGD) (see Section 1.2.4.1.5).
To compute backprop, we can apply the chain rule to tensors in the case of MLPs. The chain rule describes the derivative of the composition of two differentiable functions, f, and g. If we consider $h(x) = f(g(x))$, then the chain rule in Lagrange's notation can be expressed as in (1.31).

$$h'(x) = f'(g(x))g'(x) \tag{1.31}$$

Table 1.1 presents an example of backprop for an MLP with two hidden layers. In the forward pass, the first hidden layer receives as input the weighted sum between weights w and inputs x (we omit bias for simplicity) passed through a nonlinear activation function a, and the same goes for the successive layers. Let us consider a cost function equal to $\sqrt{y_l - t_l}^2$, in which t_l is the true value and y_l is the prediction. In the backward pass, we compute the weighted sum of the error derivatives with respect to the inputs of the units in the preceding layer [53].

1.2.4.1.3 Regularization Techniques

Parameter Norm Penalties Regularization can be achieved by adding a norm penalty term to the cost function that usually affects only the weights, leaving the biases unaltered [21]. This penalty term is multiplied by a nonnegative hyperparameter $\alpha \in [0, \infty]$, controlling the strength of the regularization φ. Equation (1.32) formally defines the regularization $\tilde{C}$ of the cost function C given the model parameters $\boldsymbol{\theta}$, the input tensor $\mathbf{X}$, and the output vector $\mathbf{y}$.

$$\tilde{C}(\boldsymbol{\theta}; \mathbf{X}, \mathbf{y}) = C(\boldsymbol{\theta}; \mathbf{X}, \mathbf{y}) + \alpha\varphi(\boldsymbol{\theta}) \tag{1.32}$$

We can list two commonly used types of parameter norm penalty:

- *Weight decay.* Also known as L^2 *regularization* or *ridge regression*, it acts on the vector of weights $\mathbf{w}$ by driving them towards the origin and adding to the cost function the term $\frac{1}{2}\|\mathbf{w}\|_2^2$.
- L^1 *regularization.* This regularization strategy considers the sum of the absolute values of the weights $\|\mathbf{w}\|^1 = \sum_i |w_i|$. Compared to weight decay, L^1 regularization leads to more sparse solutions. We can exploit this behavior to perform *feature selection*, which consists in selecting the subset of more relevant features to the task at hand [54].

Layer	Forward Pass	Backward Pass
Input I	$s_j = \sum_{i \in I} w_{ij} x_i$ $y_j = a(s_j)$	
Hidden H1	$s_k = \sum_{j \in H1} w_{jk} y_j$ $y_k = a(s_k)$	$\frac{\partial E}{\partial y_j} = \sum_{k \in H2} w_{jk} \frac{\partial E}{\partial a_k}$ $\frac{\partial E}{\partial a_j} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial a_j}$
Hidden H2	$s_l = \sum_{k \in H2} w_{kl} y_k$ $y_l = a(s_l)$	$\frac{\partial E}{\partial y_k} = \sum_{l \in O} w_{kl} \frac{\partial E}{\partial a_l}$ $\frac{\partial E}{\partial a_k} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial a_k}$
Output O		$\frac{\partial E}{\partial y_l} = y_l - t_l$ $\frac{\partial E}{\partial a_l} = \frac{\partial E}{\partial y_l} \frac{\partial y_l}{\partial a_l}$

Table 1.1: Backpropagation for a multilayer perceptron with two hidden layers. The cost function is equal to $\sqrt{y_l - t_l}^2$.

Dropout While inexpensive from the computational point of view, *dropout* allows for regularization by randomly dropping (i.e. setting to zero) hidden units from the base network [55]. This strategy is comparable to bagging, which trains and evaluates several models on each test sample. However, the main difference between the two is that models are independent in bagging, whereas dropout comprises parameter sharing.

In the case of dropout, we perform the arithmetic mean over the sub-models, each identified by a mask vector $\mathbf{m}$ defining a probability distribution $p(y \mid \mathbf{x}, \mathbf{m})$ as in (1.33), where $p(\mathbf{m})$ is the probability distribution to sample $\mathbf{m}$ during training.

$$\sum_{\mathbf{m}} p(\mathbf{m}) p(y \mid \mathbf{x}, \mathbf{m}) \tag{1.33}$$

Instead of the arithmetic mean, we can use the geometric mean to find an appropriate model approximation with only one forward propagation, as shown in [21].
Besides computational efficiency, we can apply dropout to models that can be trained with SGD and use distributed representations. Furthermore, we force hidden units to perform well regardless of the interaction with other units and the context in which we utilize them. This principle is again biologically inspired: there are genes able to switch between different species while maintaining their usual features [56].
Fig. 1.21 provides an example of dropout.

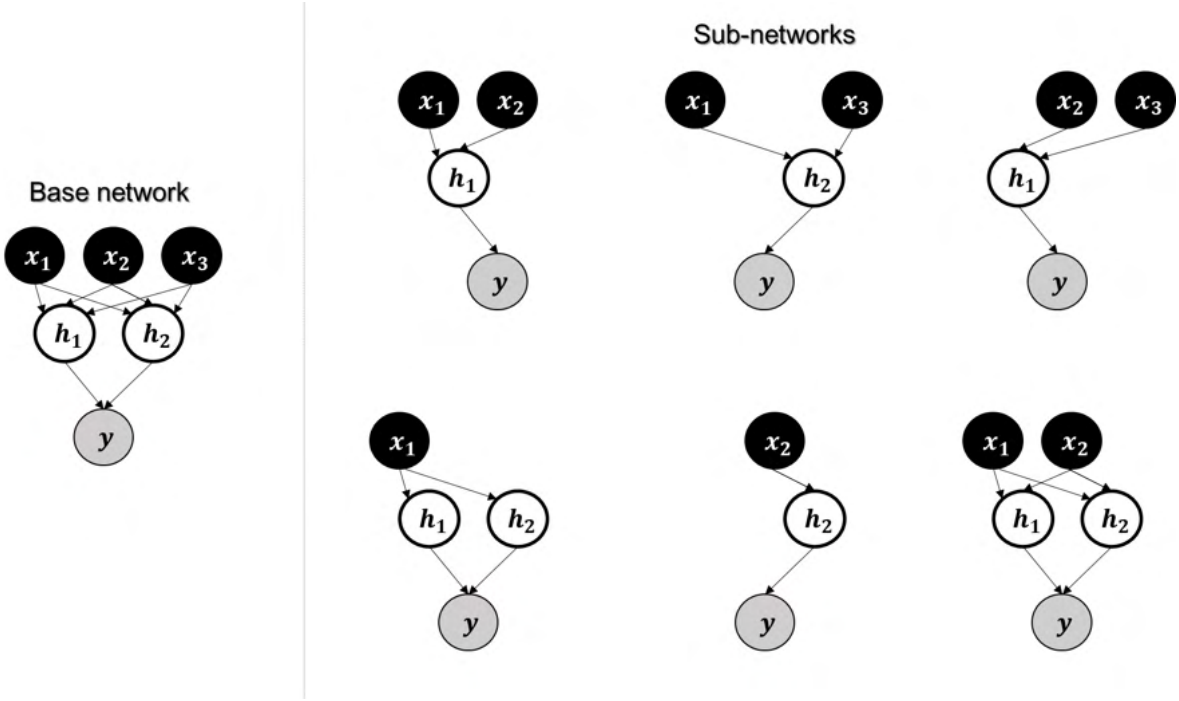


Figure 1.21: Example of dropout applied on a simple neural network with two hidden units. We show a few combinations of unit dropping by omitting units and their connections. h: hidden unit; x: input unit; y: output unit

Dataset Augmentation We can improve generalization by training the model with more data. However, the amount of data we have is usually limited and highly dependent on the task to solve. An alternative to deal with this issue is to create *realistic* synthetic data. That is simpler for tasks like classification. For instance, if we consider object recognition, we can reproduce many variation factors from the images with a realistic outcome [21]. Even just translating a few pixels has shown performance improvement, regardless of the translation invariance characterizing CNNs.
Nevertheless, one must be careful about the operations that may change the nature of the

augmented class (e.g. "6" transformed to "9" by rotation). Data augmentation must collide with the characteristics of each label to avoid introducing biases instead of ameliorating performances.

Adversarial Training Neural networks have amazed the world with their near-human or even above-human performances in different tasks, such as image recognition and playing Go [33, 57]. Hence, the curiosity to discover whether they used similar criteria as humans would to achieve these performances. To this end, *adversarial training* forces the network to recognize ad-hoc perturbed examples from the training set. These adversarial examples can fool the network into changing its prediction, albeit the human eye cannot perceive the modifications made [58]. Apart from the applications in fields such as cyber-security, we can use adversarial training as a form of regularization to make training more robust.

Transfer Learning Another possibility is *transfer learning*, exploiting the knowledge learned in one context to improve generalization in a different setting. For instance, we can consider that the first task for the network is to identify cats and dogs. We can assume that if enough data are available, the learned representations can serve to recognize also pandas and bears. That relies on the fact that low-level notions (e.g. edges, changes in lighting) are common to visual categories [21]. One must be careful, however, that the transferred domain is compatible with the target domain and adapt the choice for the transferred layers consequently.
Transfer learning has been applied successfully to medical images to address the lack of data [59, 60].

1.2.4.1.4 Parameter Initialization

Parameter initialization is crucial for a neural network to work well. Indeed, the generalization ability and optimization algorithm can be affected by the scale of the initial distribution [21]. It has revealed cumbersome to devise initialization strategies that are beneficial for generalization and optimization, and they usually favor one over the other. Recently developed initialization strategies are not so complex since neural network optimization is still not fully understood.

One necessary requisite is that initialization must allow for *breaking symmetry* [21]. For instance, in the case of hidden units with the same activation function and connected to the same inputs, these units must present different initialization so that each computes a different function, supported by random parameter initialization. Furthermore, a compromise should be found between too large or too small initial parameters, as the former can lead to exploding values, whereas the latter can cause an excessive shrink in the range of activations.

So far, we have referred to weight initialization, but we must also initialize biases. A common strategy is setting them to zero, compatible with most initialization techniques.

Glorot initialization is among the most used initialization strategies [61]. It initializes all layers to keep the same activation variance as well as the same gradient variance. We define it mathematically in (1.34), where $U(-\frac{1}{n_{outputs}+m_{inputs}}, \frac{1}{n_{outputs}+m_{inputs}})$ is the uniform distribution with n the number of units and W the weights.

$$W \approx U(-\frac{1}{n_{outputs} + m_{inputs}}, \frac{1}{n_{outputs} + m_{inputs}}) \tag{1.34}$$

1.2.4.1.5 Optimization Algorithms

At this point, one question comes to mind: how does an artificial neural network learn? Learning in ANNs is usually carried out by optimization, i.e. minimizing or maximizing some function $g(x)$, called the *objective function* or *criterion* [21].
Considering minimization, which is usually the preferred optimization strategy, $g(x)$ is called the *cost function*, *error function* or *loss function*. Hence, learning is an iterative process with the goal of minimizing the errors, and adjusting the weights in order to ameliorate performances.
Designing a neural network also includes choosing a cost function. Since deep networks usually define a distribution $p(\mathbf{y} \mid \mathbf{x}; \theta)$, we may use the maximum likelihood principle. It estimates a parameter maximizing the joint probability of the training data $\mathbf{x}$ as a function of the model parameters θ [21], $\mathbf{y}$ being the output.
We can consider a cost function given by the cross-entropy between training data and model's predictions $\mathbf{y}$, with the *entropy* expressing a measure of uncertainty with respect to a given distribution. Moreover, we can add a regularization term to cost functions in MLPs.
In the case of binary classification, we can define the binary cross-entropy $C_p(q)$ for two distributions p and q, as in (1.35).

$$C_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i)) \tag{1.35}$$

Due to nonlinearity, cost functions can become non-convex. That leads to small values of the cost function rather than an actual minimum. Moreover, unlike convex optimization, non-convex functions do not provide any guarantee that they will reach convergence.
Fig. 1.22 represents the cycle of learning applied to an artificial neuron. This process involves millions, if not billions, of parameters to be optimized, although a small amount compared to the numerous synapses populating the human brain [20].

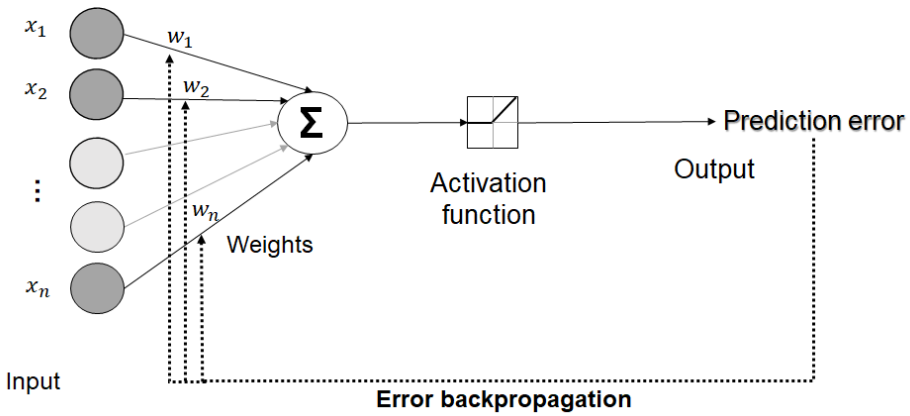


Figure 1.22: The learning cycle of an artificial neuron. After processing the input x and computing the weighted sum of the weights w, the result goes through an activation function producing an output. As the difference between the true and predicted label, the prediction error is injected back into the network to modify the weights and improve performance

One key difference between pure optimization and learning in deep networks is that the latter aims at improving some performance measurement (e.g. accuracy) rather than just minimizing the cost function for the sake of it, as in pure optimization [21].
In machine learning, we can reduce the expected generalization error (also called *risk*, corresponding to the error on unseen data) by minimizing the expected loss on the training set. That is known as *empirical risk minimization*. Due to many loss functions being nondifferentiable, we use a *surrogate loss function* that is easier to optimize. To reduce overfitting, techniques such as *early stopping* can stop learning based on a convergence criterion.
We can employ different optimization algorithms depending on the number of samples used for the gradient update [21]:

- *Batch.* Also known as *deterministic*, this approach implies using the entire training set.
- *Stochastic.* This optimization algorithm processes only one sample at a time. The *online* variant is specific for when data are created simultaneously from a stream.
- *Minibatch.* Widely employed in deep learning, it considers a subset of examples randomly selected from the training set. Additionally, we need to shuffle data before random selection to avoid bias due to their ordering.

Gradient Descent One way to minimize a function $y = g(x)$ is to compute its *derivative* denoted as $g'(x)$ or $\frac{dg}{dx}$. We can interpret the latter as the slope of $g(x)$ at the point x.
We can use this derivative to establish how to change the input in order to improve the output. In this regard, the *gradient descent* technique postulates to decrease $g(x)$ according to small steps of x following the opposite sign of the derivative (see Fig. 1.24) [62].

There can be points in which the derivative equals zero, knows as *critical points* or *stationary points*, giving no information about which direction to take. Moreover, we can define two types of *saddle points* [21]:

- *Local minimum*, as a point in which $g(x)$ is lower than at all neighboring points and $g(x)$ cannot decrease by taking infinitesimal steps;
- *Local maximum*, as a point where $g(x)$ is higher than at all neighboring points and $g(x)$ cannot be increased by taking infinitesimal steps.

Fig. 1.23 shows the different types of critical points.

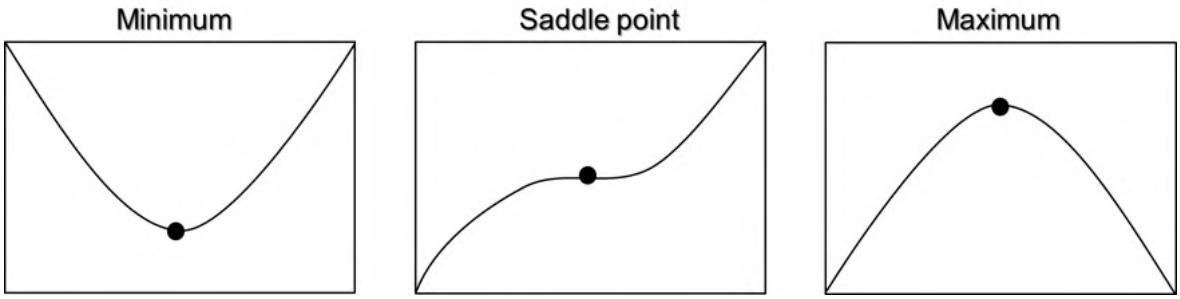


Figure 1.23: Types of critical points

Instead, a *global minimum* is a point for which $g(x)$ assumes the absolute lowest value. There can be multiple global or local minima, making optimization more difficult especially for multidimensional functions (e.g. in deep learning). For this reason, we generally compromise to accept low function values, which do not necessarily qualify as minimal [21].

For functions with multiple inputs, for instance $g : \mathbb{R}^n \to \mathbb{R}$, we must consider *partial derivatives*, indicated as $\frac{\partial}{\partial x_j} g(x)$. We can then compute the derivative with respect to each variable x_j, obtaining the gradient which is the vector of all partial derivatives $\nabla_{\mathbf{x}} g(\mathbf{x})$. According to this formulation, we evaluate how small changes in each variable influence the output.
An example is the *gradient descent* or *steepest descent*, which decreases the function following the direction of the negative gradient. The notion of *directional derivative* in direction $\mathbf{u}$ (a unit vector) can be used to find the direction in which g is reduced the fastest, as in (1.36).

$$\min_{\mathbf{u}, \mathbf{u}^\top \mathbf{u}=1} \mathbf{u}^\top \nabla_{\mathbf{x}} g(\mathbf{x}) \tag{1.36}$$

We define the point resulting from this operation in 1.37, where ϵ is the *learning rate*. The latter is a positive number (usually small) establishing the step size.

$$\mathbf{x}' = \mathbf{x} - \epsilon \nabla_{\mathbf{x}} g(\mathbf{x}) \tag{1.37}$$

Fig. 1.24 provides an example of gradient descent. Gradient descent reaches convergence when each gradient entry equals zero or practically is very close to zero.

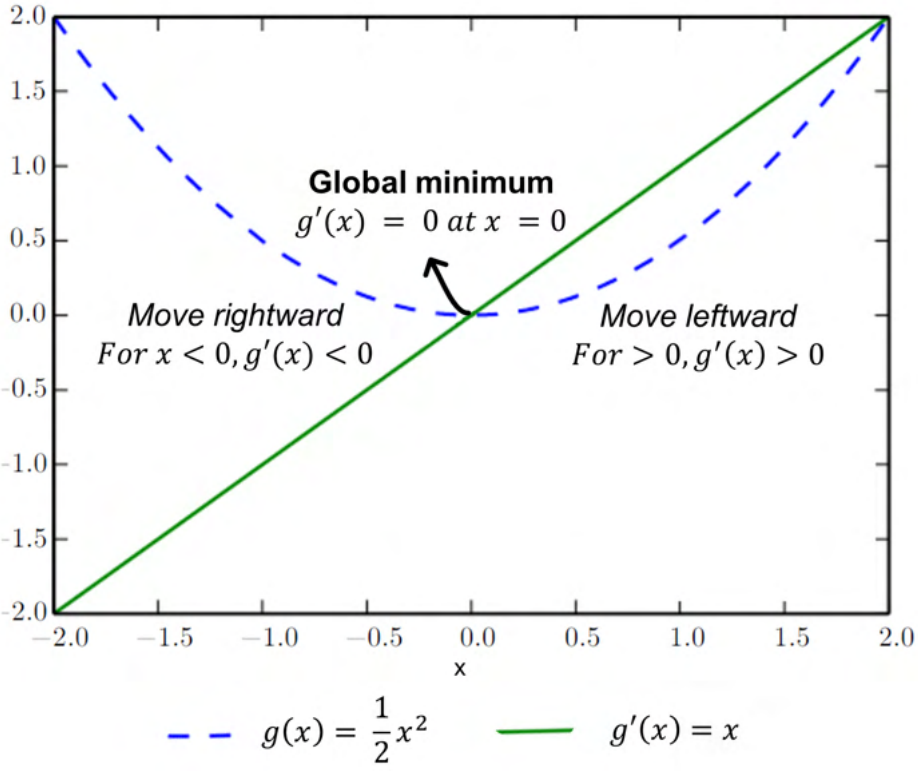


Figure 1.24: Representation of gradient descent, showing how the function derivative can guide to reach a minimum. Adapted from [21]

When the function output is a vector, we can define the *Jacobian matrix*, including all the partial derivatives as in (1.38), with $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ and $\mathbf{J} \in \mathbb{R}^{m \times n}$.

$$J_{k,j} = \frac{\partial}{\partial x_j} g(\mathbf{x})_k \tag{1.38}$$

Moreover, we can compute the derivative of a derivative known as the *second derivative*. The latter can inform about the *curvature* of a function (see Fig. 1.25). For instance, considering a quadratic function, we have the following cases:

- No curvature (i.e. flat line), characterized by a second derivative equal to zero;

- Negative curvature, the function goes downward;
- Positive curvature, the function goes upward.

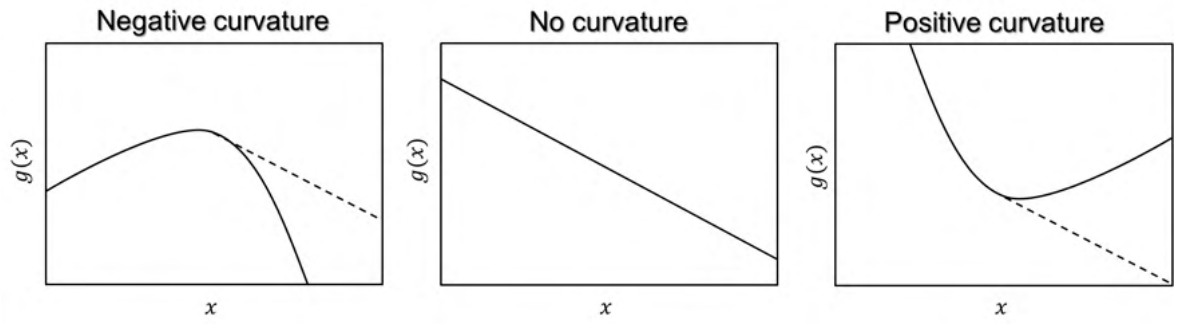


Figure 1.25: Example of curvatures

In the case of many second derivatives due to multiple inputs, we can define the *Hessian matrix* in (1.39). The Hessian matrix is symmetric owing to the commutative property in the points where the second partial derivatives are continuous: $H_{k,j} = H_{j,k}$. The directional second derivative can give us an idea about the performance of each gradient descent step.

$$\mathbf{H}(g)(\mathbf{x})_{k,j} = \frac{\partial^2}{\partial x_k \partial x_j} g(\mathbf{x}) \tag{1.39}$$

However, since each direction has a second derivative, one point can have many second derivatives. This complicates the procedure to find the optimum value due to the different speeds at which derivatives can decrease or increase.
One solution is to exploit the information provided by the Hessian matrix by applying *Newton's method*, which uses a second-order Taylor series expansion to approximate the function closely to some point. Newton's method is an example of *second-order optimization*, in contrast to gradient descent which falls into the category of *first-order optimization algorithms*.

Given the complexity of deep learning functions, we may apply some restrictions to these functions, like the requirement to have a Lipschitz continuous derivative.
If other restrictions are required, we can refer to the field of *convex optimization*, which applies to convex functions only [63]. The latter is less employed in deep learning but can be used to prove the convergence of some DL algorithms.

Stochastic Gradient Descent SGD is a variant of gradient descent widely used in deep learning [21]. It is common practice to calculate the cost function as a sum of this function

evaluated for each example. We can thus expect the computational cost needed to make the gradient descend to increase rapidly with training set size.
SGD is based on the gradient as an expectation, so we can compute it on a small set of samples, called *minibatch*. That enables fitting large training sets with gradient updates considering a few examples.
SGD estimate denoted as $\mathbf{d}$ on s examples belonging to the minibatch is described in (1.40), where C is the cost function, $\mathbf{x}$ is the input vector, y is the output, and $\boldsymbol{\theta}$ represents the parameters to be updated.

$$\mathbf{d} = \frac{1}{s}\nabla_\theta \sum_{j=1}^{s} C(\mathbf{x}^{(j)}, \mathbf{y}^{(j)}, \boldsymbol{\theta}) \tag{1.40}$$

The gradient update is closely related to the learning rate, which in practice reduces gradually according to some condition (e.g. no improvement in the loss function after a certain number of epochs).

Momentum Although efficient, SGD can become very slow. The *Momentum* method makes learning faster by exploiting an exponential decay of the moving average from past gradients, defined with the variable $\boldsymbol{\nu}$ [64]. The latter represents a sort of velocity, determining the direction and speed to move parameters within the parameter space. The exponential decay depends on a hyperparameter α in the range $[0, 1)$. Equations (1.41) and (1.42) define parameters update using momentum, in which ϵ is the learning rate and $\nabla_\theta \frac{1}{s} \sum_{j=1}^{s} C(g(\mathbf{x}^{(j)}; \boldsymbol{\theta}), \mathbf{y}^{(j)})$ are the gradient elements.

$$\boldsymbol{\nu} \leftarrow \alpha\boldsymbol{\nu} - \epsilon\nabla_\theta \left(\frac{1}{s} \sum_{j=1}^{s} C(g(\mathbf{x}^{(j)}; \boldsymbol{\theta}), \mathbf{y}^{(j)}) \right) \tag{1.41}$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{\nu} \tag{1.42}$$

Adaptive Learning Rate The learning rate is one of the most tricky hyperparameters to set. As shown in Fig. 1.26, it can considerably influence network performance by determining the pace at which we move along the cost function to reach a minimum. Rather than choosing a fixed value, different optimization algorithm proposes an adaptive learning rate, such as AdaGrad [65], RMSProp [66] and Adam [67].

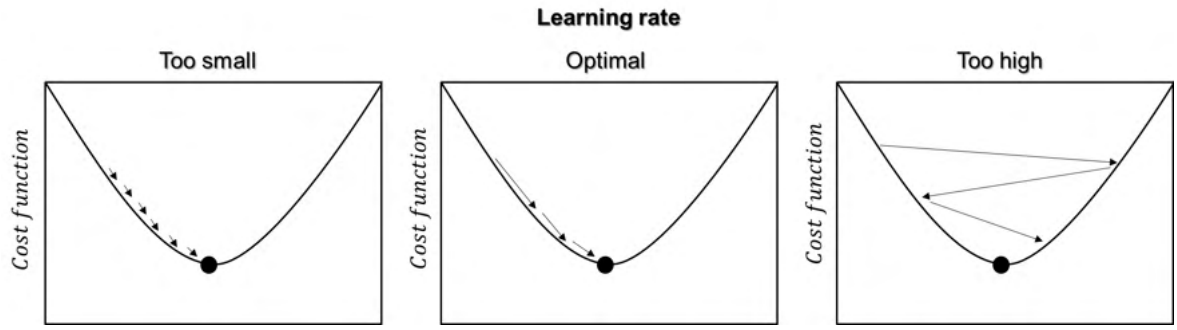


Figure 1.26: Representation of the impact of learning rate on performances. When it is too small, this slows performances. Instead, if it is too high, it can miss a valuable optimum point. The best approach is to have an adaptive learning rate, gradually decreasing according to performance improvement

Adam stays for adaptive moments and can be interpreted as a combination of momentum and RMSprop but with some differences [21]:

- Momentum is included as an estimate of the first-order moment of the gradient;
- Estimates of first- and second-order moments undergo bias corrections to consider their initialization at the origin.

Adam is particularly suited for tasks involving multidimensional arrays and seems robust regarding hyperparameters.

1.2.4.1.6 Batch Normalization

Covariate shift is one of the most feared enemies when training deep networks. It is due to the changing distribution of layers during the training phase, thus forcing to reduce the learning rate and carefully initialize parameters [68]. Batch Normalization (BN) tries to contrast this issue by normalizing groups of examples rather than a single group [69].
We can train neural networks efficiently with stochastic gradient descent, which minimizes the parameters θ to minimize the loss as in (1.43), where $t_{1...N}$ represents the training set.

$$P = arg \min_{P} \frac{1}{N} \sum_{j=1}^{N} C(t_j, P) \tag{1.43}$$

Training gets divided into steps, each characterized by a *mini-batch* of size s. Using the minibatch, we can approximate the gradient of the loss function C with respect to the parameters as in (1.44).

$$\frac{1}{s} \frac{\partial C(t_j, \theta)}{\partial \theta} \tag{1.44}$$

Two main simplifications have been advanced to perform batch normalization [69]:

- Independently normalizing scalar features to have zero mean and unit variance [70];
- Each minibatch providing statistics such as mean and variance for every activation.

Experiments performed on bench-mark CNNs, like for ImageNet [71] and MNIST [72], confirmed the efficacy of batch normalization. Although these networks were slightly modified, for example, using higher learning rates and removing dropout (see Section 1.2.4.1.3), they reached a lower top-5 error than the one obtained with the same CNNs without BN implementation.

1.2.4.2 Convolutional Neural Networks

Convolutional neural networks have advanced image recognition thanks to automatic feature extraction optimized for the task [19]. The strength of such an approach relies upon a data-driven character coupled with the powerful discrimination abilities of artificial neural networks. Before delving into their description, let us briefly explain their origin.

1.2.4.2.1 Neuroscientific Foundations

One fundamental inspiration for CNNs came from the study on the cat's visual cortex by Hubel and Wiesel, dating back to 1962 [28]. Their work showed that object recognition is performed in stages from the retina to the inferotemporal cortex, identifying the so-called *ventral stream*.
For instance, when we look at an object, the signal goes from the primary visual cortex V1 to visual areas V2 and V4 through a series of filters. Here in the inferotemporal cortex, the neurons related to the concept associated with the considered object activate and make us recognize the object. In V1, millions of bundles of pyramidal neurons (from 50 to 100) connect to small areas of the visual field, called *receptive fields*, which react to simple traits, such as lines with various orientations. These are known as *simple cells* and can detect the same pattern at different locations. Then, there are *complex cells*, neurons able to aggregate the information carried out by simple cells with a certain tolerance to position, distortions, or shifts (see Fig. 1.27).

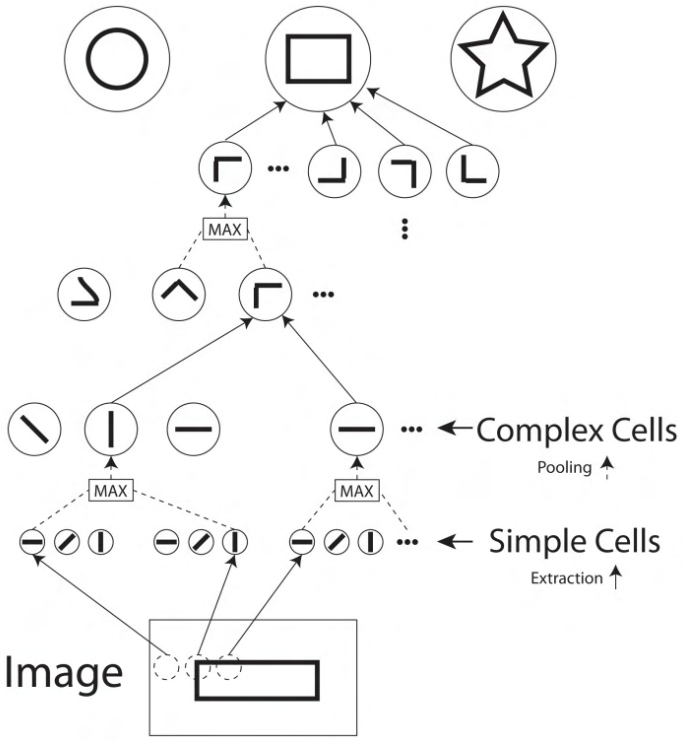


Figure 1.27: Example of shape recognition performed by simple and complex cells. Note how complex cells integrate the information retrieved by simple cells by performing basic operations (e.g. finding the maximum value) to obtain the final shape. Reproduced from [73], (Vincent de Ladurantaye, Jean Rouat and Jacques Vanden-Abeele, 2019). CC BY-SA 3.0

According to the two-streams hypothesis [74], the ventral (or "what") pathway leading to the temporal lobe involves object recognition. The dorsal (or "where") pathway instead relates to spatial localization and arrives at the parietal lobe. Fig. 1.28 depicts these two pathways.

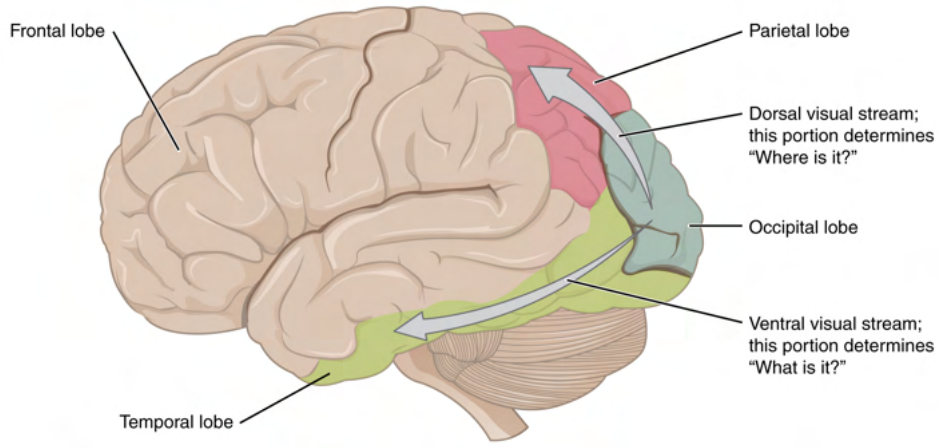


Figure 1.28: Representation of the ventral (or "what") and dorsal (or "where") streams for visual processing in humans. Reproduced from [75], OpenStax College, 2013. CC BY 3.0

In light of the visual processing mechanism, the following concepts constitute the basis for CNN functioning:

- Local receptive field, as neurons in V1 are connected to small parts of the image;
- Repetition of the same operation on the visual field, as different neurons can detect the same feature in different areas of the image.

In analogy with simple cells, we find the *convolution* operation in CNNs (hence their name) for detecting simple patterns. Similarly to complex cells, the *pooling* operation aggregates the information retrieved by convolutional layers to reduce input dimension and sensibility to distortions and shifts.
In a typical CNN architecture, convolution and pooling layers alternate to perform automatic feature extraction. Given that we optimize parameters during training, we obtain feature extractors optimized for the task. That finally frees from manually devising and computing features by directly exploiting the optimization procedure. An MLP can process the outcome as a feature vector for each input to perform some task.

1.2.4.2.2 Convolution and Pooling

We can characterize a convolutional layer by a *filter* or *kernel*, a matrix of numbers learned during training. Each filter passes on the image producing a *feature map*, showing the parts of the image which activated it.
The convolution operation is typically denoted by an asterisk, as denoted in (1.45), where f is a real-valued function and w a weighting function [21].

$$c(t) = (f * w)(t) \tag{1.45}$$

In the case of a three-dimensional image, we can compute convolution as in (1.46), with X, the image, and K, the three-dimensional kernel. Multidimensional arrays, such as images, are also known as *tensors*. We can interpret discrete convolution as matrix multiplication.

$$C(h, i, j) = (X * K)(h, i, j) = \sum_{l} \sum_{m} \sum_{n} X(l, m, n) K(h - l, i - m, j - n) \tag{1.46}$$

Fig. 1.29 provides a numeric example of the convolution operation. We can perform convolution with a specific *stride*, i.e. the number of pixels to consider when passing over the image. If the stride is equal to one for all directions, we slide the kernel of one pixel in each direction to compute the output.

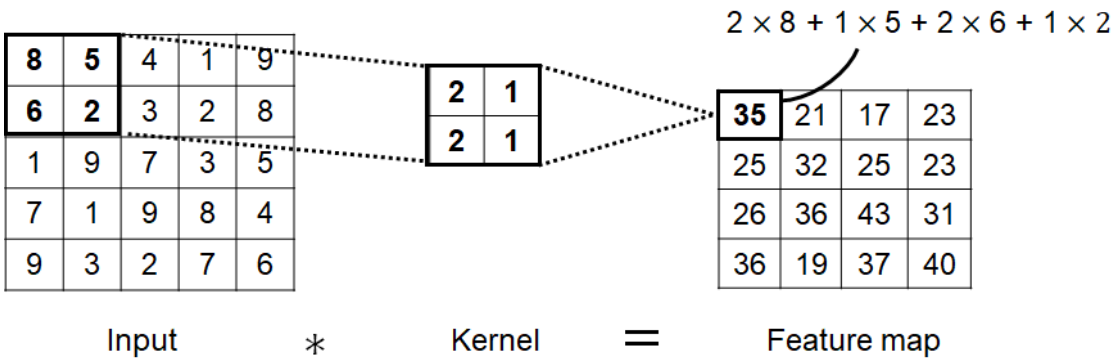


Figure 1.29: Example of the convolution operation computed on a 2D image with stride equal to 1. The *valid* method is applied as the kernel lies entirely in the image

Convolution presents at least three advantages [21]:

- *Sparse interactions.* Convolutional kernels are usually much smaller than the input. That leads to a smaller number of parameters, meaning less memory and fewer computations. More interestingly, units in deep layers may cover a nonnegligible part of the input.

- *Parameter sharing.* In the case of an image, instead of having each pixel connected to a single weight (as it would be in the fully connected scheme of an ANN), kernel weights in CNNs can learn the same pattern found at different locations in the image.

- *Equivariant representations.* A function $h(x)$ is said to be equivariant if changes in the input correspond to equivalent changes in the output. This property comes in handy when a function activates in several input locations (e.g. for edge detection).

Convolution leads to the computation of linear activations, typically followed by a nonlinear activation function (examples are available in Fig. 1.19). We then perform the *pooling* operation to summarize the convolutional output of its neighborhood by computing the maximum or mean value. Fig. 1.30 shows the two most used types of pooling.

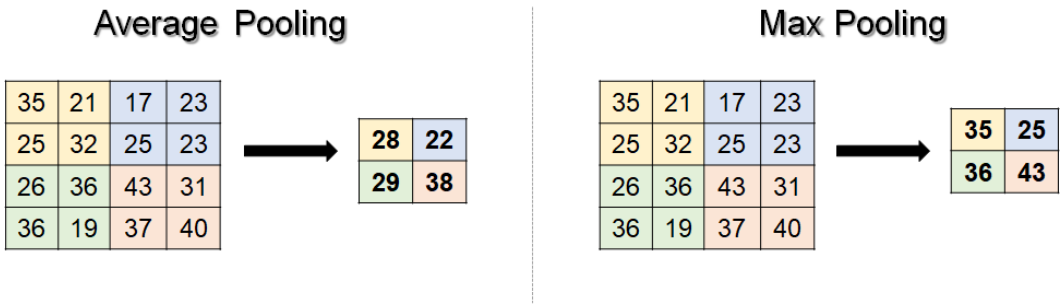


Figure 1.30: Examples of average and max pooling computed on a 2D image

The alternation of convolution and pooling layers produces a set of features, organized hierarchically, going from simple to very abstract levels [19]. Fig. 1.31 illustrates an example of feature extraction obtained with a CNN.

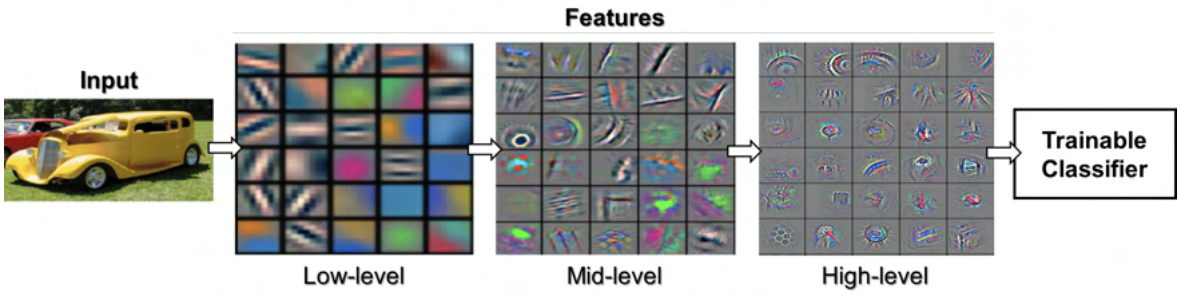


Figure 1.31: Example of hierarchical feature extraction obtained using a CNN. Adapted from [76]

Over the years, CNN architectures have evolved to tackle different upcoming issues (e.g. the difficulty of optimizing deep networks and overfitting). In the following, we describe the most famous CNNs that continue to inspire new variants.

1.2.4.2.3 Main Architectures

LeNet-5 Dating back to 1998, *LeNet-5* was the first CNN architecture, named after its creator Yann Le Cun and devised for digit recognition [72]. Although constituted by only two convolutional layers due to the computational limitations of the time, it obtained an acceptable error rate.

AlexNet In 2012, *AlexNet* won the ImageNet competition (ImageNet Large Scale Visual Recognition Challenge (ILSVRC)), an object recognition task consisting in predicting for each image 5 out of 1000 possible categories [33]. If the correct answer fell among the five proposed ones, then the output was correct.
AlexNet totaled only 16% for the error rate compared to 25% the year before. This result represented the breakthrough for CNNs to gain their place as powerful image recognition methods. Compared to LeNet-5, AlexNet has more filters per layer and different convolution sizes. The network was trained simultaneously on two GPUs for six days.

ZFNet One year later, *ZFNet* triumphed as the winner of the ILSVRC by reaching 14.8% as the top-5 error rate [76]. This architecture resembles AlexNet but with some hyperparameter tweaking.

VGGNet In 2014, the Visual Geometry Group (VGG) devised a deeper network named *VGGNet*, characterized by 16 convolutional layers and 3×3 convolutions [77]. VGGNet trained for 2-3 weeks on 4 GPUs.

Fig. 1.32 illustrates these famous architectures.

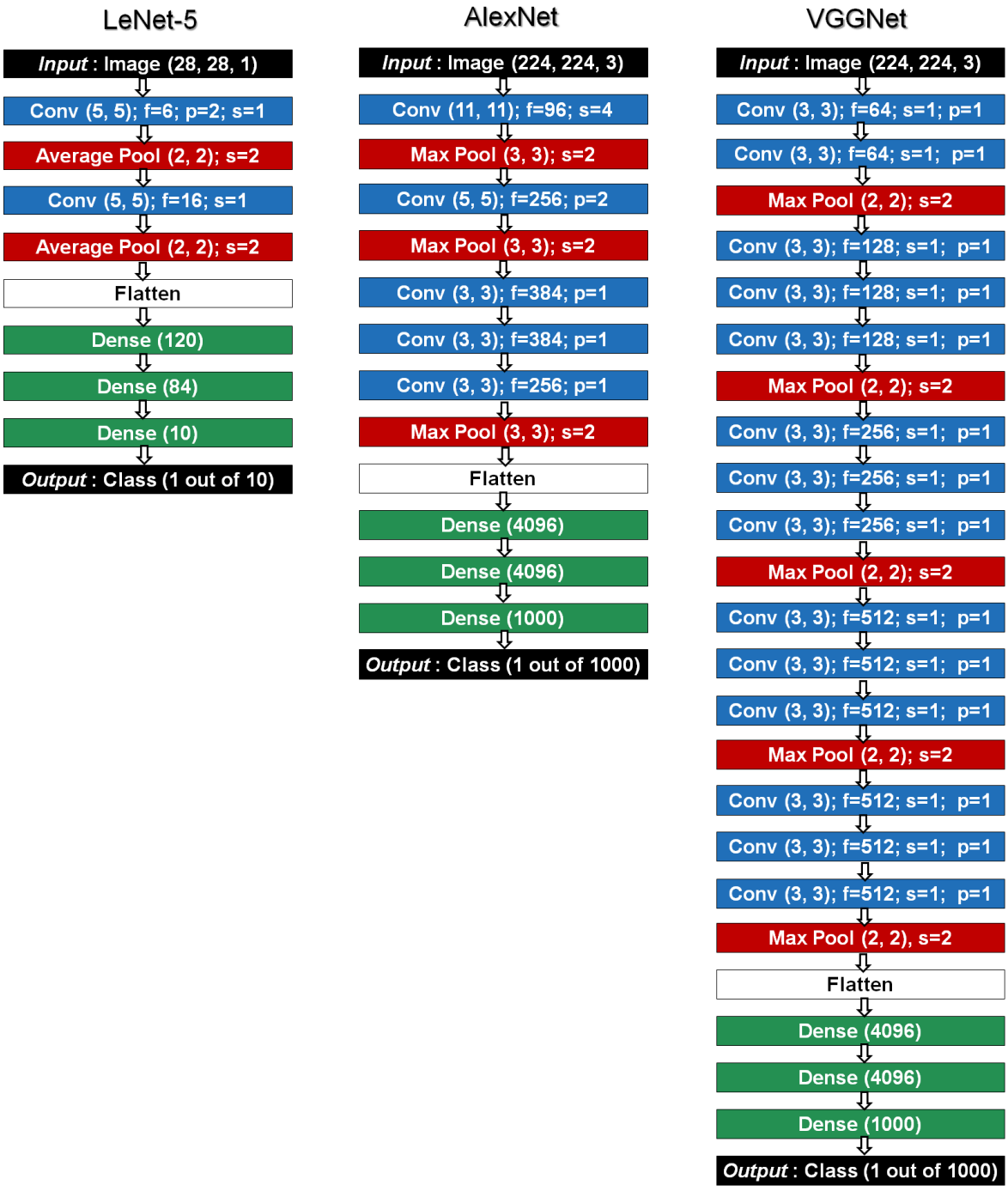
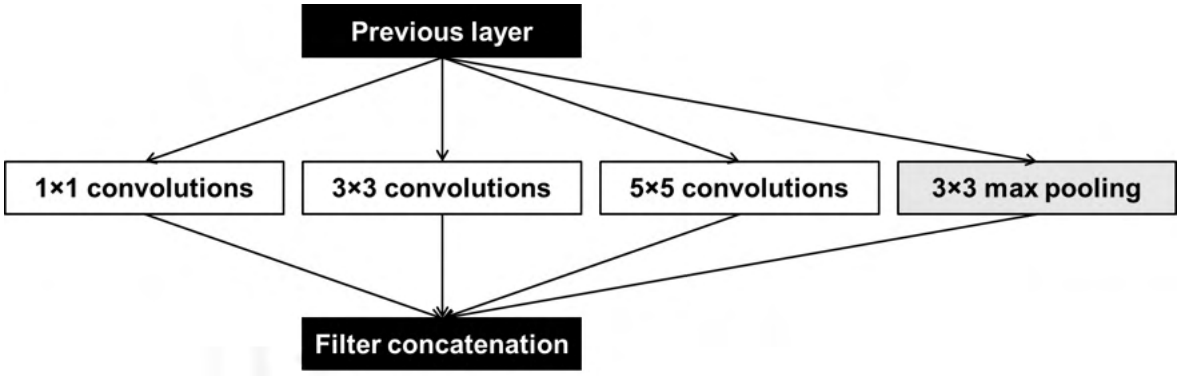
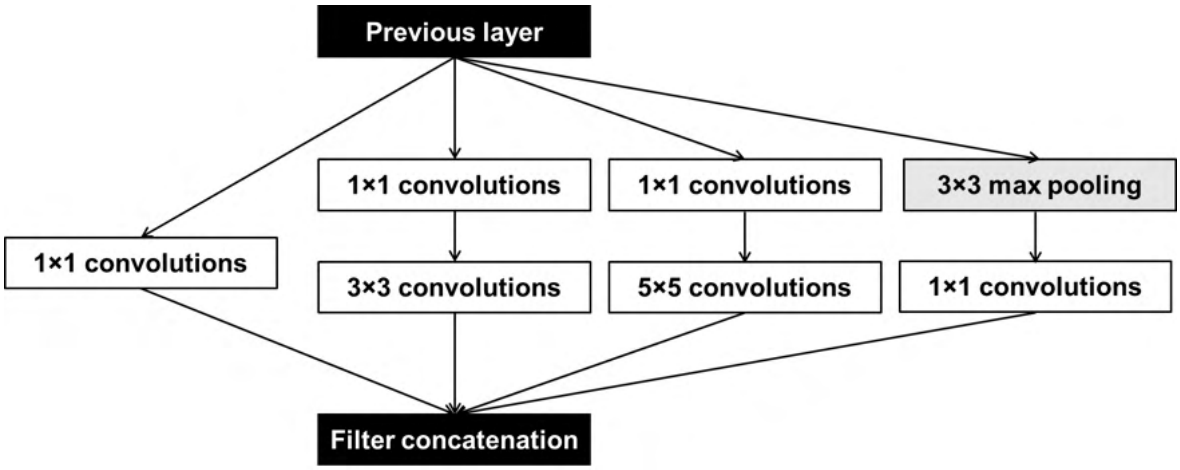


Figure 1.32: Most famous CNN architectures: LeNet-5 [72], AlexNet [33], and VGGNet [77]. Image dimensions is reported as (height, width, channels). Filter size is specified for Conv and Pool layers. The number of units is indicated for Dense layers. Before inputting to the Dense layers, features are reshaped in a 1D vector (Flatten). Conv: convolutional layer; Pool: pooling layer; f: number of filters; p: padding; s: stride

GoogLeNet Another architecture worth mentioning is *GoogLeNet* (also called *Inception V1*), developed by Google, which achieved a 6.67% top-5 error rate, winning the ILSVRC 2014 [71]. Inspired by LeNet-5, it presents 22 layers (excluding pooling). The novelty introduced by this architecture was the *inception* module, consisting of a variable number of convolutional filters with different sizes and additional pooling layers, whose results are concatenated and fed to the next layer. That led to significantly ameliorated performances while keeping the computational burden limited.
Fig. 1.33 provides two variants of the inception module.



(a) Plain version



(b) Version including dimensionality reduction via 1×1 convolutions

Figure 1.33: Variants of the inception module. Adapted from [71]

ResNet As the winner of the ILSVRC 2015, ResNet achieved a 3.57% top-5 error on the test set [51]. This architecture represents an efficient alternative to training deep networks (up to 152 layers, eight times deeper than VGGNet) to solve the problem of vanishing/exploding gradients [61, 78]. To this end, the network presents *residual* blocks, as schematized in Fig. 1.34. The underlying assumption is that the model tries to optimize a function closer to

an identity mapping than a zero mapping, hence the strategy of fitting a residual mapping. Residual blocks are called *identity blocks* when the input and output dimensions are the same. ResNet addresses the *degradation* problem (accuracy reaches a plateau and then decreases fast) by increasing the network's depth. The cause of such an issue is not attributable to overfitting but rather highlights that optimization can differ according to the network.

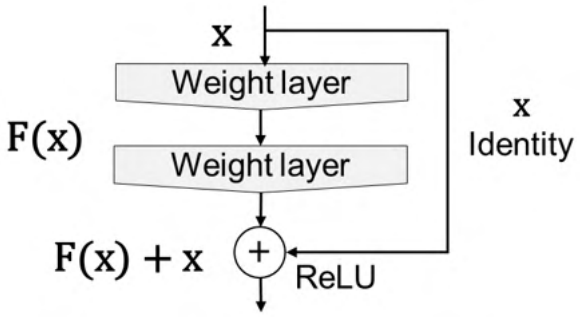


Figure 1.34: Residual block used in ResNet to allow for residual learning. Adapted from [51]

All-Convolutional Network This architecture, developed in 2015, has only convolutional layers for the feature extraction part [79]. Pooling layers are replaced by convolutional layers with stride equal to 2, thus making the network learn the pooling operation. Moreover, convolutional filters are kept small (size<5), thus diminishing the number of parameters and even introducing a form of regularization.

U-Net Specially designed for biomedical image segmentation, U-Net represents an upgrade of a previous fully convolutional network [80]. It has won several awards, including the Cell Tracking Challenge at International Symposium on Biomedical Imaging (ISBI) 2015. U-Net is composed of a [81]:

- *Contracting path*, following the structure of a traditional CNN;
- *Expansive path*, performing an upsampling of the feature maps computed in the contracting path.

This architecture was designed to work with few training samples, as is often the case in the medical domain. Data augmentation falls indeed in the pipeline, with techniques adapted to the type of data, to improve the network's learning ability.
Fig. 1.35 illustrates the original U-Net architecture scheme.

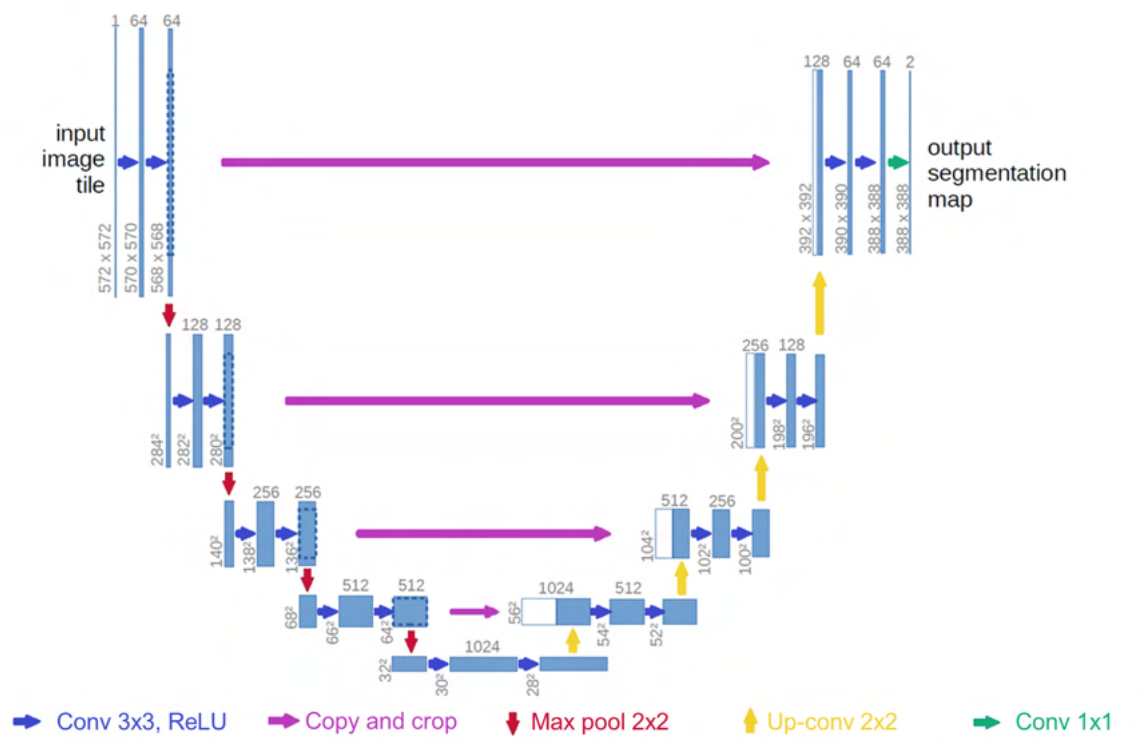


Figure 1.35: U-Net architecture. Conv: convolutional layer; Pool: pooling layer; ReLU: Rectified Linear Unit. Adapted from [81]

1.2.5 Explainable AI

One of the foremost concerns of AI is creating tools comprehensible to human beings to favor their acceptance and use. That is a concrete issue in the medical domain, in which integrating different and non-homogeneous, often missing, high-dimensional data is part of the everyday clinical routine [82].
When dealing with the so-called *usable intelligence*, it becomes imperative to [83]:

- Learn from the available set of data;
- Gain insights from these data;
- Develop the ability to generalize;
- Contrast the *curse of dimensionality*, denomination given by Richard E. Bellman in the context of dynamic programming in the case of high-dimensional spaces [84, 85]. The latter can cause data to become very sparse fast, thus requiring more data whose amount increases exponentially with the dimensionality.

Transparency from the predictors' decision-making process must be assured so that professionals can understand their underlying mechanisms. That is why *explainable AI* comes into play. The struggle of explainability has always accompanied AI due to the failures in elucidating algorithms' decisions in light of their undeniable achievements [86].

Deep learning algorithms are often defined as *black boxes* because human operators still cannot fully understand some aspects of their decision-making process [87,88]. Two notions have turned out to be valid for neural networks [89]:

- *Compactness.* We can write learning rules in a few pages of high-level code;
- *Compressibility.* It is higher for specialized systems such as DL algorithms. However, even simplified to plain models, these algorithms remain hardly understandable.

If we compare the brain to neural networks, it also appears like a black box from the outside as we do not know a person's thoughts, yet we still trust humans [89]. That is as fascinating as it can be frightening.

In this context, *understanding* generally means elucidating the black box without limiting the explanations to low-level algorithm descriptions [82]. Nonetheless, we can distinguish between two concepts [90]:

- *Interpretation*, i.e. representing an abstract concept using an area familiar to humans;
- *Explanation*, i.e. including an ensemble of features used for decision-making, which fall into the interpretable domain.

Moreover, we can establish the value of an explanation by considering *completeness*, which can accurately describe operations performed by the system [91].

1.2.5.1 Visualization Techniques

The explainability of deep networks can be described using three principal categories, following the taxonomy in [91]:

- *Processing*. It suggests using a *proxy model* akin to the original model but more straightforward to explain or the generation of *saliency maps* to reveal regions of the input relevant to the prediction [92]. One of the most representative examples of the linear proxy model approach is Local Interpretable Model-agnostic Explanations (LIME), able to locally approximate any black-box function adopting an interpretable model [93].

Decision trees are another type of proxy model, making the effort of decomposing neural networks. An example is DeepRED [94], which generates trees very similar to neural networks but expensive in terms of time and computational memory. Automatic-rule extraction methods are an alternative to the previous techniques [95]. Various techniques allow for retaining more information throughout the network e.g. CAM [96], Grad-CAM [97], SmoothGrad [98], LRP [99], Integrated Gradients [100].

- *Representations*. Exploiting the granular organization of deep networks, we can examine them at different levels:
  - *Units*, contributions of single neurons in a qualitative manner (e.g. visualizing their response) or quantitatively by solving a transfer learning problem;
  - *Vectors*, exploring directions by linear combinations of neurons;
  - *Layers*, probing whether they are capable of addressing problems in a domain other than what they were trained for (i.e. transfer learning).
- *Explanation-producing systems*. Grouped in:
  - *Attention networks*, including functions that networks learn to find out the influence of inputs and features, thus making the decision process more comprehensible;
  - *Disentangled representations*, characterized by single dimensions with insightful and uncorrelated variation factors;
  - *Generated explanations*, with explanations incorporated in the training process.

There exist three main types of saliency methods [101]:

- *Gradients.* Also known as *sensitivity* [92, 102], they show modifications of the output when the input slightly varies;
- *Signal methods*. They aim to detect input patterns provoking neuron activation in deeper layers;
- *Attribution methods*. They decompose the value at output neurons into contributions from each input dimension (e.g. Deep-Taylor Decomposition [103] and Integrated Gradients [100]). They insist on completeness, unlike gradients methods.

In the following, we shortly present some of the most employed visualization methods with applications in neuroimaging.

1.2.5.1.1 Gradients

Saliency maps For a given class, saliency maps can be obtained as follows [92]:

1. Computation of the derivative, arranged as a vector of the input image using backpropagation;
2. Rearrangement of each entry to a specific pixel and computation of its absolute value. In the case of images with more than a single channel, we consider the channel presenting the maximum value.

These maps can inform how the output responds when moving in a specific direction over the input [92, 102]. This technique is presented as a generalization of Deconvnet (see Section 1.2.5.1.2). The principal drawback of saliency maps is their reliance on fully connected layers.

1.2.5.1.2 Signal Methods

Deconvnet The idea behind *Deconvnet* is to create a network able to reconstruct the activity of each layer, leading to pixel mapping of the input space [76].
Representing the positions of local maxima belonging to a specific pooling region, the so-called *switches* connect the convolution network to Deconvnet. Such a method allows tracking the most frequently considered pixels and distinguishing some features thanks to the activation maps computed for each layer. Another contribution of Deconvnet is *occlusion sensitivity*, to find which parts of the input are relevant for classification (e.g. the desired object or the surroundings). Deconvnet performance may improve by using a loss function to detect multiple objects in the images [76].

Guided Backpropagation *Guided backpropagation* was introduced in 2015, along with a novel CNN architecture made exclusively of convolutional layers [79] (see Section 1.2.4.2.3). This method relies on the absence of pooling, which theoretically ensures independence from the input and allows for investigating intermediate layers.
Guidance is provided by higher layers. Inspired by deconvnet (see Section 1.2.5.1.2) and backpropagation, higher layers neglect all negative values in the former or the latter method rather than considering only one option. That preserves negative gradients belonging to neurons reducing the activation of deeper layers.

PatternNet PatternNet is a visualization method based on projecting the signal obtained at each layer back into the input space [104]. It corresponds to gradient calculation with *informative directions* replacing network weights.
To inspect deep models, PatterNet examines a simple network setup (composed of a linear model) using data generated by a linear model. In such a way, we can clearly define the following [105]:

- *Signal*, components of the input containing relevant information;
- *Distractor*, an ensemble of factors that complicate correct identification of the desired output.

In signal methods, neural networks retrieve the signal standing for the input parts causing activation. Instead, attribution methods give quantitative information about the contribution of signal dimension passing throughout the network. We can compute the attribution for a linear model by element-wise multiplication between the weights and the signal.

1.2.5.1.3 Attribution Methods

PatternAttribution This approach provides explanations of classifier decisions at the pixel level, producing heatmaps to compensate for non-linearity [99].
Two main differences separate sensitivity maps from the pixelwise decomposition approach:

- The function value at the prediction point x and its differential are not directly linked;
- Pixelwise decomposition wishes to understand classifier predictions in a given state by a set of roots from the prediction function.

One of the most delicate aspects of Deep Taylor Decomposition [103] is to find a root point, i.e. the point at which a differentiable function is zero. Instead, PatternAttribution focuses on pixelwise importance by exploiting the contribution of each neuron to the final classification [104].

Deep Taylor Decomposition Deep Taylor decomposition is a method aiming at explaining nonlinear classifier decisions by determining the contribution of each decision to the input [103]. The goal is to directly associate every pixel n of image $\mathbf{I}$ with a *relevance score* $R_n(\mathbf{I})$. The latter provides information about the contribution of each pixel in the explanation for the classifier prediction $f(\mathbf{I})$. The heatmap $R_n(\mathbf{I})$ must present the following properties:

1. *Conservative*, if the sum of pixel-wise relevance matches the final relevance found by the model;

2. *Positive*, if it is composed of values equal to or greater than zero;

3. *Consistent*, if the previous two properties are fulfilled. The heatmap must be empty with no label of interest in the image.

This method relies on the *divide-and-conquer* paradigm, based on the fact that deep neural network functions divide into less complex subfunctions such as neurons.
Deep Taylor decomposition comes with usability on trained models with diverse input and structures, as it does not need hyperparameter tuning, offering transparency for classifier decisions.

1.2.5.1.4 CAM & Grad-CAM

Class Activation Mapping In 2015, Class Activation Mapping (CAM) was proposed to discover relevant regions identified by CNNs for a specific label [96].
CAM consists in projecting the output layer weights into the convolutional feature maps. Given an image, the activation of a particular unit is represented at a specific spatial location in the last convolutional layer. Global average pooling is performed on the feature maps relative to that unit, and the softmax input is obtained by summing the activations multiplied by the unit weights.
The class activation map CAM_l for label l at grid position of coordinates x and y is presented in (1.47), where:

- p_j^l is the weight p considering unit j and label l;
- $a_j(x, y)$ is the activation of unit j at grid position (x, y).

$$CAM_l(x, y) = \sum_k p_j^l \, a_j(x, y) \tag{1.47}$$

The ultimate step comprises an upsampling operation to visualize CAM with a size matching the input image.
Regarding global average pooling, its use is encouraged compared to global max pooling. The former can recognize an entire object, whereas the latter focuses on the most defining parts [96].
One limitation of this approach is the necessity to use a *softmax* or an SVM as an output layer, even removing the fully connected layer.

Grad-CAM Introduced as a generalization of CAM, Grad-CAM [97] comes with much larger usability on different CNN models, for instance:

1. Characterized by fully connected layers, such as VGGNet;
2. Dealing with structured output, such as in the case of captioning;
3. When there are inputs with different modalities, such as visual question answering and reinforcement learning.

Good visual explanations must be *class-discriminative*, able to identify the desired class, and *high-resolution*, revealing even the details [97].
Grad-CAM retains gradient information passing through the network until the last convolutional layer to reveal the relevance of each neuron for a particular decision. The chief advantage is the possibility of visualizing whichever activation in a deep architecture.
However, networks without a connection between feature maps and outputs through weights require retraining.
Guided Backpropagation (see Section 1.2.5.1.2) and Grad-CAM can be fused using pointwise multiplication to improve the ability to show finer details.

1.2.5.1.5 CNN Eyes Vision

CNN Eyes Vision is a straightforward visualization method developed to highlight the most relevant parts of the input for CNN decision-making process [106, 107]. This method was one of the central topics of the Ph.D. research conducted by Edouard Villain, supervised by Marie-Véronique Le Lann, and Xavier Franceries, one of my Ph.D. supervisors [106]. Given the direct application in the medical field, developing a visualization technique adapted to 3D MRI data became essential to accompany the results of a CNN-based approach to discriminate patients with MSA from controls.

Fig. 1.36 provides a scheme of the proposed visualization method. First, we extract the output of each filter in the convolutional layers. Then, we remove negative values, as they do not carry useful information. To match the input dimension, we perform bicubic interpolation obtaining the activation map. Applying a threshold to activation values can facilitate visual interpretation. Finally, we compute the average of activation maps from each convolution layer. Averaging activation maps from each convolutional layer leads to a single visualization map for the considered model.
To highlight salient regions, we perform the absolute difference between the averaged maps per class. Alternatively, we can consider the difference between correctly classified samples per class to reduce the noise due to misclassifications.

This technique has been compared to Grad-CAM and saliency maps, showing similar results [106]. In addition to pixel-wise activation values, there is no constraint on model

applicability, and no retraining is needed. Furthermore, it takes a lower computation time compared to Grad-CAM.
Using CNN Eyes Vision to inspect CNN predictions for the classification of 3D brain MRI data showed the expected target regions in the visualization maps [107].

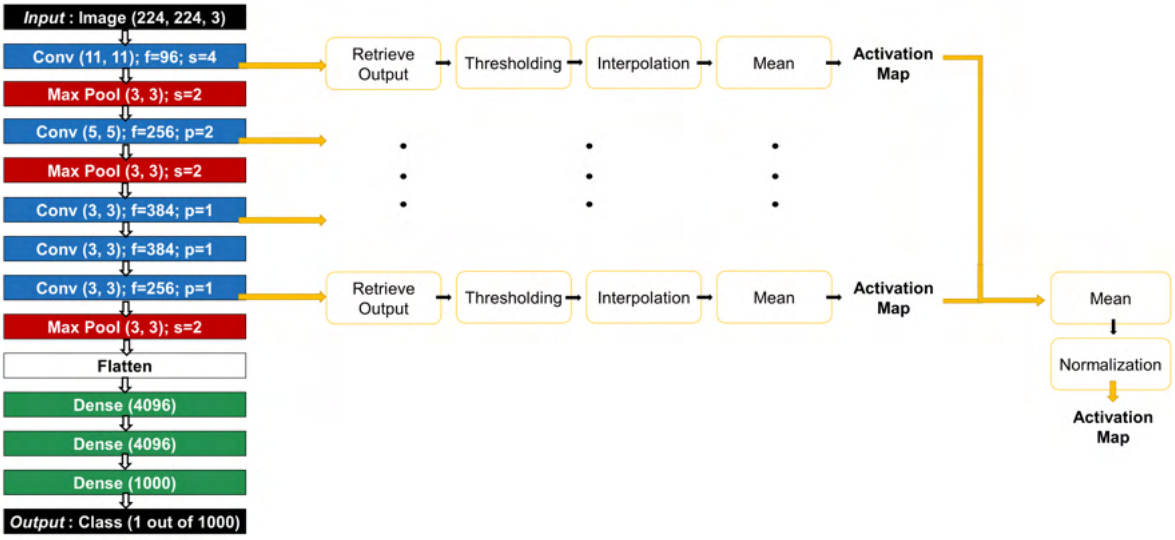


Figure 1.36: Scheme of CNN Eyes Vision applied to AlexNet architecture. The output from each convolutional layer is retrieved to be thresholded and interpolated to the input dimension. Activation maps for each convolutional layer are obtained by averaging the results from each convolutional filter. Normalizing the mean of all activation maps provides the final activation map [106, 107]

1.3 AI for Neuroimaging

Though by no means exhaustive, this section provides an overlook of some insights into the application of AI in neuroimaging. The mind cap illustrated in Fig. 1.37 can guide you through the covered topics.

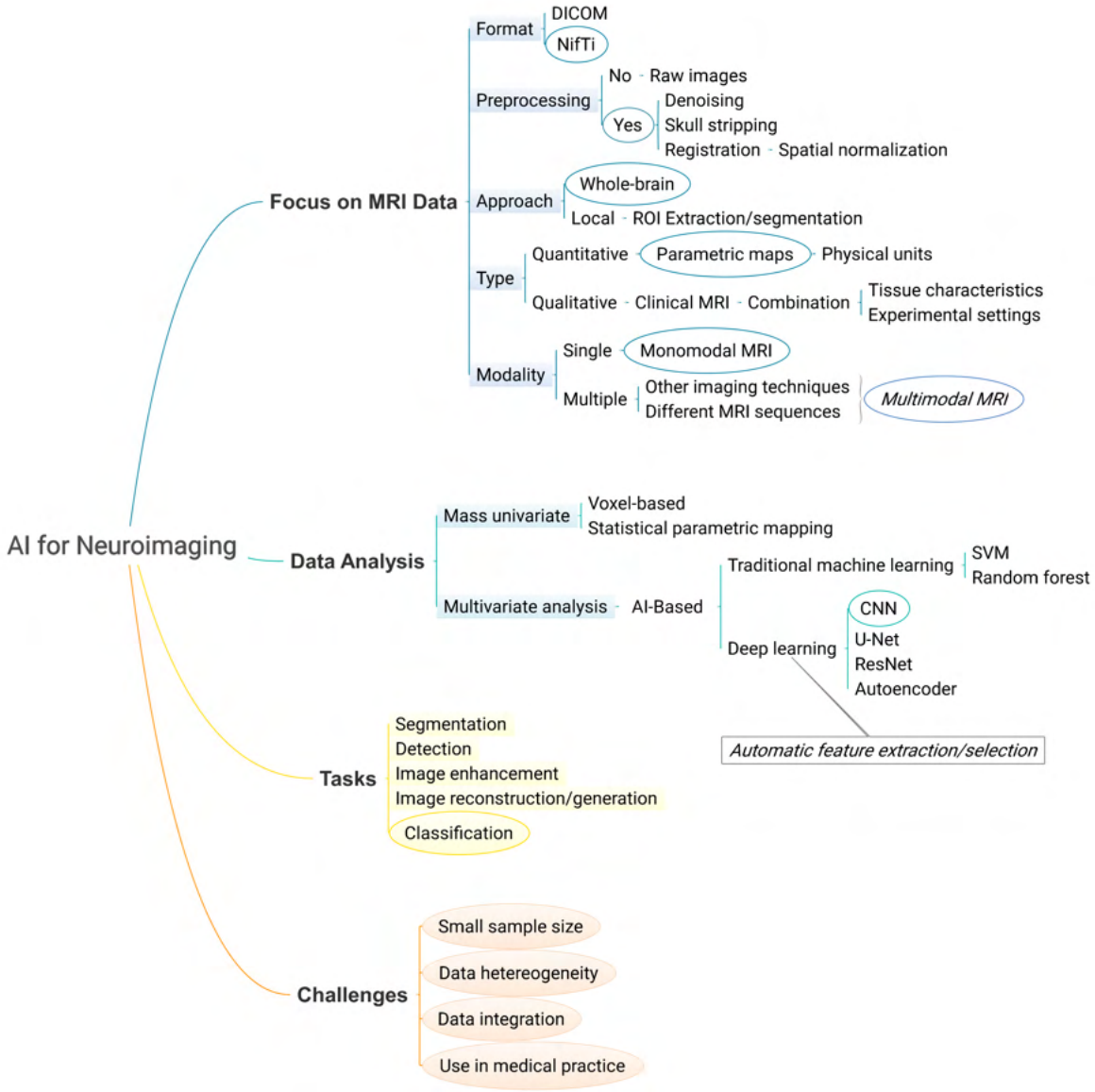


Figure 1.37: Mind map providing an overlook of the topics covered in the section *AI for Neuroimaging*. Concerning the experimental part of this dissertation, we focused on the topics highlighted in the rounded boxes

1.3.1 Focus on MRI Data

We introduce here a brief parenthesis regarding image processing of brain MRI.
MRI images can be used in Digital Imaging and COmmunications in Medicine (DICOM) format, a standard for storage and transmission of medical data, including information about acquisition parameters [108]. DICOM images are two-dimensional as each corresponds to a slice acquired from the subject. It is possible to merge these 2D images to constitute a 3D volume, usually registered in Neuroimaging informatics Technology initiative (NifTi) format [109]. The latter is easier to use for image processing and analysis.
We can use MRI data in their raw form with no preprocessing or after some preprocessing steps, which include but are not limited to:

- *Skull stripping*, i.e. the segmentation of brain tissue;
- *Registration*, i.e. the process of aligning multiple images so that they are anatomically coherent, performing spatial normalization, for instance, by using a brain template such as Montreal Neurological Institute (MNI);
- *Denoising*, i.e. the process of removing or practically lowering the noise from the signal.

In neuroimaging, we can also find different practices to analyze an image, going from whole-brain to more local approaches. For example, it is common practice to extract specific cerebral structures of interest, such as Gray Matter (GM) or White Matter (WM) maps, to conduct separate and more targeted analyses [110, 111].
Depending on the sequence type or other acquisition settings, MRI data may be affected by a variable degree of noise. The latter impacts image quality and the consequent potential extraction of biomarkers. DL techniques can offer a valid alternative to cope with this issue by developing automated and reliable tools [112].

An aspect worth mentioning is the difference between *qualitative* and *quantitative* MRI. As explained in a previous report [113], most clinical MRI acquisitions are qualitative since they convey *weighted images* with contrast determined by experimental parameters and tissue characteristics. Abnormality detection is achievable through the localization of focal or evident contrast differences in areas that should be normal. On the other hand, quantitative imaging refers to maps featuring a physical or chemical variable expressed in physical units. Consequently, they allow for comparison between regions and among individuals.
More systematic use of quantitative MRI could reveal beneficial to increase the diagnostic power of brain MRI based on quantitative measurements. The latter would enable comparing

a single individual to a healthy population's standards and keeping track of the alterations indicating disease progression [113].
In this dissertation, we focused on a quantitative parametric map, called MD, derived from diffusion-weighted images and informing about water molecules diffusion (for more details, see Section 3.1).

AI-based methods appear suitable for gathering information from multiple MRI indices [114, 115]. Given the diversity of knowledge offered by the various MRI indices or the combination with other imaging techniques, multimodal Magnetic Resonance Imaging (mMRI) has shown tremendous promise.
mMRI can benefit from multiple MR indices to reach a more reliable diagnosis by error compensation [116–118]. However, mMRI presents some drawbacks: it is complex, time-consuming, and subject to the reader's interpretation. Hence, the need to find a way of efficiently and automatically merging and examining these data. Our research group has recently proven the effectiveness of mMRI to discern patients with PD from patients with MSA and Healthy Controls (HC), devising a fully automated data-driven pipeline [41]. This approach overcame the limitations of previous works, based on single modality MRI or using intermediate user-dependent and bias-prone steps. Moreover, this fully-automated data-driven pipeline was compared to a CNN-based approach to discriminate MSA patients from HC with similar results [106].

1.3.2 Data Analysis

AI has been gaining ground in neuroimaging with applications from tumor segmentation to brain age estimation [119–122].
Machine learning techniques have enabled us to analyze and detect diffuse and diverse imaging patterns in contrast with *mass univariate analysis* [123]. Dating to the mid-90s, the latter comprises methods such as voxel-based analysis and statistical parametric mapping, which allowed for the characterization of cerebral functions and structure, leading to fundamental knowledge discovery [124]. The main goal of the mass univariate analysis is searching for differences between groups or correlations between data such as imaging, clinical, or cognitive [125], albeit at the population level. Although very informative, they cannot provide individually-based indices essential for developing and finding biomarkers at the subject level. The advent of ML revolutionized the neuroimaging field by directing its attention to a single-subject analysis. [125].

Multivariate methods can retrieve global signatures revealing especially suitable for application to MRI data, instead of only limiting at the voxel level as in mass univariate anal-

ysis [126]. Among the most widely used methods, random forests gained popularity given the reduced errors thanks to the ensemble of models merged by averaging or voting strategies [127]. They have been extensively applied for the classification of AD [128] or autistic spectrum disorder [129].
Another family of successful methods is the SVM, owing to the easiness of use and the wide availability of kernels in addition to very satisfying performances [31]. For example, patients with AD have been discerned from HC using the combination of many SVMs to randomly select and evaluate the previously extracted features, leading to a 94.4% accuracy [130]. Automatic classification of patients with autism spectrum disorder has been proposed using an SVM and cortical thickness data, reaching an accuracy of 84.2% [131].

1.3.3 Tasks

Despite the promising results of traditional ML methods, deep learning has recently shown great promise in neuroradiology for a range of tasks, e.g. for producing radiologists' reports using an image as input [132, 133]. To give a general idea of the possibilities and challenges, we briefly discuss just a few examples of these applications according to the task, focusing on classification. Exhaustive reviews on the applications of machine learning on neuroimaging data are available in [119–122, 125, 134, 135].

- *Segmentation.* DL-based segmentation of the brain, substructures, or even malignant lesions is widely employed, proposing several alternatives to the U-Net architecture (see Section 1.35) [136–138]. For instance, Natekar and colleagues [139] investigated the performance of three network architectures inspired by U-net [81] for brain tumor segmentation. They provided visualizations of the network's activations at different levels to prove that each architecture detected tumors at its own pace. Moreover, pyramidal CNN architectures with encoder-decoder-based modules allowed accurate identification of cancerous regions [140].
  Aggregating Grad-CAMs at different scales provided representations of hierarchical features from the tumors, outperforming Grad-CAM by 23% in localization accuracy.
  Another application saw the use of a U-Net architecture to fully segment the Substantia Nigra pars compacta (SNc) on neuromelanin-sensitive MRI to detect neurodegenerative changes due to the isolated REM sleep behavior disorder, considered a prodromal stage of parkinsonism [141].
  Comprehensive reviews are available in [142, 143].

- *Detection.* This task allows the localization of an object in an image, e.g. by contouring it with a rectangular box. Therefore, it can point out regions or abnormalities, often

followed by other tasks such as segmentation or classification. The first step is to find all anomalies, thus requiring high sensitivity, whereas the second is the classification of these patches. DL methods can be used for both or just one phase. An example is the detection of microbleeds from brain MRI with 3D CNNs [144].

- *Image Enhancement.* The main goal of image enhancement is to improve different aspects of an image, such as resolution or signal-to-noise ratio [132]. Among the most applied techniques, there are denoising (e.g. by using a residual CNN [145]) and super-resolution (e.g. creating synthetic MR images with enhanced resolution via GANs [146]).

- *Image Reconstruction/Generation.* Image acquisition is highly dependent on hardware and parameter settings which strongly affect image quality [132]. One possibility is to generate synthetic images varying the parameters within the same MR sequence or different MR sequences from other MRI sequences (e.g. using a fully convolutional neural network [147] or GANs [148]). Another promising application is image synthesis from other imaging techniques, for instance, by creating CT from T1-weighted brain images [149].

- *Classification.* DL techniques have shown promising results in analyzing neurological disorders [150, 151], including neurodegenerative diseases such as AD and PD [152, 153]. In particular, CNNs have led to exceptional performances for analyzing multidimensional images, as those issued by MRI (see Section 1.2.4.2).
  Khosla and coworkers [154] used a 3D CNN on fMRI data to discern autistic patients against healthy controls, reaching an accuracy of 72.8%. The computation of mean saliency maps highlighted relevant regions by averaging the results of each method across the adopted strategies. They also performed a regression task for age prediction and localized areas involved in aging via the saliency maps.
  In another study, an accuracy of 100% was achieved on a test set of 27 PD patients and 29 HC using a 3D CNN with six convolutional layers and brain MRI [155]. Occlusion sensitivity was applied to highlight salient regions for PD diagnosis, in line with medical findings.
  Korolev and coworkers [156] compared residual and plain 3D CNNs for the discrimination of healthy controls against AD patients. Both CNNs reached comparable performances (accuracy around 0.80). Attention maps revealed regions involved in AD physiopathology.
  In the multi-center study of Yuan and colleagues [157], a 3D CNN architecture was devised for gender classification, accompanied by visualization of the regions of inter-

est in each deconvolutional layer. Their implementation obtained an accuracy of over 92.5% by harnessing the issue related to data acquired with multiple MR scanners.

One advancement brought by DL is the possibility of analyzing the entire brain volume, feeding the networks with 3D images. Such an approach enables the integration of spatial information from 3D data, such as MRI, on a whole-brain level instead of considering only 2D slices [158]. Moreover, CNNs accept MRI data in their minimally processed form, usually after spatial normalization, e.g. in MNI space.
CNNs also allow for avoiding prior feature extraction and selection steps, sources of potential bias in the performance.

1.3.4 Challenges

The transition from traditional ML to DL techniques has been quite considerable in the past few years, leading to exciting findings [159]. Nevertheless, there are some concerns about whether deep networks can perform well when data are limited, as in medical applications. One aspect to consider is that compared to 2D images, like those used for the ImageNet competition, medical images exhibit fewer variations in their appearance [160]. Furthermore, recent studies showed that deep networks could develop the ability to generalize on unseen data even with a small sample size [161, 162].
One solution to deal with the lack of sufficient data may be to gather data from different medical centers. This strategy is prone to other issues, such as high variability in scanner settings, acquisition parameters, and heterogeneity in subject demography or disease manifestations.
An alternative other than standard data augmentation techniques (e.g. rotations, translations) is to generate artificial examples from a ground-truth distribution, using generative methods like GANs [50]. There already exist applications for medical image synthesis, including MRI and CT, holding the promise for future successful developments [163–165]. However, synthetic image generation requires a validation phase to assess image quality, as it is susceptible to artifacts and abnormalities [166, 167].

The work presented in this Ph.D. dissertation covers some of the abovementioned aspects, such as coping with a small sample size when using a CNN and better understanding the functioning of these powerful tools. To do so, we propose to create realistic synthetic brain MRI data to interpret CNN behavior according to data whose characteristics we master. Furthermore, we tested the validity of these synthetic data when enclosing features of a rare disease, such as MSA, and analyzed the impact of limited data on CNN performance.
Let us walk you through the objectives and motivation at the core of this research.

2 Objectives

This doctoral thesis aims to better understand the behavior of convolutional neural networks for the classification of 3D brain MRI data. Many like us have wondered about the reasons underpinning CNN's outstanding performances and abilities, being as much a curiosity as a necessity. That is especially true in the medical domain demanding transparency about the decisions regarding patients' life. We are well aware that as long as these methods will not be, if not utterly at least in part, understood, they will never take their place in clinical practice.

Some questions prompted us from the beginning:

- To what extent can specific input features shape CNN performance? Or, in other words, how much do data from each patient impact the CNN learning process?
- Do CNNs learn what we hope to perform a task? Do we have enough information about input data to exploit for a better understanding of CNN behavior?

To find an answer to these questions tightly related to one another, we decided to study CNN behavior by feeding ad hoc modified brain MRI input data. Our hypothesis stated that when we master the content provided to the network, we can interpret CNN performance more easily. We must bear in mind that these methods learn from experience, i.e. input data, albeit with possibly different criteria than us. Hence, we propose to alter brain MRI parametric maps belonging to healthy subjects in specific regions, thus creating the Altered Parametric Maps (APMaps). To this end, we applied a linear transformation to increase the intensity of these regions, keeping in line with the physical significance of the MRI sequence used without mimicking any particular pathology. Exploiting data from healthy subjects represented a considerable advantage because of the greater availability while preserving inter-individual variability. For this first phase, we could not afford to include pathological data with their intrinsic complexity since we needed to control the information given as input to the network. Building on the findings of a previous Ph.D. thesis exploring the ability of a CNN to discriminate healthy from pathological patients [106], we devised a 3D CNN to perform binary classification between original and altered parametric maps. We tracked CNN performance according to input changes and explored a more complex case with two altered regions in the input images.
Moreover, the APMaps may serve as ground truth to verify whether the network searches

for the known differences between the classes to discern. Indeed, we used them to validate a straightforward visualization technique to find the most relevant image parts for CNN prediction [107].
The first experimental chapter (Chapter 3) is dedicated to thoroughly explaining and discussing these findings. To our knowledge, this is the first attempt to study CNN behavior in the case of 3D neuroimaging data by using targeted input modifications.

Inevitably, the next step was the analysis of pathological data, driven by these issues:

- Can we use these altered brain MRI data to identify similar traits in unseen pathological data?
- May creating APMaps with specific features of a rare disease improve disease classification via controlled data augmentation?
- Can a CNN be capable of generalization when trained with a small dataset?

The second experimental chapter (Chapter 4) is devoted to answering these questions by focusing on MSA, a rare neurodegenerative disorder whose similarities with PD complicate differential diagnosis [14, 168].
Our goal was to efficiently discriminate patients with MSA from healthy controls despite the small sample size. To do so, we first examined the value of pathology-agnostic APMaps to detect similar features in MSA patients. We then refined this method to create APMaps containing specific MSA features with different approaches.
Secondly, we directed our analysis toward the importance of training content when it comes to discriminating MSA patients from normal individuals. Besides considering the limited quantity of data, we analyzed the impact of data heterogeneity on the classification of a rare disease such as MSA. By grouping patients according to a z-score-based approach, we trained a CNN with patients featuring different degrees of modification to track CNN performance accordingly

To conclude this dissertation, we discuss in the last experimental chapter (Chapter 5) an application of CNNs to the discrimination between healthy controls and patients in coma. We considered a multimodal MRI protocol comprising structural and functional MRI. Furthermore, we supported our findings with the localization of the most salient regions for the prediction. This work represented one of the first attempts to study coma patients using a deep learning approach and data from multimodal MRI.

3 Altered Parametric Maps for CNN Interpretability

3.1 Introduction

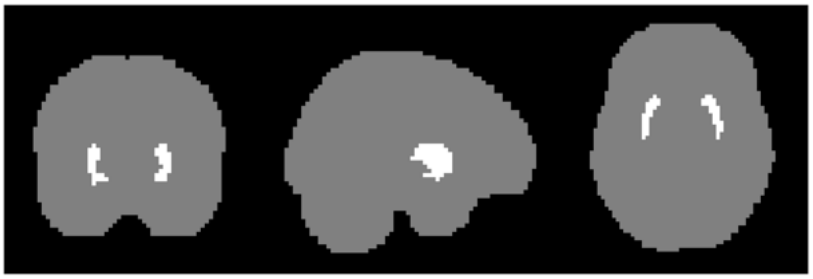
No matter how enlightening visualization methods can be, they cover only marginal CNN aspects or sub-parts [88]. We reviewed some of these methods in Section 1.2.5. Furthermore, we can examine CNN behavior by considering architecture, learning rules, and objective functions [169].
As one can well imagine, there are infinite possibilities to better grasp CNN behavior, e.g. by testing different structures or developing novel explanation techniques. For this experimental part, we chose to keep the proposed CNN architecture constant while focusing on the content of training data. We hypothesized that mastering the input could facilitate the interpretation of CNN results specifically applied to 3D brain MRI data. To this end, we modified brain parametric maps of healthy individuals by altering the intensity of two specific regions. Let us illustrate the reasons and motivations leading us to this approach.

In neurological disorders, brain alterations can present complex patterns owing to several regions involved with varying pathophysiological changes [170]. Learning from these data may thus reveal quite challenging as we cannot know how each patient contributes to the CNN pattern retrieval.
Our research group has concretely faced these difficulties in interpretation during the doctoral thesis of Edouard Villain, under the supervision of Marie-Véronique Le Lann and Xavier Franceries, one of my Ph.D. supervisors [106]. This research assessed the feasibility and effectiveness of a CNN-based approach to discriminate MSA patients against HC using mMRI by comparing it to the fully automated data-driven pipeline based on a traditional ML approach from our research group [41]. Focusing on the classification of MSA patients against HC, the CNN-based approach achieved comparable performances considering the different combinations of MRI modalities. Moreover, a visualization technique was developed to investigate the network's decisions and verify that the CNN had based its decisions on similar regions to the reference pipeline (see Section 1.2.5.1.5). Despite the reassuring correspondence between the regions considered by the network and the reference pipeline, we acknowledged that some aspects, such as prediction errors, remained obscure owing to the unknown component proper of pathological data. The latter are intrinsically heterogeneous

due to variegated patterns characterized by a single, sometimes considerably altered, region or more diffuse alterations.
In light of this, we postulated that being aware of the content provided to the network could help infer how input characteristics influence CNN performances. Hence, we created altered brain MRI data by introducing region-specific modifications.
Many valuable alternatives are available to generate new data, including, for instance, deep networks like GANs. The only drawback is that their functioning is not transparent as in the case of CNNs. Indeed, that was the issue we were trying to overcome, so we decided to modify existing data from healthy subjects by introducing calibrated yet realistic alterations.

The present study aimed to ascertain whether it is feasible to analyze CNN behavior according to changes in input data. To do so, we modified brain MRI parametric maps derived from DWI acquired from healthy subjects, with linear intensity-based alterations to two brain regions, the cerebellum and putamen. We named these altered data APMaps in contrast to the original data called Original Parametric Maps (OPMaps).
We chose mean diffusivity, a type of parametric map providing information about the Brownian motion of water molecules [7]. This index expresses the mean voxelwise diffusion of water molecules, quantitatively measured in mm^2/s [171, 172].
MD values can increase due to pathophysiological changes and have been observed in MSA [14], PD [13], and AD [12]. These increases usually indicate water diffusion anomalies and reduced microstructural integrity [7]. We did not conceive these alterations to resemble any specific pathology, but we kept them realistic in light of the physical significance of the chosen parametric map. As evoked in Chapter 2, this approach allowed us to establish a ground truth while exploiting the intrinsic inter-individual variability of the healthy subjects.
Fig. 3.1 illustrates the cerebellum and putamen, selected as regions of interest, given their distinct characteristics summarized in Table 3.1.

(a) Cerebellum



(b) Putamen

Figure 3.1: Atlas-based masks (in white) highlighted over the brain (in gray) for localization of the regions considered in the creation of the Altered Parametric Maps (APMaps)

	Region	
Characteristic	*Cerebellum*	*Putamen*
Size	300 cm^3	3.6 cm^3
Position	Underneath the brain hemispheres, surrounded by gray matter dorsally (the occipital lobe), and by the meninges and cerebrospinal fluid ventrally and posteriorly	Base of the forebrain, surrounded mainly by white matter
Morphology	Single rounded structure	Bilateral rounded structure
Tissue composition	Gray and white matter	Mainly gray matter

Table 3.1: Main differences between the cerebellum and putamen, the two brain regions selected for creating the Altered Parametric Maps (APMaps). The reported average size was retrieved from [173] for the putamen and [174] for the cerebellum

Several brain diseases can cause anomalies in these regions, including movement disorders and cognitive dysfunctions [175–177]. For instance, cerebellar ataxia and putaminal alterations can be found in the pathophysiological pattern of MSA [14, 168, 178–180]. Recent studies have focused on putaminal biomarkers to distinguish PD from atypical syndromes [181].

In addition to *monoregion* APMaps with only one altered region, we produced *biregion* APMaps, including two modified regions. Without loss of information about input characteristics, biregion APMaps made us approach a more complex condition in which more than one cerebral area presents abnormalities. As it often occurs in neurodegenerative diseases, pathological data can comprise more widespread alterations involving different regions.
To investigate CNN behavior given the changes in input characteristics, we trained the designed 3D CNN to distinguish between OPMaps and APMaps in a binary classification task. We supported these results by computing visualization maps to search for the targeted regions.
A preprint version of this work is available online [182]. In the following, we explain in detail the creation of APMaps and how we used them to interpret CNN behavior.

3.2 Material and Methods

3.2.1 Participants and MRI Protocol

A total of 89 individuals (100% male) underwent brain imaging in a 3T MRI scanner (Philips Achieva) with a 32-channel head coil at the INSERM/UPS UMR1214 ToNIC technical platform (Toulouse, France). Mean age of the participants was 56.19 years (SD = 18.08 years, range = 20.67-85.25 years).
DWI acquisition parameters were as follows: TE = 55 ms; TR = 12.36 s; flip angle = 90°; FOV = 112×112 voxels; number of slices = 65; voxel size = 2×2×2 mm^3; EPI factor = 59; parallel factor = 2; phase encoding direction = postero-anterior; *b* value (number of directions) = 0 (1), 500 (32), 1000 (32) s/mm^2; total acquisition time = 16 min.
This study was approved by the local ethics committee and was conducted following the Declaration of Helsinki. All participants gave written informed consent. For more information, please refer to previous work [183].

3.2.2 Image Processing

We processed DW images with the standard FSL pipeline [184]. We computed MD maps and registered them in MNI space with a resolution of 3×3×3 mm^3. Spatial harmonization

can help compensate for the anatomical differences in the native MRI space [185].

3.2.3 Creation of APMaps

We devised a method for introducing region-specific alterations to brain MRI parametric maps, thus creating the APMaps.

Fig. 3.2 summarizes the main steps of our approach. First, we extracted the regions of interest from the MD maps of healthy subjects (i.e. the OPMaps) using an atlas [186]. Second, we applied a linear transformation to MD values, as in (3.1). $y_{r,n}$ is the altered region (indicated by r) and $x_{r,n}$ is the original region, whose MD values lie below the n^{th} percentile, whereas p is the intensity increase as a percentage (3% to 99%, in increments of 3%). In each image, we modified only the region of interest, leaving the rest unaltered.

$$y_{r,n} = (1 + p) \cdot x_{r,n} \tag{3.1}$$

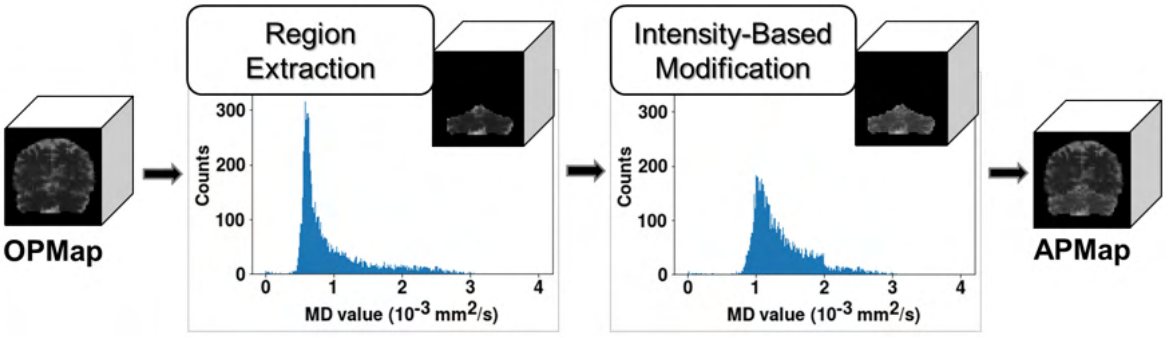
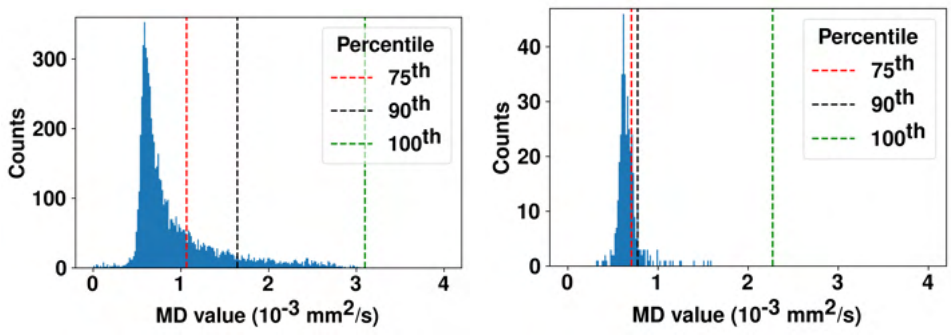


Figure 3.2: *Creation of the APMaps*. To create an APMap, we first extracted the region of interest from the OPMap, corresponding to the MD map of a healthy subject. We then applied a linear intensity-based transformation to increase each MD value of a percentage in the range [3%, 99%]. The resulting APMap presents only the region of interest modified, leaving the rest of the image unaltered. APMaps: Altered Parametric Maps; MD: Mean Diffusivity; OPMaps: Original Parametric Maps. Adapted from [182] (Giulia Maria Mattia, 2021). CC BY-NC-SA 4.0

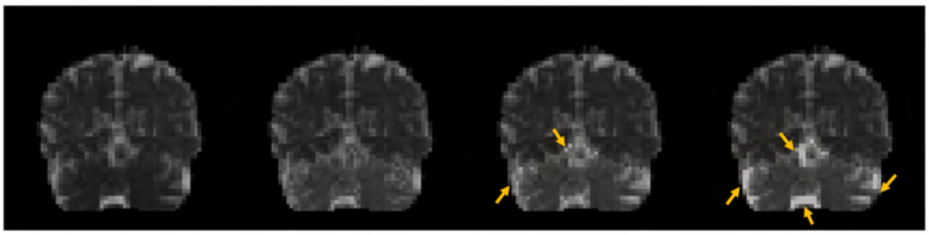
We evaluated the 75^{th}, 90^{th}, or 100^{th} percentile to limit image saturation effects. Finally, we chose the 75^{th} for the cerebellum and the 90^{th} for the putamen. Fig. 3.3 shows examples of histograms for the two regions.



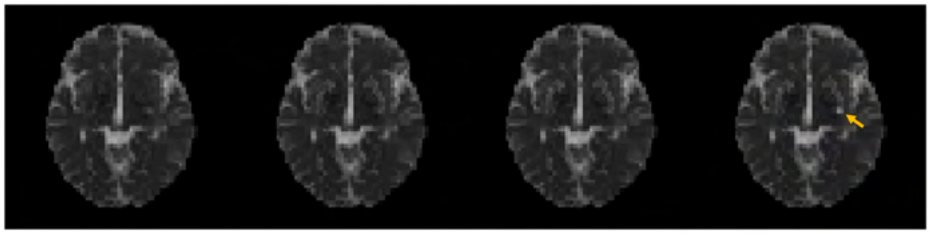
(a) Cerebellum (b) Putamen

Figure 3.3: *Creation of APMaps.* Examples of histograms computed on mean diffusivity (MD) values with considered percentiles for each brain region. We used 256 bins to calculate the histograms. Reproduced from [182] (Giulia Maria Mattia, 2021). CC BY-NC-SA 4.0

Image saturation can occur when, due to some operation, voxels overcome the maximum image value. Signs of saturation in the cerebellum already appeared using the 90^{th} percentile as a threshold, whereas only with the 100^{th} percentile in the putamen (see Fig. 3.4). We can also notice that the putamen is not delineated in the OPMaps, so when considering healthy individuals in this particular type of parametric maps. We can observe the presence of gray and white matter in the cerebellum.



(a) Cerebellum



(b) Putamen

Figure 3.4: *Monoregion APMaps.* From left to right: OPMap and APMaps created using the 75^{th}, 90^{th}, and 100^{th} percentile as a threshold to limit image saturation. We applied an intensity increase of 75% to both regions. Arrows indicate areas showing saturation. Reproduced from [182] (Giulia Maria Mattia, 2021). CC BY-NC-SA 4.0

Along with the intensity, we examined whether the position of the modified region could impact CNN performances. To this end, we harmonized region size to obtain a comparable number of modified voxels. We performed the following morphological operations on the respective atlas-based masks:

- Erosion of the cerebellum (Eroded Cerebellum (E-Cerebellum)) to reach a size comparable to the putamen (about 400 voxels, given our resolution in MNI space);
- Dilation of the putamen (Dilated Putamen (D-Putamen)) to reach a size comparable to the cerebellum (about 7200 voxels, given our resolution in MNI space).

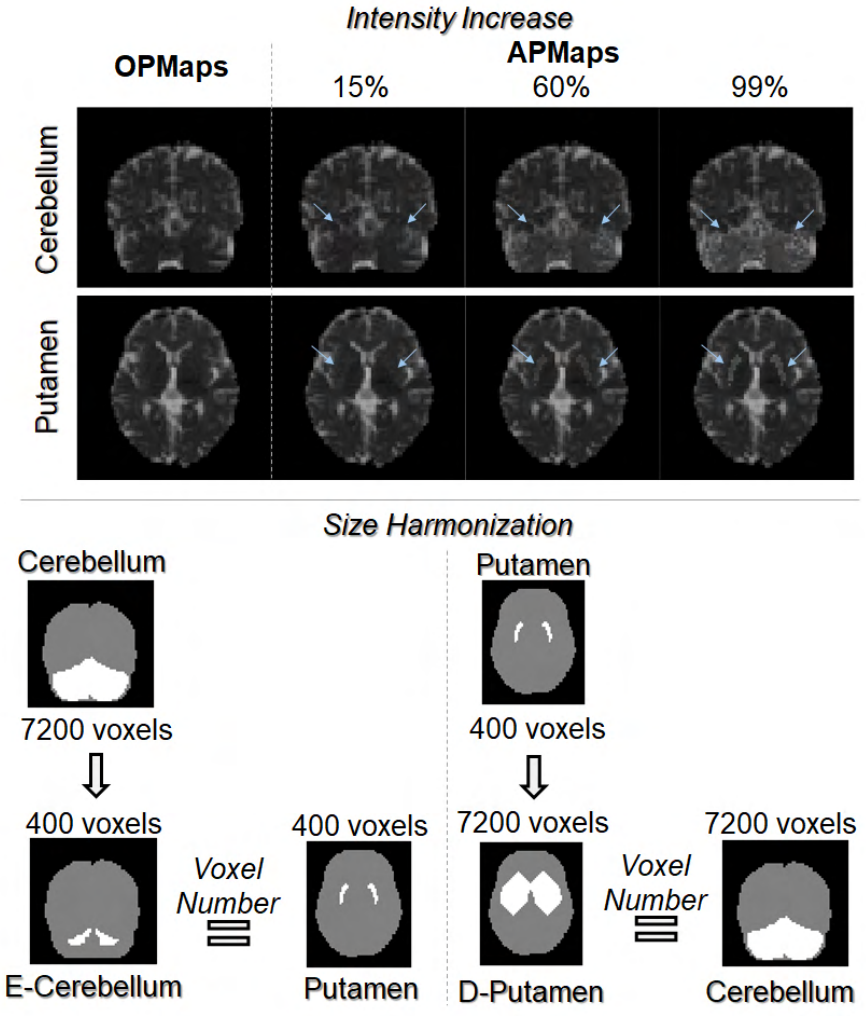


Figure 3.5: *Top*: Examples of APMaps with different intensity increases in percentage. Arrows indicate the altered regions. *Bottom*: Size harmonization for the brain regions with the corresponding number of voxels in each mask. The brain is displayed in gray, and the relevant region in white. APMaps: Altered Parametric Maps; D: Dilated; E: Eroded; OPMaps: Original Parametric Maps. Adapted from [182] (Giulia Maria Mattia, 2021). CC BY-NC-SA 4.0

Fig. 3.5 gives a schematic representation of the intensity modification and regional size harmonization. We altered region size only to compare with the anatomical reference with no intention of resembling any pathological trait.
We named *monoregion APMaps*, the APMaps presenting one altered region, and *biregion APMaps*, those with two altered regions. The former guided us in the creation of the latter getting thus closer to a more complex yet realistic brain condition. Section 3.2.5 further describes biregion APMaps.

3.2.4 CNN Implementation

Due to the promising performances achieved by the 3D CNN proposed in the previously mentioned doctoral thesis to classify MSA patients against HC [106], we considered that model a reference and performed some preliminary tests. These exploratory experiments led us to design a similar CNN architecture characterized instead by a smaller size for the convolutional filters, which seemed more sensitive to variation in region size. Further insights are available in Appendix A, Section A.1. In this work, we proposed a 3D CNN for supervised classification, the task being to distinguish OPMaps from APMaps for performance assessment according to input changes. Using the entire brain volume as input preserves the spatial information of the whole MRI at a 3D participant level [187].
Fig. 3.6 summarizes the main steps of the proposed approach.
The CNN received the images (i.e. 89 OPMaps and 89 APMaps) in the shape of (60, 72, 60) voxels. Given the limited sample size, we carried out cross-validation as customary in the neuroimaging field [187, 188]. We randomly split each dataset to use 80% for training and validation with 10-fold cross-validation and 20% as a hold-out set to assess CNN performance in the testing phase. The random seed for cross-validation was kept constant.
We normalized data considering the maximum value of the training set for each fold to lie in the range [0, 1]. We selected the best-epoch model with minimum loss value on the validation set and tested it on the hold-out set.
To design our 3D CNN architecture, we took inspiration from AlexNet [33] and VGG-Net [77], considering also a model already used for the discrimination between MSA patients and HC 107. Fig. 3.7 offers a schematic diagram of the proposed model, comprising the following building blocks [182]:

- *ConvBlock*, composed of a convolutional layer characterized by filter size = 3×3×3, stride = 1, with an increasing number of kernels going deeper into the network, and a batch normalization (BN) layer to speed up learning through a reduction in internal covariate shift [69], followed by an exponential linear unit (ELU) as the activation function [**?**];

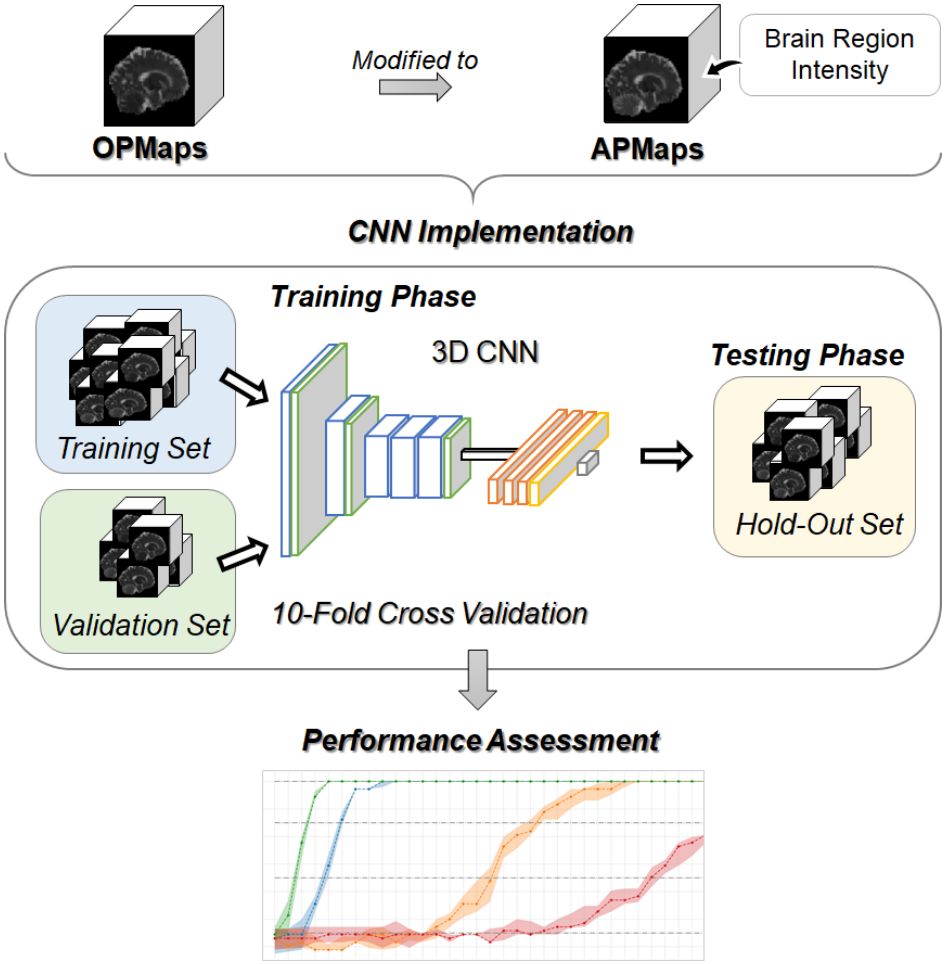
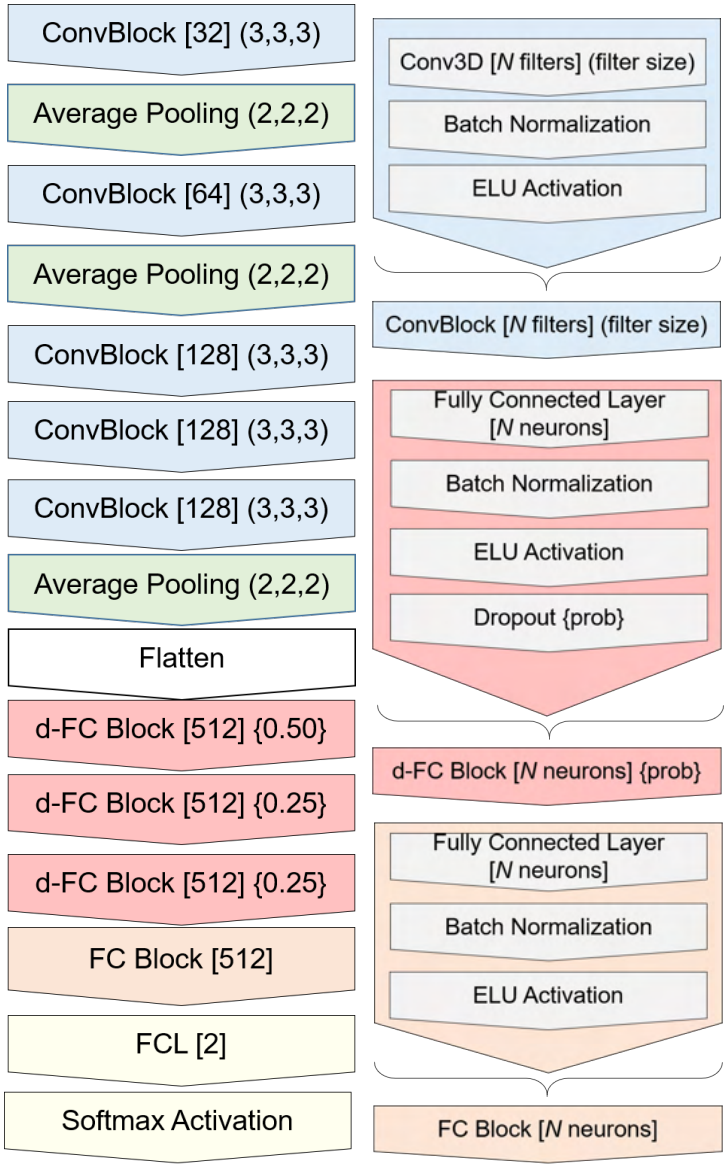


Figure 3.6: *APMaps for CNN Interpretability.* Representative scheme of the proposed approach. We modified brain MRI parametric maps of healthy individuals to create the APMaps by introducing linear intensity-based alterations to specific regions of interest in the OPMaps. We split the dataset composed of the original and altered parametric maps, thus obtaining the training set and validation set from a 10-fold cross-validation scheme and a hold-out set for the testing phase. We devised a 3D CNN to distinguish APMaps from OPMaps. Using the APMaps with different regional intensity increases as training data helped assess how CNN performance varied according to changes in the input. APMaps: Altered Parametric Maps; CNN: Convolutional Neural Network. MRI: Magnetic Resonance Imaging: OPMaps: Original Parametric Maps. Adapted from [182] (Giulia Maria Mattia, 2021). CC BY-NC-SA 4.0

- *Average Pooling*, to retain as much information as possible throughout the network, with filter size = 2×2×2 and stride = 2;
- *d-FC Block*, including a Fully Connected Layer (FCL) with 512 neurons to ensure that enough units were available for the final classification, followed by a BN layer, an ELU activation, and a dropout layer, as part of a regularization technique intended to prevent overfitting [55];
- *FC Block*, same as *d-FC Block*, but without dropout;

- *FCL*, a fully connected layer for binary classification with two neurons, followed by the softmax activation function.



(a) Architecture (b) Building blocks

Figure 3.7: Architecture and building blocks of the proposed 3D CNN. FC layers receive as input a one-dimensional layer obtained with the flatten operation. BN: Batch Normalization; CNN: Convolutional Neural Network; ELU: Exponential Linear Unit; FC: Fully Connected; FCL: Fully Connected Layer; prob: dropout probability. Reproduced from [182] (Giulia Maria Mattia, 2021). CC BY-NC-SA 4.0

We implemented the model using Keras library version 2.2.4 [189] and TensorFlow library version 1.13.1 [190], supported by an NVIDIA® Quadro RTX™ 6000 GPU.
We applied L2 regularization with a factor of 0.0005. We set the valid method for convolutional layers to avoid padding [189]. We trained the model as follows:

- Over 100 epochs to prevent overfitting, with an initial learning rate of 0.00005, subject to dynamic reduction if there was no improvement in performance after five epochs;
- Using mini-batch gradient descent, with a batch size of eight samples to meet computational requirements:
- Using categorical cross-entropy (i.e. logarithmic loss function) and the Adam optimizer [67].

To assess model performance, we used the accuracy defined in (1.17), given as median and Interquartile Range (IQR) over the ten folds.

3.2.5 Experiments

As a first step, we assessed CNN performance using OPMaps and monoregion APMaps as input, with intensity increases between 3% and 99% for each region.
Reasoning about biregion APMaps, we realized that trying all possible combinations of intensity increase would have been time-consuming and not necessarily informative. Therefore, instead of blindly considering the intensity increase, we decided to exploit the performance from monoregion-trained CNNs to guide our experiments.
To do so, we established four levels of accuracy corresponding each to a reference value: Very Low (VL) = 0.45, Low (L) = 0.65, Fair (F) = 0.85, and High (H) = 1.00. To create the biregion APMaps, we combined regions according to their size and the accuracy levels achieved by the CNN trained with the respective monoregion APMaps. For brevity's sake, we defined *monoregion-trained* and *biregion-trained CNNs* according to the input data (i.e. monoregion or biregion APMaps, always paired with OPMaps).

Biregion APMaps presented two modified regions, paired according to their size: either different (i.e. Cerebellum/Putamen) or comparable (i.e. D-Putamen/Cerebellum and E-Cerebellum/Putamen). Altering two brain regions led us to a more complex yet realistic pathologic condition, still mastering the content of training data.
We associated monoregion-trained CNNs with each accuracy level by considering the closest value they achieved. If needed, we computed additional intensity increases in 1% increments to match the accuracy levels as closely as possible.
When the same accuracy value (e.g. equal to 1.00) corresponded to different intensity increases, we chose the one with the highest minimum accuracy across the ten folds presenting the lowest intensity increase.
We produced biregion APMaps by applying the method described in Section 3.2.3 using the

intensity increase corresponding to the accuracy level and region dictated by the monoregion-trained CNNs.
To discover the contribution of each accuracy level to the CNN pattern retrieval, we compared monoregion-trained CNNs with biregion-trained CNNs by testing monoregion-trained CNNs on biregion APMaps and vice versa. To do so, we analyzed two cases:

- Biregion-trained CNNs considering the H/H accuracy combination and tested on monoregion APMaps, with intensity increases dictated by the corresponding H accuracy level;
- Monoregion-trained CNNs considering intensity increases dictated by the H accuracy level and tested on the corresponding biregion APMaps with the H/H accuracy combination.

3.2.6 Visual Interpretation

To show a concrete application of the APMaps as ground-truth data, we computed the visualization maps by applying the straightforward visualization technique developed by our group [106, 107] (see Section 1.2.5.1.5).
Each visualization map resulted from the average map obtained over the folds for training and test sets. Visualizations show the absolute difference between the mean of the correctly classified samples per class normalized beforehand to highlight salient regions for CNN prediction.
We provide two different points of view:

1. We considered monoregion-trained CNNs reaching an accuracy level equal to low (L: 0.65) and high (H: 1.00) and biregion-trained CNNs with L-L and H-H combinations, as representative examples.
2. Following the same scheme for testing monoregion-trained CNNs on biregion APMaps and vice versa (Section 3.2.5), we computed visualizations for each model on the corresponding testing images.

3.3 Results

3.3.1 Monoregion-Trained CNNs

We examined monoregion-trained CNNs considering increasing intensity increases in the APMaps for each region, as shown in Fig. 3.8. Results from the additional intensity increase are available in Appendix A, Section A.2.

Cerebellum and D-Putamen CNNs behaved similarly, although the former obtained its best performance with a higher intensity increase than the latter (27% vs. 15%).
The Putamen CNN achieved an accuracy of 1.00 at 84%, whereas the E-Cerebellum CNN only reached an accuracy of 0.81 at a 99% intensity increase. Despite comparable region size, the E-Cerebellum CNN only overcame near-to-chance accuracy with a 75% intensity increase (vs. 45% for the putamen).

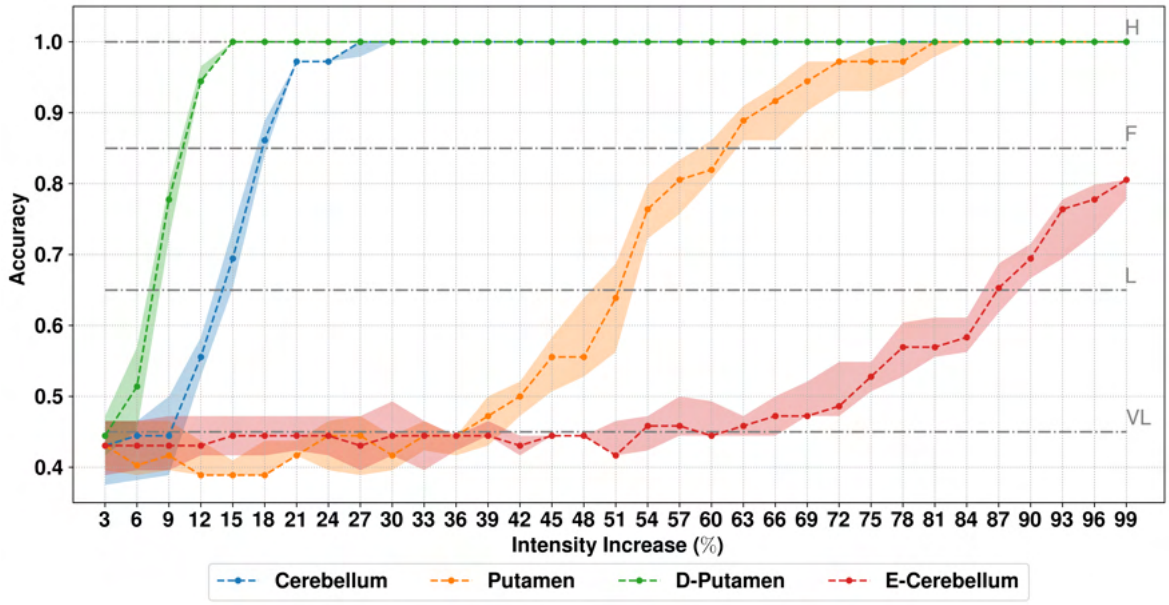


Figure 3.8: *Monoregion-Trained CNNs.* Accuracy on the hold-out set given as median and IQR obtained from a 10-fold CV according to intensity increase in the APMaps. Gray lines indicate the four accuracy levels used for performance assessment. APMaps: Altered Parametric Maps; CNN: Convolutional Neural Network; D: Dilated; E: Eroded; F: Fair; H: High; IQR: Interquartile Range; L: Low; VL: Very Low. Reproduced from [182] (Giulia Maria Mattia, 2021). CC BY-NC-SA 4.0

In Appendix A, Section A.2.2, we provide additional insight into monoregion-trained CNNs by progressively increasing training set size. We found that the greater and more intense the modified region, the less training data are necessary to obtain satisfactory performances.

3.3.2 Biregion-Trained CNNs

Table 3.2 reports the intensity increases corresponding to the accuracy levels for creating biregion APMaps.

Monoregion-Trained CNN	Accuracy Level							
	Very Low (VL)		Low (L)		Fair (F)		High (H)	
	Accuracy	Intensity Increase	Accuracy	Intensity Increase	Accuracy	Intensity Increase	Accuracy	Intensity Increase
Cerebellum	0.44 (0.11)	9%	0.63 (0.05)	14%	0.86 (0.05)	18%	1.00 (0.00)	36%
D-Putamen	0.46 (0.07)	5%	0.67 (0.08)	8%	0.85 (0.07)	10%	1.00 (0.00)	18%
Putamen	0.44 (0.03)	36%	0.64 (0.13)	51%	0.83 (0.03)	61%	1.00 (0.00)	96%
E-Cerebellum	0.46 (0.03)	63%	0.65 (0.07)	87%	0.86 (0.10)	105%	1.00 (0.00)	153%

Table 3.2: *Biregion-Trained CNNs.* Median accuracy (IQR) on hold-out set obtained with a 10-fold CV according to the assigned accuracy level and corresponding intensity increase used to create biregion APMaps. APMaps: Altered Parametric Maps; CNN: Convolutional Neural Network; CV: Cross Validation; D: Dilated, E: Eroded; IQR: Interquartile Range. Adapted from [182]

We trained biregion-trained CNNs to discriminate between OPMaps and biregion APMaps. Fig. 3.9 provides a comparison of the performances between biregion-trained and monoregion-trained CNNs according to the accuracy levels.
We found that biregion performance was significantly higher than the best monoregion accuracy for VL/VL, L/L, and F/F combinations, by computing unpaired Student *t*-tests. Combinations of accuracy levels with at least one H reached an accuracy equal to 1.00.

Biregion-trained CNNs systematically outperformed monoregion-trained CNNs for mixed combinations of accuracy levels, including F, L, and VL. F/VL for the D-Putamen/Cerebellum CNN and VL/F for the Cerebellum/Putamen CNN did not present significant differences.
We compared accuracy combinations using a one-way Analysis Of Variance (ANOVA), considering the results from biregion-trained CNNs represented by blue, green, and orange bars in Fig. 3.9.
VL/VL yielded significant differences with respect to the other combinations (e.g. VL/L vs. L/VL, VL/F vs. F/VL). Complete results for this analysis can be found in Appendix A, Section A.3.

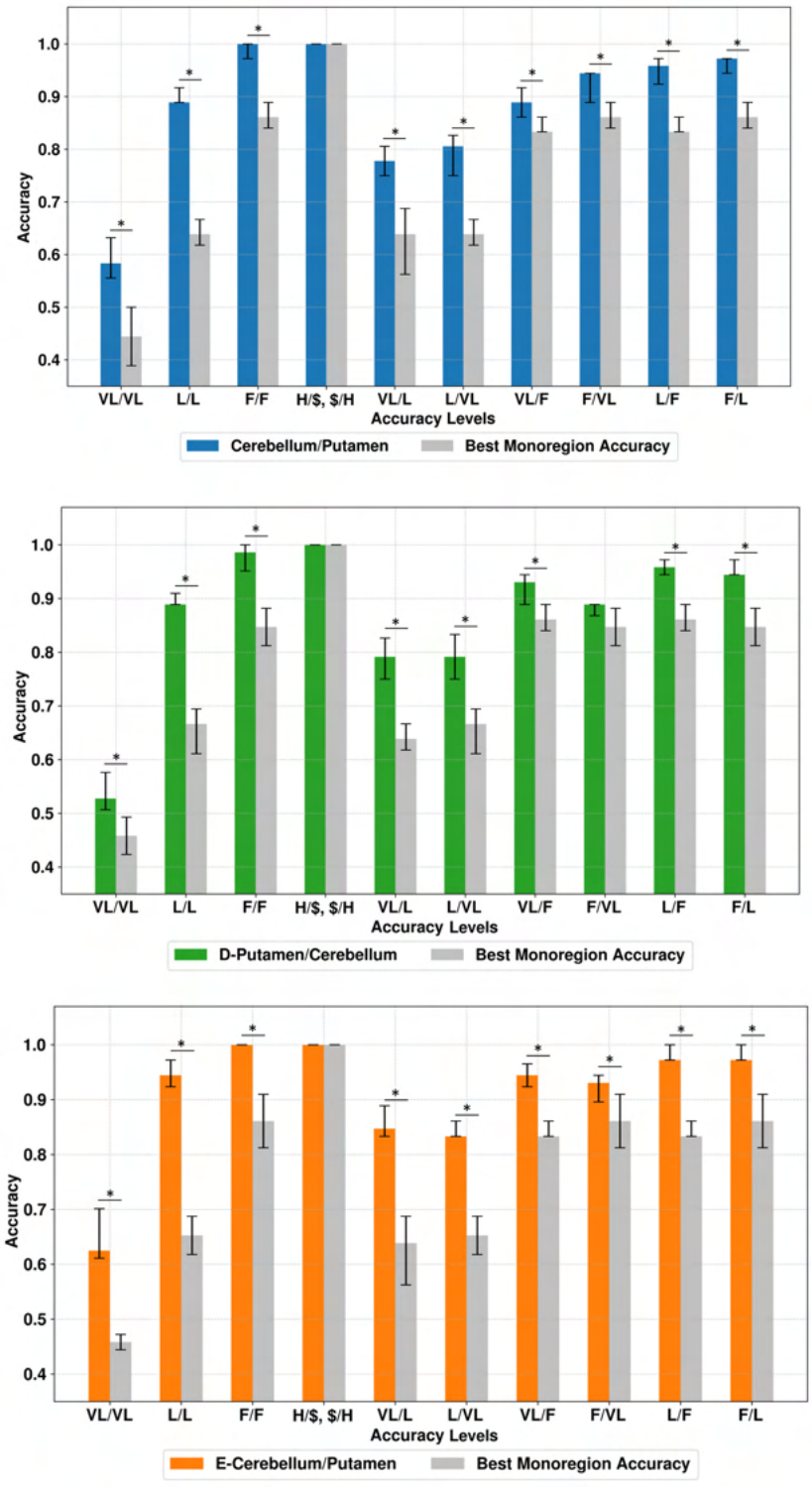


Figure 3.9: *Biregion-Trained CNNs*. Median accuracy and IQR on hold-out set obtained with a 10-fold CV compared with the best performance of monoregion-trained CNN. The dollar sign stands for VL, L, F, and H, as all combinations featuring at least one H yielded equal performances. $* \ p < 0.05$. CNN: Convolutional Neural Network; CV: Cross Validation; D: Dilated; E: Eroded; F: Fair; H: High; IQR: Interquartile Range; L: Low; VL: Very Low. Reproduced from [182] (Giulia Maria Mattia, 2021). CC BY-NC-SA 4.0

3.3.3 Monoregion- vs. Biregion-Trained CNNs

Table 3.3 reports the performances obtained testing monoregion-trained CNNs on biregion APMaps and vice versa. Except for E-Cerebellum and D-Putamen CNNs reaching 0.97 of median accuracy, monoregion-trained CNNs performed incredibly well on biregion images.

Table 3.3: *Monoregion- vs. Biregion-Trained CNNs.* Monoregion-trained CNNs with the H accuracy level were tested using the corresponding H/H biregion hold-out set of APMaps and vice versa. Accuracy is provided as the median (IQR) obtained with a 10-fold CV. Best performances are highlighted in italic. CNN: Convolutional Neural Network; CV: Cross Validation; D: Dilated; E: Eroded; IQR: Interquartile Range.

Training	*Testing*	
Monoregion-Trained CNN	**Biregion APMaps**	**Accuracy**
Cerebellum	Cerebellum/Putamen	1.00 (0.00)
Putamen		1.00 (0.00)
E-Cerebellum	E-Cerebellum/Putamen	0.97 (0.03)
Putamen		1.00 (0.00)
D-Putamen	D-Putamen/Cerebellum	0.97 (0.03)
Cerebellum		1.00 (0.00)
Biregion-Trained CNN	**Monoregion APMaps**	**Accuracy**
Cerebellum/Putamen	Cerebellum	0.97 (0.02)
	Putamen	0.50 (0.02)
E-Cerebellum/Putamen	E-Cerebellum	0.64 (0.08)
	Putamen	0.65 (0.03)
D-Putamen/Cerebellum	D-Putamen	0.56 (0.03)
	Cerebellum	0.89 (0.10)

Biregion-trained CNNs instead performed differently according to the considered regions. For instance, D-Putamen/Cerebellum CNN could classify the cerebellum with a median accuracy of 0.89, whereas it was not capable of doing so with the D-Putamen images.
Biregion-trained CNNs with E-Cerebellum/Putamen APMaps yielded the poorest performance achieving only median accuracy of 0.65 on both monoregion images.
The Cerebellum/Putamen CNN achieved median accuracy of 0.97 on cerebellum APMaps but performed poorly on putamen APMaps.
In addition, we compared monoregion- and biregion-trained CNNs, considering biregion APMaps modified using the same intensity increase for the two regions. Results are provided in Appendix A, Section A.4.

3.3.4 Visual Interpretation

Fig. 3.10 shows visualization maps according to the region and accuracy level.

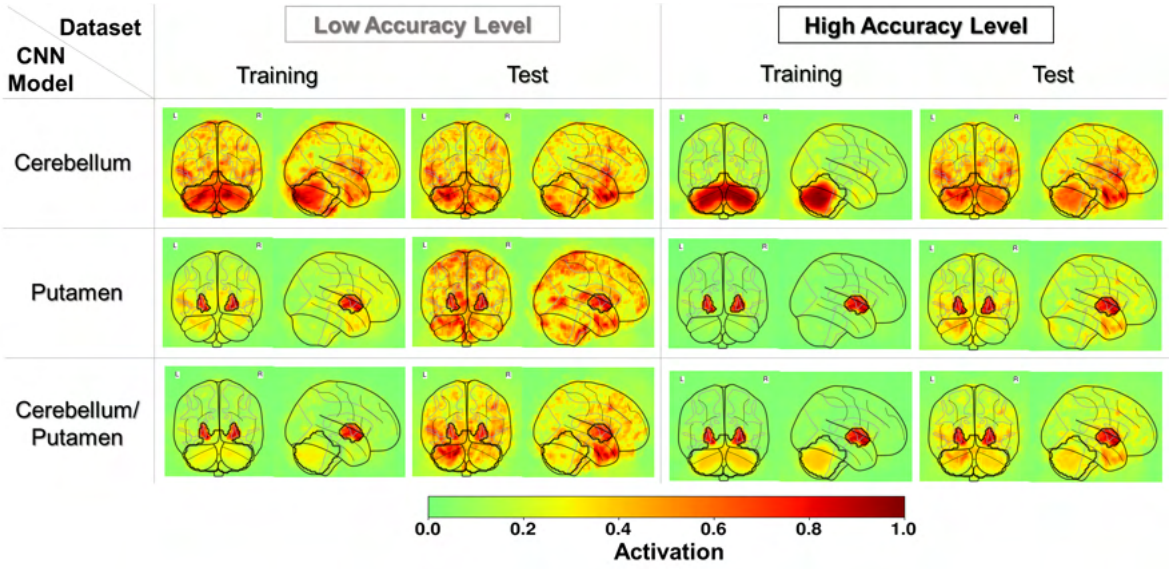


Figure 3.10: *APMaps for CNN Interpretability - Visual Interpretation.* Mean visualization maps showing the absolute difference between the average of correctly classified patients per class. We considered Low (L = 0.65) and High (H = 1.00) as accuracy levels per region. Black contours delineate the regions targeted in training. APMaps: Altered Parametric Maps; CNN: Convolutional Neural Network; CV: Cross Validation; D: Dilated; E: Eroded; IQR: Interquartile Range

Considering the low accuracy level, we can see that the targeted regions presented high activation values in training, albeit with activated outside voxels. In the test set, there was considerable noise, although the target region exhibited some activation.
Concerning the high accuracy level, we can observe cleaner visualizations with the regions of interest well delineated in the training and test sets, despite some noise in the latter.
Regarding biregion-trained CNNs regardless of the accuracy level, we can point out that the region presenting lower intensity increase presented lower activations, for instance, the cerebellum compared to the putamen considering the Cerebellum/Putamen model in Fig. 3.10. Similar considerations can be drawn for the E-Cerebellum and D-Putamen (results are available in Appendix A, Section A.5).

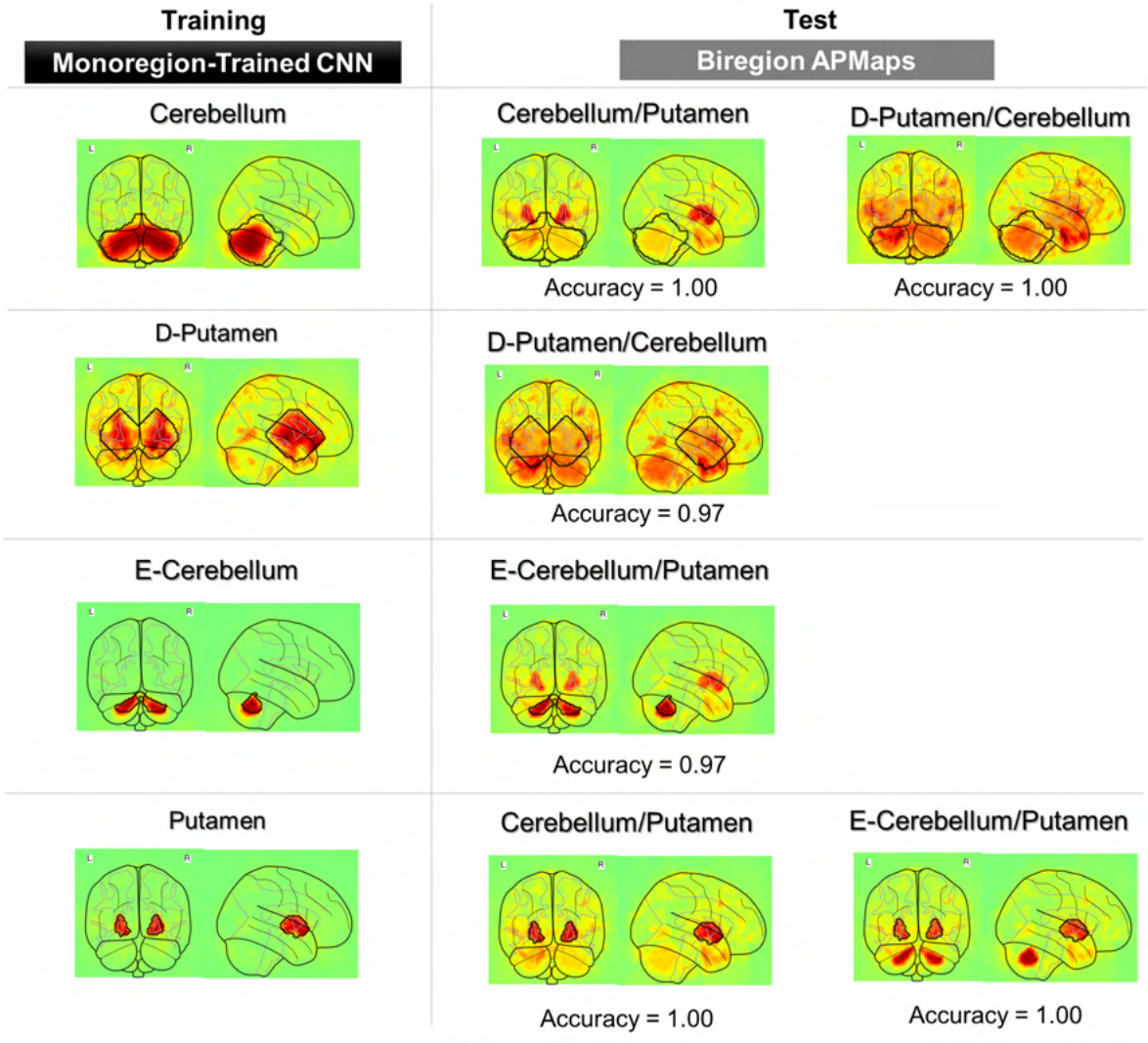


Figure 3.11: *APMaps for CNN Interpretability - Visual Interpretation.* Mean visualization maps showing the absolute difference between the average of correctly classified patients per class. We considered monoregion-trained CNNs tested on the corresponding biregion APMaps. Black contours delineate the regions targeted in training. APMaps: Altered Parametric Maps; CNN: Convolutional Neural Network; CV: Cross Validation; D: Dilated; E: Eroded; IQR: Interquartile Range

Considering monoregion-trained CNNs, we can observe from Fig. 3.11 that the region targeted in training was well highlighted. Looking at the visualizations from biregion APMaps, we found correspondence with the training region, albeit the other brain regions altered in the biregion APMaps presented high activation.

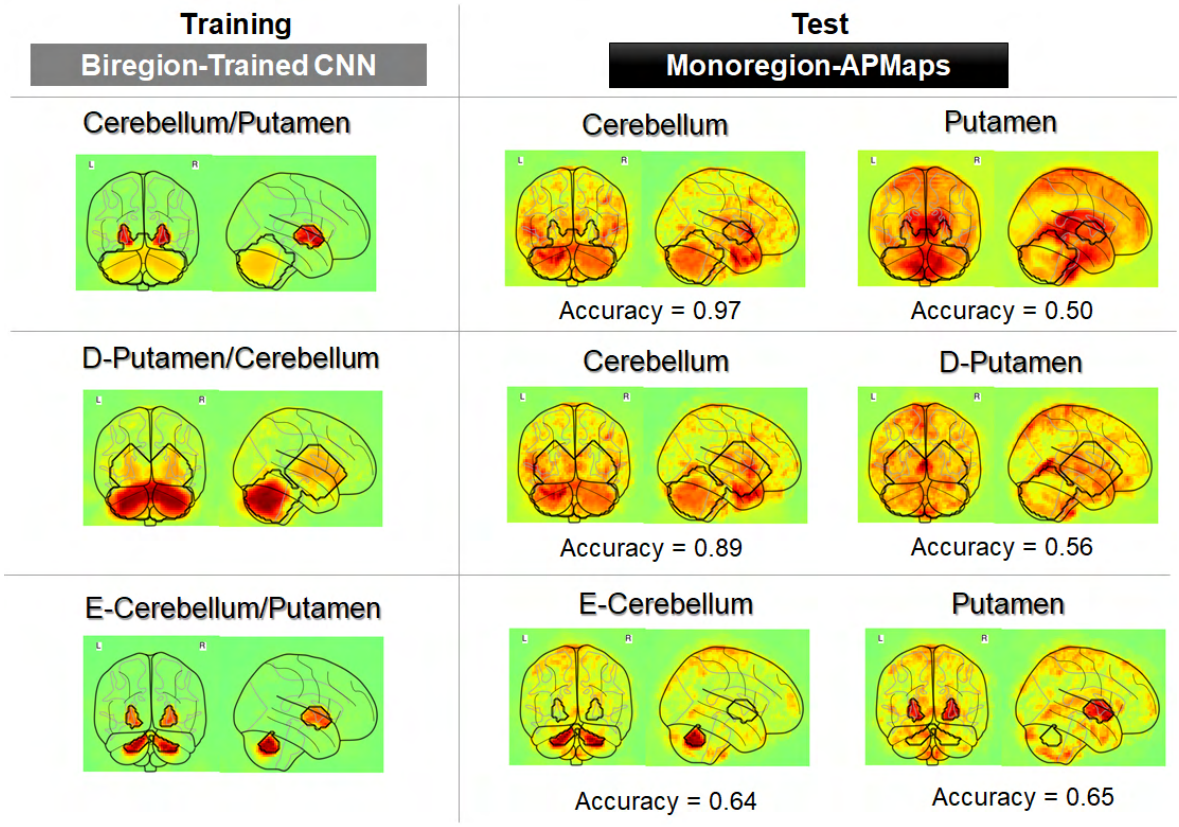


Figure 3.12: *APMaps for CNN Interpretability - Visual Interpretation.* Mean visualization maps showing the absolute difference between the average of correctly classified patients per class. We considered monoregion-trained CNNs tested on the corresponding monoregion APMaps. Black contours delineate the regions targeted in training. APMaps: Altered Parametric Maps; CNN: Convolutional Neural Network; CV: Cross Validation; D: Dilated; E: Eroded; IQR: Interquartile Range

Considering biregion-trained CNNs in Fig. 3.12, the region of interest in the monoregion APMaps was well highlighted in the putamen for the E-Cerebellum/Putamen CNN (despite an accuracy of around 0.65). The same goes for the cerebellum for Cerebellum/Putamen and D-Putamen/Cerebellum CNNs and for the E-Cerebellum in the E-Cerebellum/Putamen.
In the case of the Cerebellum/Putamen tested on the putamen APMaps, the visualizations actually represented the activations of all images, as only one class was predicted (accuracy = 0.50).
Concerning the D-Putamen APMaps tested by the D-Putamen/Cerebellum, there were diffuse activations in the area of the D-Putamen, despite the low accuracy (0.56).

3.4 Discussion

Insofar, we investigated the discrimination ability of 3D CNNs to discern original from altered whole-brain MRI parametric maps. We kept these alterations realistic by exploiting the inter-individual variability proper to the healthy subjects. By linearly modifying the intensity of one (monoregion) or two (biregion) brain regions, we showed how salient features of the input (such as size, position, and intensity) influenced CNN performance. Let us discuss these findings in greater detail.

In line with our expectations, monoregion-trained CNNs proved that the greater and more intense the altered region, the easier its discrimination. We showed that performances changed according to the position of the altered region despite the comparable number of modified voxels. E-Cerebellum and Cerebellum CNNs did not perform as well as their equally sized counterparts. In this regard, we can notice that the putamen and D-Putamen are more centrally located compared to the other regions. However, further experiments are in order to help clarify this behavior.
In addition, we examined the performance of monoregion-trained CNNs with gradually increasing training set size, considering cerebellum and putamen APMaps (see Appendix A, Section A.2.2). In line with our expectations, we observed that as the differences between the two classes to discern were more evident (higher intensity increase and bigger region), CNN performance considerably improved.
From these findings, we can perceive how sensitive the network is to input characteristics such as the intensity and position of the altered region. These results are more easily interpretable thanks to our knowledge of input data. Imagine how complicated this would be in the case of pathological data enclosing variegated and more heterogeneous information. Although relatively simple, the alterations obtained with the linear intensity increase of MD values respected the physical significance of MD maps by creating water diffusion anomalies.

Getting closer to a more complex situation, we examined the behavior of biregion-trained CNNs. The latter systematically outperformed their monoregion counterparts, even when the initial accuracy levels were low (< 0.65). Emblematic is the performance achieved by the L/L biregion-trained CNN with an accuracy of 0.90, exceeding by 23% the reference accuracy value. In general, combining two accuracy levels led to performance improvement (see L/L, VL/L, in Fig. 3.9), which was incredibly surprising in light of the poor performances of the monoregion counterparts. When at least one H was present in the accuracy combinations, performances remained high. We can also observe that by comparing mixed accuracy combinations, e.g. F/L vs. L/F, there was no significant difference in performance, suggesting

that the accuracy levels contributed to the biregion performance independently of regional characteristics.

Moving on to the comparison between biregion-trained with monoregion-trained CNNs, we found some results worth discussing.
We could suppose that modifying a second region in an image would increase the probability of detecting each singly by only implementing one network trained with biregion APMaps, instead of two separate networks training each with one type of monoregion APMaps. Consequently, we would be persuaded to assume an *additive* pattern retrieval for the CNN, e.g. if biregion-trained CNNs trained with biregion APMaps, they should be able to recognize each of the altered regions taken individually.
Monoregion-trained CNNs performed well on biregion APMaps, suggesting they looked for the region targeted during training. By contrast, performances of the biregion-trained CNNs on the monoregion APMaps degraded for at least one of the two regions. From this behavior, we can infer that regional characteristics impact CNN pattern retrieval in a non-predictable way. One hypothesis is that biregion-trained CNNs retrieved a multi-spatial signature absent from monoregion APMaps. Another possibility is that one region became more relevant to the prediction due to its more prominent features, such as size or intensity (e.g. for the same accuracy, the cerebellum presented a higher intensity increase than D-Putamen. Hence, biregion-trained CNNs correctly classified the cerebellum APMaps, as in Table 3.3).
These findings may be relevant to clinical research. We can interpret monoregion APMaps as representing early pathological conditions (with a single altered region), whereas biregion APMaps would be closer to more advanced states (with a more diffuse pattern). Indeed, neurodegenerative diseases initially involve one specific area and progressively spread to the entire brain [191].
We can therefore imagine that monoregion-trained CNNs may detect regional anomalies in earlier and later stages since they are blind to alterations outside their region of interest. On the other hand, biregion-trained CNNs would perhaps be less capable of detecting monoregion anomalies, given the more complex and interconnected characteristics of the learned patterns.

As aforementioned, supporting CNN outcome with a visual interpretation can shed light on CNN predictions. Visualizations maps confirmed the correspondence of the most salient regions between training and test sets. We can notice that by increasing the accuracy level visualizations become less affected by noise (i.e. voxels activated outside the regions of interest).
Worth mentioning is that the regions of interest were well highlighted in training despite an

accuracy of around 0.65 on the test set. However, the latter presented voxels with high activations outside the targeted regions, confirming the model's poor performance.
If we consider high intensity increases (see Fig. XIII in Appendix A, Section A.5), we can see that the activations are higher compared to lower intensity increases. The more intense the region, the more evident and unique the alteration becomes. Hence, we can suppose that convolutional filters detect fewer similar traits over the image. By contrast, when the intensity increase is lower, it is more likely that filters activate in other parts of the image due to inter-individual variability.
Nevertheless, these maps represent an average of the activations over all the samples, so we must interpret them with caution. One possibility could be looking at a single subject to investigate individual characteristics and understand CNN prediction.
Worth mentioning is that in the case of monoregion-trained CNNs testing biregion APMaps and vice versa, different areas activated outside the regions of interest, regardless of the good performances. This result underlines some discrepancy between visualization maps and performance assessment, a reminder that further work is needed to find a more reliable correspondence. Although methods, such as visualization techniques, can help get new insight into these black boxes, we must account for their limitations when we use them for interpretation purposes.

We are well aware that the alterations in the APMaps are oversimplified compared to the complexity of pathological data. Nevertheless, we showed that using the APMaps as training data can effectively find similar anomalies in pathological data. Section 4.3.1 provides additional insight into this approach.
In addition, we used the APMaps as ground truth data to validate the recent visualization technique developed for 3D neuroimaging data and described in Section 1.2.5.1.5. As expected, we found a correspondence between the areas highlighted by the method and those modified in the APMaps. We presented this work at the 2021 Institute of Electrical and Electronics Engineers (IEEE) 34th International Symposium on Computer-Based Medical Systems (CBMS) [107].
One may point out as a limiting factor of this study the restricted sample size, owing to the use of real-world brain MRI data. Although small compared to conventional DL datasets, ours ensured a realistic level of heterogeneity regarding individual characteristics, given the broad age range of the healthy population. Worth mentioning is that artificially generating data using methods such as GANs implies the presence of distortions or artifacts, which could negatively impact CNN performance [166, 167]. Moreover, it would be cumbersome to interpret black-box methods such as CNNs by using the results from another black-box approach such as GANs.

Creating APMaps with different target regions and types of MRI data may help customize CNNs and respond to specific concerns about why some patterns are easier to discriminate than others, considering the APMaps as ground truth. These findings are just the starting point to better grasp the influence of data complexity on CNN pattern retrieval.

3.5 Conclusion

In this chapter, we set the basis for a better understanding of CNN behavior applied to 3D brain MRI parametric maps by targeted modification of input data. These findings have enlightened us about the influence of specific input features related to the altered regions and how these changes can shape CNN performance.

The most surprising result was the performance of biregion-trained CNNs achieving an accuracy of over 0.90, albeit the original accuracy level of the corresponding monoregion-trained CNNs was around 0.65. That demonstrated to what extent different regions, apparently not informative separately, combined led to great performances.

Furthermore, visualization maps reassured that the CNN accounted for the expected differences between the two classes to assign the correct prediction. However, using a visualization method also warned about the discrepancies that may arise between CNN performances and interpretation results.

As evident as it can be, none of the previous discussions would make sense without our prior knowledge about input data. Given our exploration of CNN behavior in this controlled yet effective way, we moved to the natural progression of this work, which was the analysis of pathological data.

4 CNN for Multiple System Atrophy Classification

4.1 Introduction to Parkinson's Disease and Atypical Parkinsonism

Parkinson's disease is an idiopathic neurodegenerative disorder involving motor symptoms (e.g. tremor, imbalance, rigidity, bradykinesia) and non-motor symptoms, including depression and sleep disturbances [192, 193].
The first appearance of PD dates back to 1817, with the famous monograph written by James Parkinson, entitled "An essay on the shaking palsy" [194]. He described the presence of motor symptoms, such as progressive degeneration of motor functions and the characteristic resting tremor. Since then, advances in neuroimaging techniques have contributed to ameliorating the analysis and treatment of PD [195, 196].
According to the World Health Organization (WHO), PD is the most dominant movement disorder counting 8.5 million cases in 2019 [197]. Its prevalence has doubled in the past 25 years. Current treatments include levodopa/carbidopa as the most effective medication and deep brain stimulation for reducing tremors, albeit there is no definitive cure yet.
A definite assessment of PD can be confirmed only post-mortem [198]. However, creating guidelines for establishing a diagnosis and advising on treatment has been of great help to clinicians and health practitioners [199].

Atypical Parkinsonism comprises syndromes similar to PD but presenting with *atypical* features, such as recurrent falls or early dementia [200]. Besides a more rapid decline than PD, patients affected by these syndromes do not usually respond well to treatment with levodopa (one of the "red flags" which alerts about these rare syndromes).
One common aspect of neurodegenerative disorders is the abnormal accumulation of proteins in the brain, thus naming them *neuroproteinopathies* [200]. We can categorize neuroproteinopathies according to the accumulated protein:

- *Tauopathies*, involving abnormal deposits of the tau protein, such as Progressive Supranuclear Palsy (PSP) [201] and AD;
- *Synucleinopathies*, involving abnormal deposits of the α-synuclein protein, such as MSA, PD, and dementia with Lewy bodies [202].

Several studies aimed at distinguishing PD from PSP and MSA, exploiting machine learning techniques and MRI data [195, 203–205]. For instance, encouraging performances (accuracy $>$ 90%) were achieved using volumetry data for classification of PD versus atypical Parkinsonism or PSP, and MSA-C versus PD and PSP, and all Parkinsonism versus controls. Accuracy $>$ 80% can be found for the discrimination between PD-MSA-P and PSP and MSA-P. Biomarkers from MRI offer great potential to aid clinical practice, especially if coupled with easy-to-use tools to improve the diagnosis [195]. Structural MR sequences led to high classification accuracies [203, 206], as well as diffusion imaging.

Due to the similar symptoms in the early stage, the differential diagnosis between MSA and PD may be challenging [13, 168]. However, MSA prevalence goes from 3.4 to 4.9 per 100,000 people, whereas it amounts to 7.8 per 100,000 if considering a population older than 40 years [207]. The average onset age is about 55-60 years, with no difference between men and women [208, 209]. Disease progression is pretty rapid, presenting a survival rate after diagnosis between 6 and 10 years [210, 211].
MSA presents a variety of clinical symptoms [212]:

- *Extrapyramidal*, comprising motor symptoms similar to PD, such as bradykinesia, rigidity, and postural instability;
- *Autonomic*, including dysfunction of the autonomic nervous system, such as cardiovascular, urogenital, and gastrointestinal failure, sleep disorders, respiratory problems, and behavioral or emotional symptoms;
- *Cerebellar*, with gait and limb ataxia and cerebellar oculomotor dysfunction.

We can identify two subtypes of MSA based on the symptoms [212]:

- *MSA Cerebellar variant (MSA-C)*, mostly involving symptoms caused by alterations in the cerebellum;
- *MSA Parkinsonian variant (MSA-P)*, more similar to PD, including symptoms due to putaminal alterations.

MSA diagnosis can be ascertained only by post-mortem analysis of the brain. A recent revision of diagnostic criteria for MSA diagnosis is available in [213].

Extensive research has been conducted for the identification of radiological signs from MRI able to set apart these two variants [195]. Table 4.1 summarizes some of these findings. However, both variants share some imaging features, such as infratentorial atrophy in 61.1%

of MSA-P (22% with no atrophy in the putamen) and putaminal atrophy detected in 46.2% of MSA-C [214]. Alterations in the diffusion of water molecules have been reported with consistent overlap between MSA-C and MSA-P [215].

MSA Variant	Characteristic Signs	MRI Sequences	Abnormalities
MSA-P	Atrophic posterior putamen with lateral border flattening	*SWI and T2-weighted images*	Decreased signal intensity
		Proton-density and T2-weighted images	Hyperintense rim
		DWI	Increased signal intensity
MSA-C	Atrophy of cerebellum and pons	*Proton-density images*	Hot cross bun sign in the pons
		T2-weighted images	Hyperintense signal in the middle cerebellar peduncles
		DWI	Increased signal intensity

Table 4.1: Distinctive traits of the two MSA variants found with MRI. Exhaustive information is available in [180, 215, 216] for MSA-P and [217, 218] for MSA-C. DWI: Diffusion-Weighted Imaging; MSA: Multiple System Atrophy; MSA-P: MSA Parkisonian variant; MSA-C: MSA Cerebellar variant; SWI: Susceptibility-Weighted Imaging

It is important to note that the MSA patients considered for the present research were assigned an MSA subtype based on the symptoms, regardless of neuroimaging findings.
A recent review exhaustively examined several works based on the use of DWI for MSA characterization [219], identifying the cerebellum and putamen among the regions of interest. As a quantitative MRI parameter, mean diffusivity maps have also shown great potential for signaling gray and white matter widespread anomalies in MSA patients [14, 168].

4.2 AI for MSA Classification

Automated classification of MSA using machine learning methods coupled with brain MRI data has led to promising results [195, 203–205]. Biomarkers from MRI offer great potential to aid clinical practice, especially if coupled with easy-to-use tools to improve the diagnosis [195]. Particular attention has been given to the differentiation between PD, MSA,

and HC, or other atypical parkinsonism, in light of the distinct disease progression considerably affecting patients' care. Various studies have assessed the efficacy of a differential diagnosis based on MRI data and ML techniques. We briefly cite some representative examples to set the basis for our work.

Chougar and coworkers categorized parkinsonian syndromes using different ML algorithms (linear and radial SVM, random forest and logistic regression) trained on a research cohort and tested using an independent clinical replication cohort [203]. Data from 13 regions were extracted considering DTI and volumetry from HC (n=94), patients with PD (n=119), PSP (n=51) and MSA (n=35). Overall, performances with DTI were poorer than those with volumetry, the latter reaching balanced accuracies between 0.840 and 0.983 for PD vs. PSP, PD vs. MSA-C, PSP vs. MSA-C, and PD vs. atypical parkinsonism, whereas between 0.765 and 0.784 for PD vs. MSA-P and MSA-C vs. MSA-P. Logistic regression was the algorithm reporting the highest mean balanced accuracies. An added value to this study is the performed data harmonization in two strategies due to MRI acquisition from different scanners.
Scherfler and colleagues submitted a method based on volumetric data from T1-weighted MRI to discern between MSA (n=40), PD (n=40), and PSP (n=30), considering 22 subcortical regions [220]. The cerebellar gray matter, midbrain, and putaminal volumes were identified as the most relevant to the prediction model, a C4.5 decision tree, reaching an accuracy of 97.4% for PD vs. MSA or PSP.
To the best of our knowledge, applications for MSA classification involving neural networks are limited to the use of volumetric data [221] or medical information other than imaging-based features (e.g. neurological findings or data associated with the diagnosis of MSA subtypes) [222].

Our research group has extensively contributed to the characterization of MSA and parkinsonian syndromes with encouraging results. In the attempt to exploit the information delivered by mMRI, Barbagallo and colleagues proposed an analysis of multiple MRI indices to discriminate PD from MSA in the two variants [168]. One of the most remarkable results was that, by considering anatomical data, patients with MSA presented increased mean diffusivity in the putamen than PD patients.
Péran and coworkers developed a method featuring unsupervised self-organizing maps (a type of neural network for unsupervised classification) based on mMRI to group patients with PD and MSA [14]. The obtained clusters grouped patients faithfully to the clinical diagnosis. This study reported statistically significant multiparametric alterations in the cerebellum and putamen of patients with MSA compared to PD patients. However, the regions were retrieved by comparing the different groups of patients, so feature selection was not

completely unbiased. Inevitably, the next step was to develop a method capable of integrating mMRI data in a completely automatic, ideally bias-free, way.
Recently, Nemmi and coworkers proposed a data-driven integrative pipeline comprising an mMRI approach combined with a multivariate analysis [41]. The ML pipeline comprised different feature reduction steps, such as variance thresholding to eliminate features varying little among subjects and feature selection with the Relieff method based on the intra- and inter-class distances. Since adjacent voxels may belong to the same brain region, voxels were grouped with a spatial clustering technique. The total number of features was thus reduced from hundreds to tens, considering the average signal for each cluster. A cross-validated scheme with subset selection was implemented before fitting the model to automatically select the number and type of MRI modalities. The best accuracies were 0.78 for PD vs. HC, 0.94 for MSA vs. HC, and 0.88 for PD vs. MSA. Concerning specifically MSA vs. HC, the most selected MRI index was MD.
Finally, one of the latest works compared Nemmi's fully automated pipeline with a CNN-based approach for discriminating MSA patients against HC.This study was conducted by Edouard Villain during his Ph.D. thesis, supervised by Marie-Véronique Le Lann and Xavier Franceries [106]. Similar performances were obtained with the proposed 3D CNN directly fed with the different MRI indices adopting a 10-time 10-fold CV scheme. Moreover, to shed some light on CNN predictions, the visualization technique based on convolutional filter outputs revealed a spatial correspondence between the regions selected by Nemmi's pipeline and the CNN (further details in Section 1.2.5.1.5). This doctoral thesis has set the foundations for the work presented in the current dissertation by advancing the feasibility of a DL method despite the restricted sample size of the pathological cohort.

Building on the abovementioned findings, we pursued the analysis of MSA and its differentiation from healthy controls. One objective of the present doctoral research includes coping with a small sample size and data heterogeneity, affecting data from rare diseases, such as MSA. In the following chapters, we describe the two core phases characterizing this work:

1. *Utility of Altered Parametric Maps for MSA Classification* (Section 4.3). Building on our understanding of CNN behavior via the APMaps, we first exploited these altered data to identify similar traits in MSA patients, thereby discriminating them from HC. Besides using pathology-agnostic APMaps (see Section 3.2.3), we devised different pathology-oriented variants incorporating features from the disease. The main idea was to use the APMaps for training the CNN while keeping the MSA cohort as an external test set to assess the network's generalization ability. This approach allowed us to validate the use of APMaps as augmented data and to seek an improvement in

CNN performance.

2. *Impact of Small Sample Size on MSA Classification* (Section 4.4). This part begins with examining the influence of training set size, considering different CNN architectures for classifying MSA patients and HC. It will then go on to the effect of data heterogeneity on CNN pattern retrieval: according to the type of data used for training, the network's discriminating capacity varied, thereby explaining why some pathological patterns were more effective than others.

4.3 Utility of Altered Parametric Maps for MSA Classification

In the first experimental chapter of this dissertation (Chapter 3), we used the APMaps to better grasp CNN behavior by mastering the content of training data. However, we underlined that these *ad hoc* altered data represented a more simplified case because we introduced the alterations. By contrast, pathological data are intrinsically more heterogeneous and variegated, thus making interpretation more difficult.

As previously mentioned, the anatomical brain regions targeted in the APMaps corresponded to the regions of interest in MSA pathophysiology [14, 168]. Hence, we thought to exploit the APMaps as training data for the CNN while testing directly on the pathological cohort. We assumed that we could benefit from the altered data to guide the network toward specific regional anomalies, given the common involved regions and type of alteration (i.e. increase of MD values) between APMaps and MSA patients. It is necessary to point out that the APMaps described in Section 3.2.3 were completely pathology-agnostic since the applied intensity increase did not reproduce any specific pathological pattern. Nevertheless, this approach brought about encouraging results that we will discuss in Section 4.3.1.

Given the potential shown by the pathology-agnostic APMaps, we wondered about the possibility of introducing specific pathological features into the APMaps to cope with limited data and improve CNN performance.
Always considering the MSA patients exploited so far, we put forward two different strategies:

1. *Region-specific*, creating *Cluster-Based Altered Parametric Maps (CB-APMaps)* by exclusively modifying the cerebellum from brain MRI data of healthy subjects. First, we clustered MSA patients according to the distribution of MD values belonging to the cerebellum, and then we used these clusters to obtain a reference pattern. Section 4.3.2 presents this approach.

2. *Whole-brain*, creating *Z-score-Based Altered Parametric Maps (ZB-APMaps)* by modifying the entire brain volume, considering the z-score computed on MSA patients and healthy individuals. We provide all details in Section 4.3.3.

What follows is an account of the abovementioned approaches presented and described in details. Fig. 4.1 summarizes the highlights of the current section.

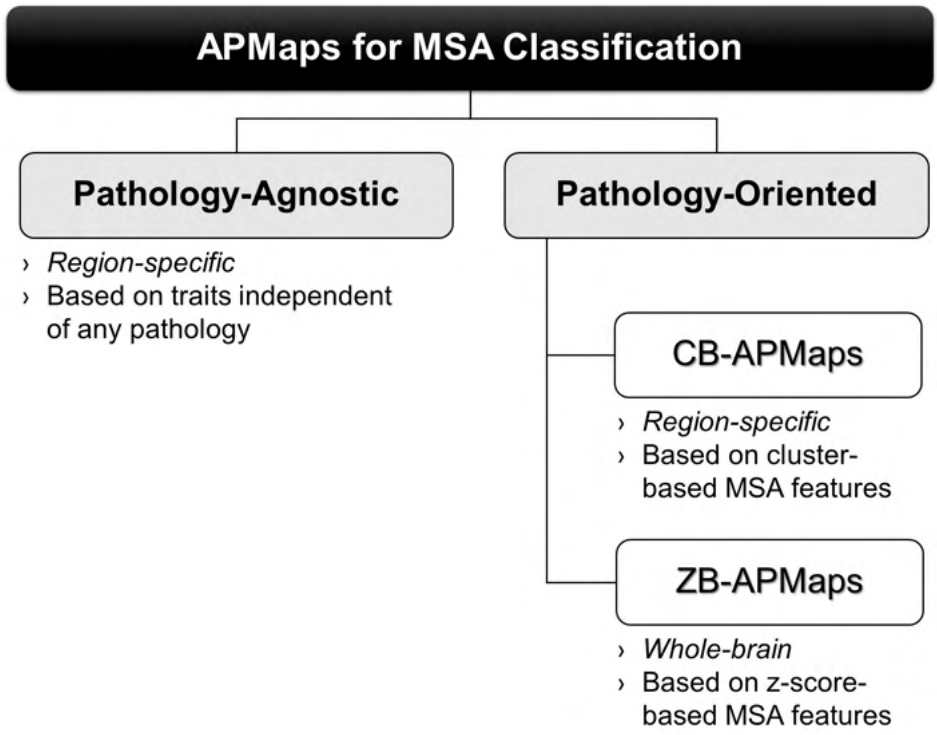


Figure 4.1: *APMaps for MSA Classification.* Diagram showing the different types of APMaps used for the classification of MSA patients against HC. APMaps: Altered Parametric Maps; CB-APMaps: Cluster-Based Altered Parametric Maps; HC: Healthy Controls; MSA: Multiple System Atrophy; ZB-APMaps: Z-score-Based Altered Parametric Maps

4.3.1 APMaps from Pathology-Agnostic Features

This section is concerned with determining the validity of pathology-agnostic APMaps as training data for a CNN to discern pathological from normal data. Given the extensive work of our research group on MSA [13,41,106], we focused on this rare neurodegenerative disorder which seemed the perfect candidate for the restricted number of samples and intrinsic heterogeneity.
Before delving into the technical part, there are two fundamental aspects to underline:

- For this approach to work, we require that the altered regions in the APMaps include regions of interest in the pathological data. That was the case for MSA patients affected

by pathological changes in the cerebellum and putamen, the same regions modified in the APMaps (see Section 3.2.3).

- There is a crucial advantage in creating the APMaps regardless of the MSA pathological patterns. The two datasets were completely independent, making it feasible to assess CNN's generalization ability.

Moving on to our method, we exploited the cerebellum and putamen APMaps with intensity increases in the range [3%, 99%] as training data for the proposed CNN. Our goal was to determine whether we could reach satisfactory performances on the unseen set of MSA patients and HC. To further support CNN performances, we employed the visualization technique described in Section 1.2.5.1.5 to find out the most discriminant voxels, thereby establishing a correspondence between the targeted regions.
We presented this work at the 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) [223]. A summary of the main findings, together with our take on the matter, is provided in the following.

4.3.1.1 Material and Methods

This study aimed to find out whether APMaps enclosing region-specific pathology-agnostic alterations could be used to detect similar traits in pathological data. However, the brain regions altered in the APMaps must coincide with those affected by the pathological changes. Indeed, patients with MSA can present high MD values in the cerebellum and putamen [14, 168], corresponding to the regions modified in the APMaps.

4.3.1.1.1 Datasets

To ensure coherence in data preprocessing, we treated both datasets uniformly, as reported in Section 3.2.2.
Henceforth, we may refer to each dataset considering only the positive class, either APMaps or MSA.

MSA/HC Pathological data comprised MD maps from a set of 26 healthy controls and 29 MSA patients (13 MSA-C and 16 MSA-P). MSA-C and MSA-P variants were assigned according to the symptoms, regardless of radiological findings from MRI data. This study was granted approval from the Toulouse Ethics Committee (ID RCB 2012- A01252-41) and conducted following the ethical principles of the Declaration of Helsinki and relevant guidelines and regulations. Participants provided written informed consent.

Previous works report further details about this cohort and the MRI acquisition protocol [14,41].

APMaps/OPMaps We used the same dataset and method to create the APMaps described in Section 3.2.3. We considered the Cerebellum and Putamen APMaps with an intensity increase from 3% to 99%.

4.3.1.1.2 CNN Implementation

We employed the 3D CNN, whose architecture and implementation are described in Section 3.2.4.
Fig. 4.2 offers a schematic diagram with the main steps.

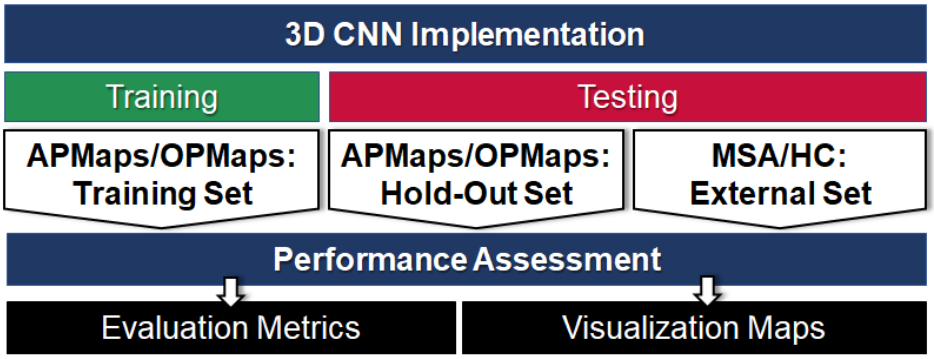


Figure 4.2: *Pathology-Agnostic APMaps for MSA Classification.* Schematic diagram of the proposed approach. We trained a 3D CNN to distinguish APMaps from OPMaps. We tested this network on a hold-out set of APMaps/OPMaps and an external set comprising patients with MSA and healthy controls (MSA/HC). We assessed performance using evaluation metrics such as accuracy and provided visualization maps to highlight the most discriminant voxels. APMaps: Altered Parametric Maps; CNN: Convolutional Neural Network; HC: Healthy Controls; MSA: Multiple System Atrophy; OPMaps: Original Parametric Maps. Adapted from [223]

We organized the experiments as follows:

- *Training.* We used Cerebellum and Putamen APMaps with intensity increases in the range [3%, 99%] as training data, adopting a 10-fold CV on 80% of the APMaps/OPMaps set, leaving the rest for testing.
- *Testing.* We evaluated CNN's performance on the:
  - Hold-out set of APMaps, to ensure that each network was capable of correctly classifying data from the same distribution as in training by testing on the left-out set of APMaps;

- MSA/HC set, to determine the feasibility of this approach and whether some intensity increases yielded good performances.

To support these results, we used the visualization method presented in Section 1.2.5.1.5 to ascertain that the expected regions of interest corresponded to the most discriminant areas identified by the CNN.
To select the best performances, we considered accuracy of at least 0.90 on the hold-out set, given the highest accuracy on the MSA/HC set. Similarly to what we did in Section 3.2.3, we created biregion APMaps considering the intensity increase relative to the best-accuracy models for the cerebellum and putamen. By doing so, we ascertained whether combining the two regions leading to the best results could ameliorate the performances of the biregion-trained compared to the monoregion-trained CNNs.
Besides accuracy, we evaluated performances on the MSA/HC set by computing the sensitivity, as in (1.18), to determine the number of correctly classified MSA patients and the specificity, as in (1.19), to quantify the number of correctly classified healthy controls.

4.3.1.1.3 Visual Interpretation

We employed the CNN Eyes Vision technique to highlight the most discriminant voxels [107]. As previously discussed, deep networks' decisions need to be accompanied by comprehensible and convincing explanations, especially in the biomedical domain [224].
We obtained visualization maps for each convolutional layer, computing a unique map per model by averaging and normalizing results considering the filter number of the convolutional layer. To show only meaningful areas to the prediction, we calculated the absolute difference between the averaged maps of correctly classified samples per class (i.e. TN and TP, obtained from the confusion matrix). We applied no threshold on activation values to retain as much information as possible.
We computed visualization maps for the best-accuracy models, considering the entire set of MSA patients and the distinction between MSA-C and MSA-P.

4.3.1.2 Results

4.3.1.2.1 CNN Performance

Fig. 4.3 provides CNN performance on the two datasets with median and IQR per intensity increase and region of interest. Considering the APMaps/OPMaps set, we found the discrimination of cerebellar alterations to be easier than putaminal ones, as exhaustively reported in Section 3.3.1. Cerebellum/Putamen CNN achieved maximum accuracy on the APMaps/OPMaps hold-out set.

We can identify a range of intensity increases leading to satisfactory performances on the MSA/HC set per region: [21%, 42%] for the cerebellum and [54%, 72%] for the putamen.

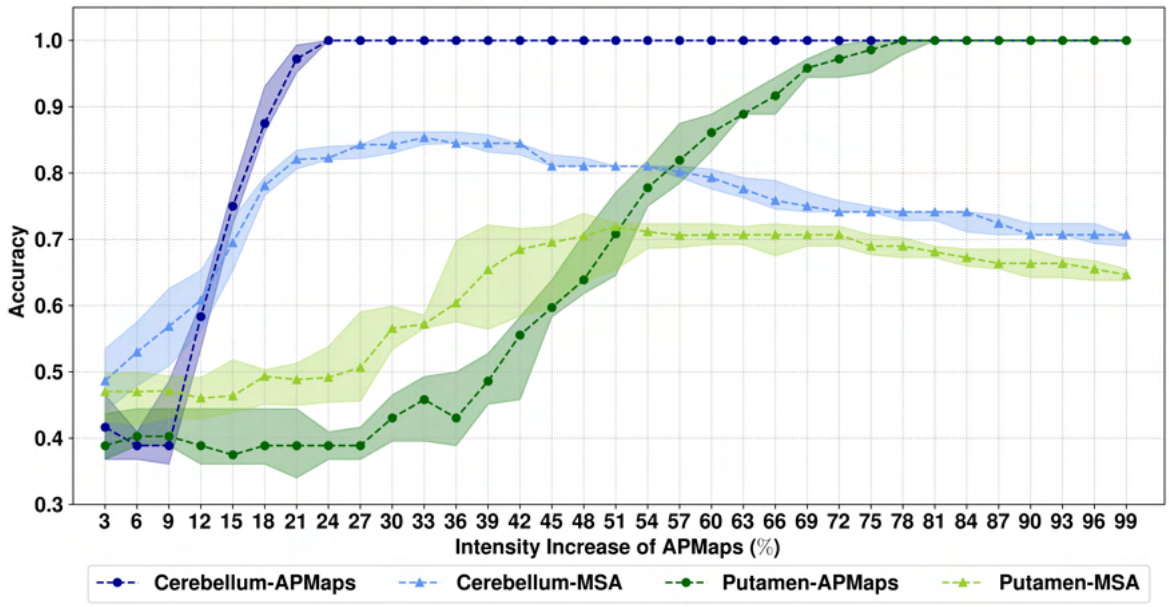


Figure 4.3: *Pathology-Agnostic APMaps for MSA Classification.* Median accuracy and IQR obtained with a 10-fold CV on the MSA/HC set (denoted as MSA) and the hold-out set of APMaps/OPMaps (denoted as APMaps), according to the intensity increase of the altered region in APMaps. APMaps: Altered Parametric Maps; CNN: Convolutional Neural Network; CV: Cross Validation; HC: Healthy Controls; IQR: Interquartile Range; MSA: Multiple System Atrophy; OPMaps: Original Parametric Maps. Reproduced from [223] ©2021 IEEE

Table 4.2 details the performances of the best-accuracy models. We can notice that almost all HC were well classified, whereas the sensitivity varied according to the region. Worth mentioning is that the Putamen CNN correctly classified almost every MSA-P, whereas the Cerebellum and Cerebellum/Putamen CNN identified both MSA-C and some MSA-P as TP.

Model	APMaps Intensity Increase	Accuracy	Sensitivity	Specificity
Cerebellum CNN	33%	0.85 (0.00)	0.72 (0.03)	1.00 (0.03)
Putamen CNN	72%	0.71 (0.00)	0.41 (0.01)	1.00 (0.00)
Cerebellum/Putamen CNN	33%/72%	*0.88 (0.02)*	*0.76 (0.03)*	*1.00 (0.00)*

Table 4.2: *Pathology-Agnostic APMaps for MSA Classification.* Performances given as median (IQR) of the best models trained with the APMaps/OPMaps set and tested on the MSA/HC set. We report the corresponding intensity increase of the APMaps used as training data. Best performances are highlighted in italic. APMaps: Altered Parametric Maps; CNN: Convolutional Neural Network; IQR: Interquartile Range; MSA: Multiple System Atrophy; OPMaps: Original Parametric Maps. Adapted from [223]

4.3.1.2.2 Visual Interpretation

As illustrated in Fig. 4.4, visualization maps highlighted the following regions:

- *Cerebellum CNN.* The MSA/HC set presented the target region in all visualizations with some voxels activated outside, especially in the case of MSA-P, compared to the APMaps/OPMaps sets;
- *Putamen CNN.* We can distinguish the target region in the APMaps/OPMaps visualizations, whereas the visualizations from the MSA/HC set are much noisier. However, the MSA-P set included the putamen, absent instead from the MSA-C;
- *Cerebellum/Putamen CNN.* The target regions presented high activations for the APMaps/OPMaps sets. Results regarding the MSA/HC set were comparable to those of the Cerebellum CNN.

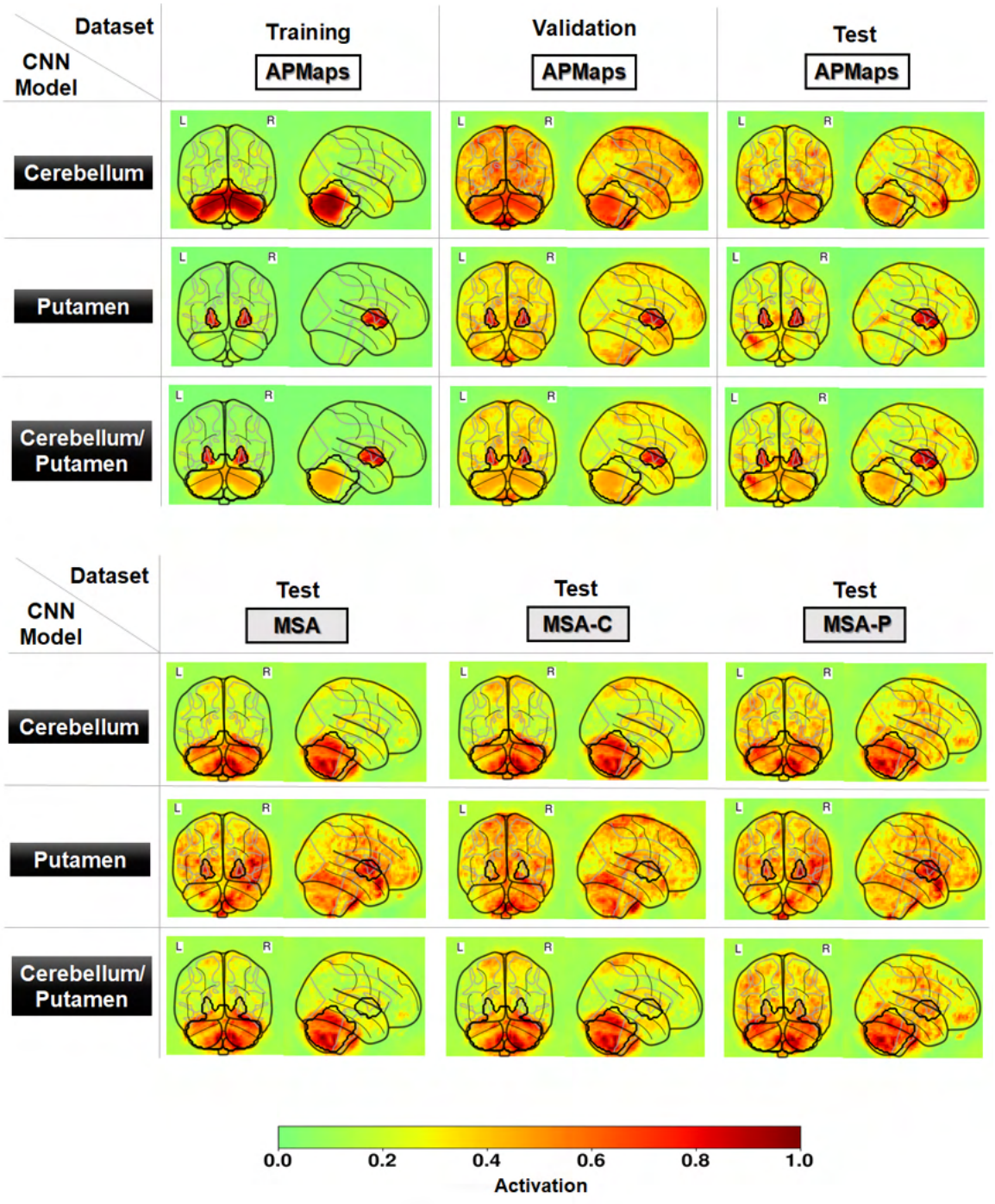


Figure 4.4: *Pathology-Agnostic APMaps for MSA Classification.* Each map shows the absolute difference between the mean maps of true positives and true negatives. The target regions (i.e. the regions altered in the APMaps) were activated in the training data and highlighted in the testing data despite some noise. Target regions are contoured in black. Each dataset is denoted by the positive class (either APMaps or MSA, MSA-C, MSA-P). APMaps: Altered Parametric Maps; HC: Healthy Controls; MSA: Multiple System Atrophy; MSA-C: MSA Cerebellar variant; MSA-P: MSA Parkinsonian variant; OPMaps: Original Parametric Maps; L: Left; R: Right. Adapted from [223]

4.3.1.3 Discussion

In this work, we used a 3D CNN trained with altered brain parametric maps to detect region-specific altered traits in pathological data. Although independent of each other, the two datasets (APMaps/OPMaps and MSA/HC) had in common the regions presenting alterations, i. e. the cerebellum and putamen.
The performances obtained with this approach were comparable to the state-of-the-art for MSA classification, even if we did not devise the alterations featuring the APMaps to resemble the pathology.

In a previous study based on volumetric MRI data and an SVM as the classifier, MSA-C and MSA-P were discerned from HC with 88.4% and 82.4% accuracies [206]. We achieved the best accuracy of 0.88 with the Cerebellum/Putamen CNN on the entire set of MSA patients. More recently, our research group advanced a fully automated data-driven pipeline for the distinction between HC, patients with MSA, and PD using an ML approach and automatic feature selection. MD was the most selected index among other structural and functional ones achieving 94% to discern HC from MSA patients. However, the approach we proposed differs from the fully automated data-driven pipeline in the following respects:

- *Classifier*: 3D CNN vs. traditional ML methods (feature extraction/selection and Sequential Minimal Optimization (SMO));
- *Input data*: Monomodal vs. multimodal 3D MRI images;
- *Aim*: Detection of region-specific abnormal traits vs. disease discrimination.

Moreover, our best accuracy is comparable to the one achieved in another study from our group (0.880 vs. 0.895) by a similar CNN architecture using only MD maps as input and the same set of MSA/HC we employed [106]. These findings put forward the discriminating power of MD maps for discerning between MSA patients and HC, hence proving the effectiveness even of a monomodal MRI approach.
Given our findings, we obtained competitive performances compared to these reference studies despite the pathology-agnostic character of the training data. We may thus infer that the patterns learned by the CNNs from the APMaps enclosed sufficient and meaningful information to detect similar alterations in the set of MSA/HC, as supported by the visualization maps.

First, we can notice that the regional intensity increase in the APMaps influenced performances on the MSA/HC set. Considering an intensity increase higher than 42% for the

cerebellum and 75% for the putamen, discrimination performances on the MSA/HC set deteriorated.
Regarding the best-accuracy models, the Cerebellum CNN obtained an accuracy of 0.85, whereas the Putamen CNN reached only 0.71. We can explain this difference considering that putaminal MD increase is not considered a sensitive biomarker for MSA diagnosis [168]. Given the smaller size of the putamen coupled with the heterogeneity of pathological data, we can imagine that the patterns encountered in the MSA are much less homogeneous than those found in the APMaps. Furthermore, we can observe that the highest variability of our approach resides in model sensitivity, whereas specificity was constantly equal to 1.00. It suggests that, according to the input data, the patterns learned by the CNNs could be more or less relevant to detecting similar traits in pathological data.

Visualizing the most discriminant parts for CNN prediction helped verify the reliability of pattern retrieval. Looking separately at MSA-C and MSA-P showed that the regions of interest were highlighted, despite some noise. We must bear in mind that the distinction between the MSA variants in our study relied on the symptoms from a clinical point of view, thus not excluding changes in MD values outside the most affected regions.
Another factor is the more homogeneous patterns created in the APMaps compared to the variability of pathological data, with each patient potentially presenting unique characteristics. One possibility could be including more than one intensity increase for the APMaps in the input data to test whether this could ameliorate performances. We can also notice from the visualization maps that activations relative to training data were much more uniform over the regions compared to those found for the MSA/HC dataset. It may suggest that the alterations characterizing MSA patients are not as homogeneous as those created in the APMaps, as expected from pathological changes. The most surprising aspect is that, regardless of the simple and coarse modifications of the APMaps, they mimicked realistic alterations concerning specific anatomical regions, which revealed useful to detect similar characteristics in pathological data.
Interestingly, regardless of the low sensitivity shown by the Putamen CNN, the target region was activated in the visualization maps so backing the prediction of well-classified patients.

Despite the limited sample size of both datasets and the differences concerning the alterations, our findings are encouraging. Especially for rare diseases such as MSA burdened by a paucity of data, our approach represents a valid alternative to detect regional alterations. This method paves the way for applications to other pathologies with common regions of interest, exploiting a priori knowledge of the most distinctive traits of a disease.

4.3.2 APMaps from Cluster-Based MSA Features

In the previous section, we used pathology-agnostic altered brain MRI parametric maps to detect similar traits in data from MSA patients. These findings have opened the way for deeper considerations about the importance of features, like the intensity and size of an altered region, especially when transferred to pathological data.

One of the foremost concerns when dealing with pathological data is the intra- and inter-individual variability complicating pattern retrieval. Even in a more homogeneous and controlled case, such as with the APMaps, we realized how evident and challenging these aspects were. For instance, consider the performance of biregion-trained CNNs on monoregion APMaps, leading to different results according to the targeted regions (see Section 3.3.3).
To address this issue, we proposed to consider specific pathological traits, in our case MSA-inspired features, to refine the creation of APMaps and obtain region-specific pathology-oriented APMaps. Given the undeniable influence of intensity modification on model performance (as shown by monoregion-trained CNNs, Section 3.3.1), we decided to group MSA patients according to the severity of alterations by considering the distribution of MD values. Indeed, we can associate high MD values with tissue microstructural anomalies, as in MSA patients [14,168]. We obtained different clusters of MSA patients, each representing a degree of alteration (from mild to severe, with increasing MD values). We exploited these clusters to create the *CB-APMaps*, used to train a 3D CNN along with the OPMaps. To evaluate the CNN generalization ability, we tested the network on the set of MSA/HC, as previously done.

Is it feasible to exploit the information about varying degrees of MD increase directly extracted from the pathology to improve the classification of pathological data? That is the main research question we will elucidate in the following sections.
This work was presented at the annual meeting of the International Society for Magnetic Resonance in Medicine [225].

4.3.2.1 Material and Methods

4.3.2.1.1 Datasets

We applied the same preprocessing steps reported in Section 3.2.2 to the following datasets:

- *MSA/HC*, comprising the MD maps from 29 MSA patients and 26 healthy controls. Further details are available in Section 4.3.1.1.1 and previous studies [14,41];

- *CB-APMaps/OPMaps*, including 89 CB-APMaps and 89 OPMaps. We created the CB-APMaps by considering the MD maps from 89 healthy participants (i.e. the OPMaps described in Section 3.2.1).

4.3.2.1.2 Creation of CB-APMaps

To create CB-APMaps, we focused on modifying MD values belonging to the cerebellum, one of the regions of interest in MSA pathophysiology [14, 41]. To do so, we used k-means clustering to group patients according to the distribution of MD values in the cerebellum and obtained the CB-APMaps by applying the histogram-matching technique to the OPMaps.

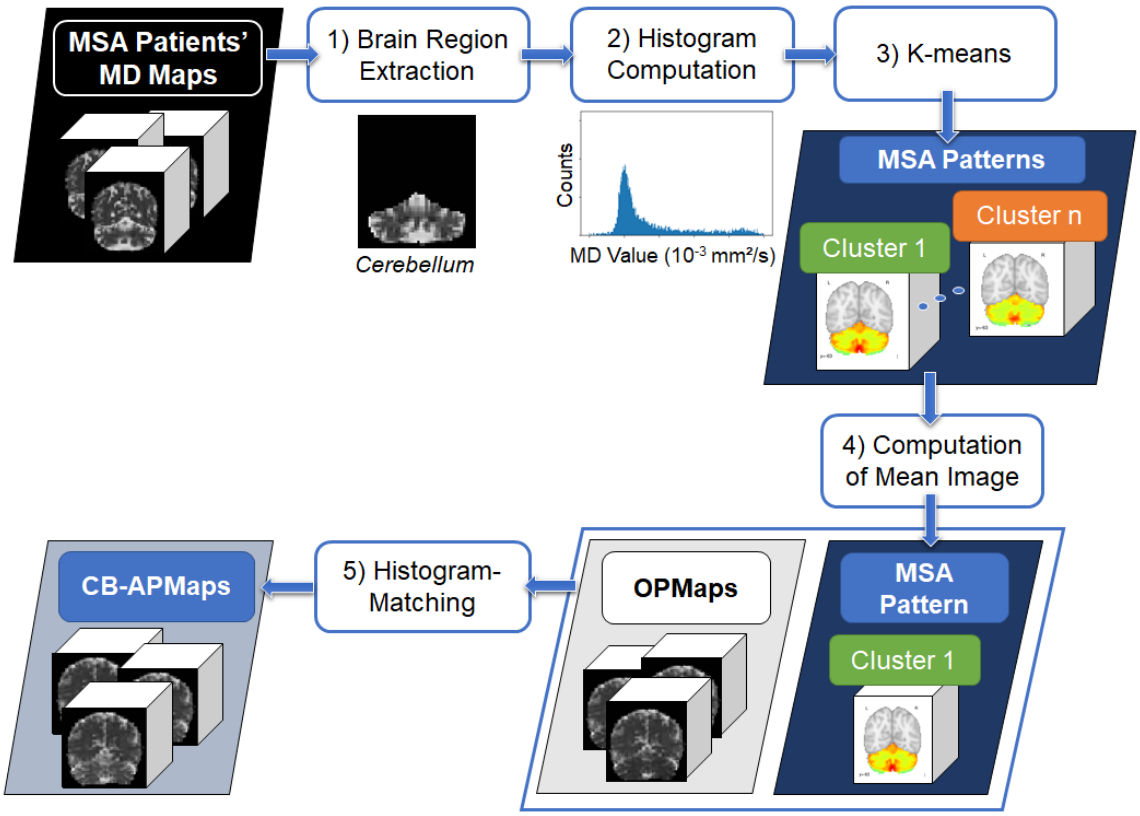


Figure 4.5: *Creation of CB-APMaps.* We extracted the cerebellum from each MD map of the 29 MSA patients and computed the histogram of MD values exclusively in this region. We applied k-means on these histograms to cluster patients according to the distribution of MD values. For each cluster, we computed the mean image used as a reference for the histogram-matching technique to obtain the CB-APMaps from the OPMaps. CB-APMaps: Cluster-Based Altered Parametric Maps; MD: Mean Diffusivity; MSA: Multiple System Atrophy; OPMaps: Original Parametric Maps. Adapted from [225]

We can summarize our method, schematized in Fig. 4.5, as follows:

1. Extraction of the cerebellum using an atlas-based mask from the MD maps belonging to the MSA patients;
2. Histogram computation of the MD values belonging only to the region;
3. K-means clustering performed on the histograms (i.e. the input to the k-means was the histogram representing the frequency associated to each of the 256 bins);
4. Computation of the mean image of each cluster, representing the pattern to be reproduced;
5. Application of the histogram-matching technique to transform the OPMaps into CB-APMaps.

To choose the optimal number of clusters k, we relied on the silhouette coefficient, measuring the similarity of a sample to its cluster compared to the other clusters [45] (see Section 1.2.3.2.1).
We created the CB-APMaps by applying the histogram-matching technique [226]. The latter allows for transforming an image such that its histogram matches the histogram of another image. In our case, we considered as a reference the mean image computed from the data belonging to each cluster, considering only the histogram of the cerebellum. After extracting the cerebellum from each OPMap, we applied the histogram-matching technique to mimic the reference image. We modified only the cerebellum, leaving the rest of each image unaltered. Therefore, we obtained 89 CB-APMaps for each cluster and each k, by applying the histogram-matching technique to the 89 OPMaps and using the mean image from each cluster as reference image.

4.3.2.1.3 CNN Implementation

We implemented the 3D CNN described in Section 3.2.4. We trained this network with CB-APMaps/OPMaps for each cluster, adopting a 10-fold CV and leaving a hold-out set for testing.
We used the MSA/HC set to test CNN performance and determine whether training the network with the CB-APMaps/OPMaps could guarantee good discrimination between HC and patients with MSA.
Accuracy, sensitivity, and specificity were the chosen metrics for performance evaluation.

4.3.2.2 Results

4.3.2.2.1 K-Means Clustering

Regarding the number of clusters, we considered $k = 2$, determined by the silhouette method, and $k = 3$ for comparison.

We defined clusters according to the increasing quantity of higher MD values in the cerebellum (Mild, Intermediate, and Severe). Fig. 4.6 provides the mean reference image for each cluster and the mean histogram, whereas Fig. 4.7 shows some examples of CB-APMaps.

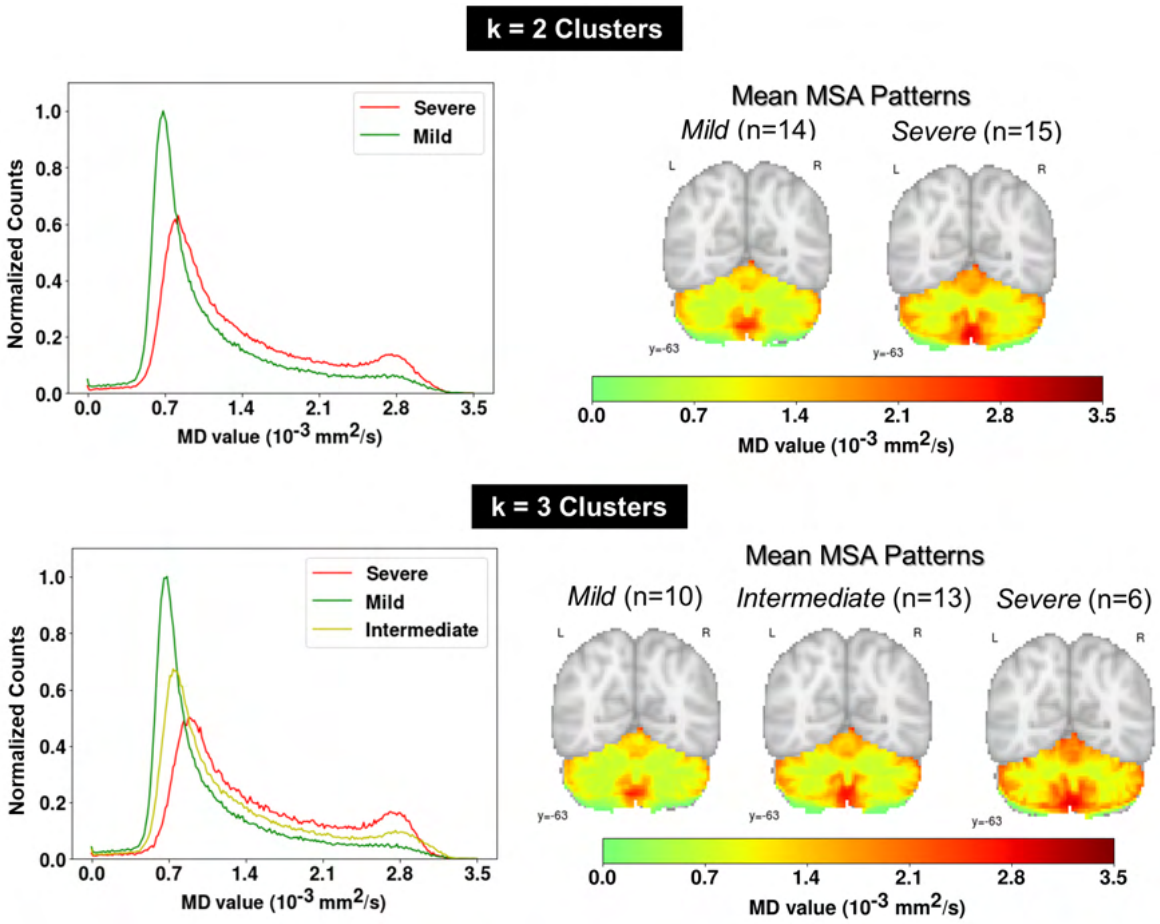


Figure 4.6: *CB-APMaps for MSA Classification.* Mean histogram and reference image for each cluster according to the total number of clusters k. CB-APMaps: Cluster-Based Altered Parametric Maps; HC: Healthy Controls; MSA: Multiple System Atrophy; OPMaps: Original Parametric Maps. Adapted from [225]

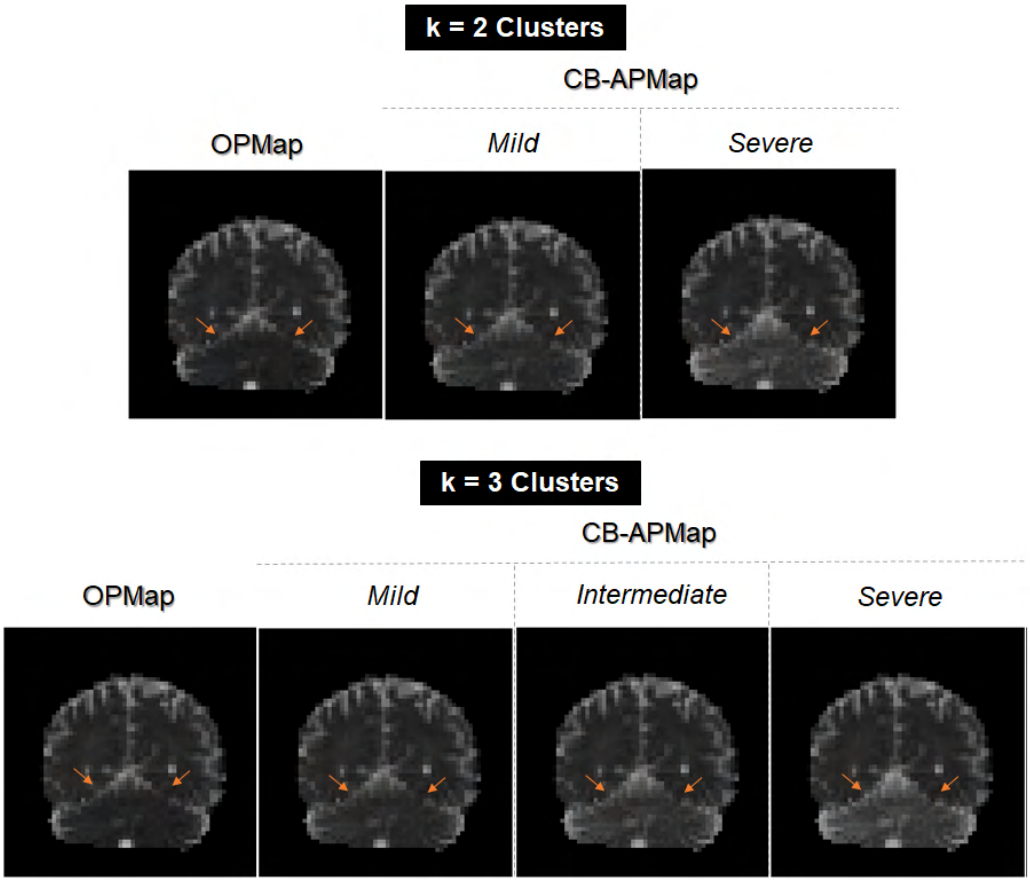


Figure 4.7: *CB-APMaps for MSA Classification.* Example of CB-APMaps obtained by applying the histogram-matching technique to the OPMap for each MSA cluster according to the total number of clusters k. Arrows point to the modified region, i.e. the cerebellum. CB-APMaps: Cluster-Based Altered Parametric Maps; MSA: Multiple System Atrophy; OPMap: Original Parametric Map. Adapted from [225]

4.3.2.2.2 CNN Performance

Accuracy on the hold-out set of CB-APMaps/OPMaps was over 0.90 for the Intermediate and Severe clusters, whereas it was inferior to 0.60 for Mild clusters.

Table 4.3 presents the performances obtained on the MSA/HC set for both k. The highest accuracy was obtained with cluster Severe from $k = 2$, reaching sensitivity equal to 0.69 and maximum specificity. However, misclassified patients belonged to Mild clusters.

Mild clusters presented the worst performances (accuracy around 0.70), whereas the Intermediate cluster performed similarly to the Severe clusters.

k = 2 Clusters

CB-APMaps Training Cluster	Accuracy	Sensitivity	Specificity
Severe	*0.84 (0.00)*	0.69 (0.00)	*1.00 (0.00)*
Mild	0.71 (0.04)	*0.83 (0.02)*	0.58 (0.08)

k = 3 Clusters

CB-APMaps Training Cluster	Accuracy	Sensitivity	Specificity
Severe	0.81 (0.01)	0.62 (0.01)	*1.00 (0.00)*
Intermediate	*0.82 (0.00)*	0.72 (0.00)	0.92 (0.00)
Mild	0.66 (0.02)	*0.92 (0.02)*	0.40 (0.05)

Table 4.3: *CB-APMaps for MSA Classification.* Performances on the MSA/HC set given as median (IQR) according to the cluster of APMaps used to train the CNN. Best performances are highlighted in italic. CB-APMaps: Cluster-Based Altered Parametric Maps; CNN: Convolutional Neural Network; HC: Healthy Controls; IQR: Interquartile Range; MSA: Multiple System Atrophy; OPMaps: Original Parametric Maps. Adapted from [225]

4.3.2.3 Discussion

In this study, we discerned MSA patients from healthy controls using a 3D CNN trained with CB-APMaps, brain MRI parametric maps modified to resemble region-specific MSA traits.

According to our findings, CB-APMaps represent a valuable source of knowledge for the proposed CNN to distinguish MSA patients from healthy controls. Our best performance reached an accuracy equal to 0.84, although, in a previous study focusing on the discrimination between MSA patients and HC, the best accuracy was equal to 0.94 using MD maps [41]. An essential difference is that we created the altered data from the same patients used for testing, even if the reference image was the mean pattern from each cluster, potentially a new example. It is possible that, besides preserving some original information that led to satisfactory performances, we may have also introduced some noise, compared to the case in which the same data are used for training and testing the network. This aspect is pivotal when thinking about designing novel data augmentation techniques.

A closer look at misclassified patients from the best-accuracy model showed that they all belonged to the Mild clusters. The latter presented minor MD increases in the cerebellum, so these patients were more similar to HC. Indeed, CNN performances on the hold-out set

of CB-APMaps/OPMaps were poor, proving that Mild CB-APMaps were not so different from OPMaps, thus more difficult to distinguish. However, we must keep in mind that MSA patients may also present alterations in regions other than the cerebellum [14, 41, 168]. In our approach, that might be a point worth improving, given that patients presenting mild cerebellar modifications were classified as healthy controls.

One unanticipated finding was that the best accuracy yielded by the pathology-agnostic APMaps (Section 4.3.1) was slightly higher than the one obtained with the CB-APMaps (0.88 vs. 0.84). It is cumbersome to determine why one type of APMaps was more effective than the other. Perhaps, a statistical comparison between the images could help clarify this point. In these approaches, we used the CNN as a validating tool to examine the discriminating power of the APMaps, with all the pros (e.g. good performances) and cons (e.g. level of uncertainty) that may derive. Nevertheless, these findings are interesting as they suggest that the proposed CNN is as sensitive to general regional modifications as it could be to more specific regional patterns.
Despite the limited samples, we obtained promising results, paving the way for further applications to a different MSA pattern or another pathology. Indeed, one could tailor our method by modifying the reference image, for instance, by integrating the standard deviation of the images from each cluster to increase the variability and constitute a training set with diverse types of CB-APMaps.
Another possibility is to compare other CNN architectures to assess the validity of these altered data, regardless of the CNN model. That is an interesting point for future research.

4.3.3 APMaps from Z-Score-Based MSA Features

Similarly to pathology-agnostic APMaps, the main limitation of CB-APMaps (see Section 4.3.2) was that they enclosed only region-specific pathological traits. The former presented coarser alterations independent from any pathology, whereas the latter reproduced cerebellar anomalies from the MSA patients at our disposal. However, pathological alterations may include more diffuse alterations all over the brain. To improve this aspect, we proposed a method for creating pathology-oriented APMaps incorporating whole-brain changes due to the disease under consideration, i.e. MSA. We produced *ZB-APMaps* considering the z-score computed from the MD maps of patients with MSA and healthy individuals.
To allow for comparison with the other variants of APMaps, we used the ZB-APMaps to train a 3D CNN and investigated performances on the set of MSA/HC, similarly to Sections 4.3.1.1.2 and 4.3.2.1.3. The ultimate goal was to determine whether we could benefit from MSA-inspired whole-brain altered MRI data to better discriminate between healthy controls

and MSA patients.

We divided this experimental part into two phases, briefly presented in Fig. 4.8:

1. *One-Pattern Approach.* The first setting was intended to keep the level of complexity as low as possible by evaluating CNN performances when fed with ZB-APMaps reproducing a single pattern. In this way, we could determine the degree of variability among patients affecting CNN performance and thus get an overall idea of the possible outcomes.

2. *Multi-Pattern Approach.* A pathological cohort is characterized by data heterogeneity because it comprises diverse pathological patterns. The multi-pattern approach aimed to reproduce this condition by first establishing a *reference*, i.e. training the network with data from 20 randomly selected MSA patients and HI. Once established this baseline performance, the next step was to train the network with the ZB-APMaps and progressively increase the representation of each pattern in training by feeding an increasing number of ZB-APMaps for each pattern (from 20 to 200 ZB-APMaps, i.e. each pattern represented from one to ten times). Our objective was to determine whether this *amplification* could improve CNN performance due to the augmented training content. To make this approach feasible, we included additional data comprising another set of MSA patients and Healthy Individuals (HI) from different datasets.

Henceforth, we will use the words *patterns* or *patients* for reference to data belonging to MSA patients.

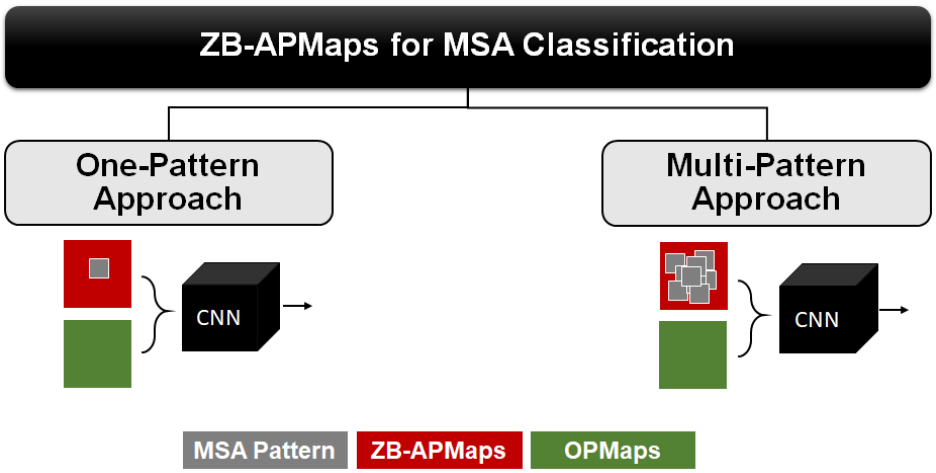


Figure 4.8: *ZB-APMaps for MSA Classification.* The main difference between the one-pattern and multi-pattern approach is that the former evaluates the discriminating power of a single MSA pattern, by feeding as input ZB-APMaps from a single pattern in training, whereas the latter considers ZB-APMaps from multiple patterns for training. CNN: Convolutional Neural Network; MSA: Multiple System Atrophy; OPMaps: Original Parametric Maps; ZB-APMaps: Z-score-Based Altered Parametric Maps

4.3.3.1 One-Pattern Approach

The objective of the one-pattern approach was to analyze the discriminating power of ZB-APMaps from a single pattern with respect to the entire set of MSA patients. Using a single pattern for training the CNN enabled us to keep a fair level of interpretation as we could inspect each patient visually.
Fig. 4.9 illustrates a schematic diagram of the one-pattern approach.

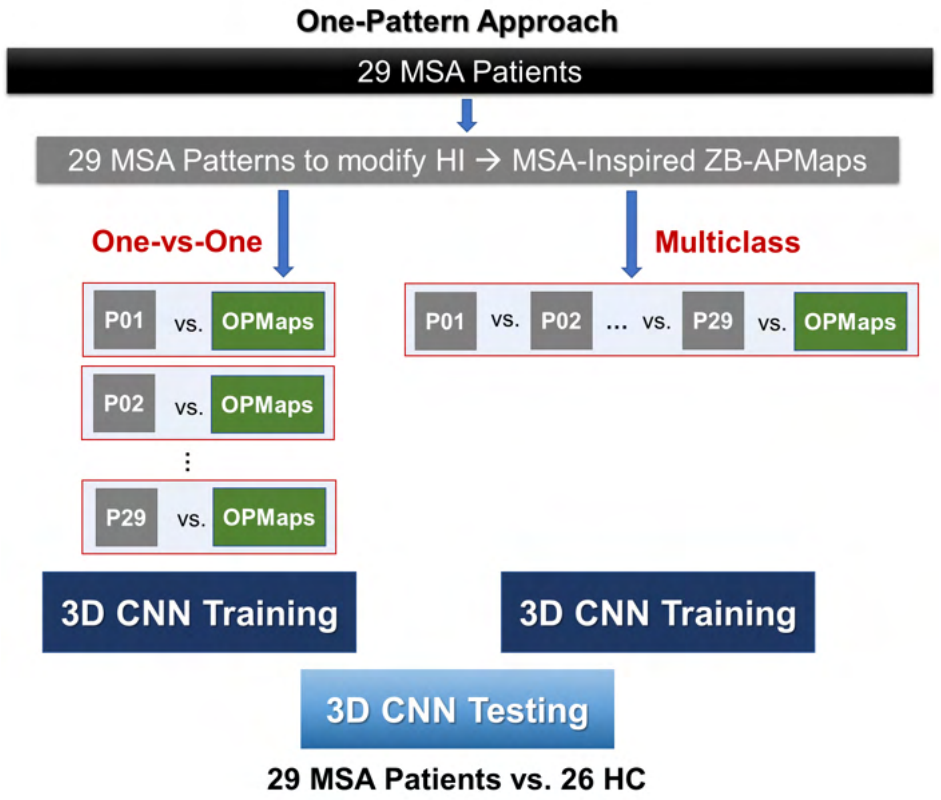


Figure 4.9: *ZB-APMaps for MSA Classification - One-Pattern Approach*. We compared CNN performances, evaluated on the set composed of the MD maps from the 29 MSA patients and 26 HC, by considering: 1) *One-vs-One*, 29 networks each trained to distinguish the ZB-APMaps created with one pattern (P01, P02, ..., P29) from the OPMaps; 2) *Multiclass*, one network trained to discern the OPMaps and the 29 classes of MSA patterns. CNN: Convolutional Neural Network; HC: Healthy Controls; MD: Mean Diffusivity; MSA: Multiple System Atrophy; OPMaps: Original Parametric Maps; ZB-APMaps: Z-score-Based Altered Parametric Maps

We investigated the informative content of each pattern, identified by the z-score obtained from each MSA patient, adopting two types of classification:

- *One-vs-One*: we trained the network to distinguish the ZB-APMaps from each pattern against the OPMaps, thus obtaining 29 networks, each trained with ZB-APMaps coming from a different pattern.

- *Multiclass*: ZB-APMaps from each pattern constituted a separate class, obtaining a total of 30 classes: 29 corresponding to the 29 MSA patterns and one for the OPMaps.

This allowed us to establish the presence of patterns capable of generalization as we assessed CNN performance by testing on the same set of MSA patients and HC.
Let us guide you through the proposed approach.

4.3.3.1.1 Material and Methods

4.3.3.1.1.1 Datasets

We employed the following datasets, processed as described in Section 3.2.2:

- *MSA/HC*, including MD maps from 29 MSA patients and 26 healthy controls. For further details, please refer to Section 4.3.1.1.1 and previous studies [14, 41];
- *ZB-APMaps/OPMaps*, including 89 ZB-APMaps and 89 OPMaps. We created the ZB-APMaps from the MD maps of 89 healthy participants (i.e. the OPMaps described in Section 3.2.1).

4.3.3.1.1.2 Creation of ZB-APMaps

We can summarize the creation of ZB-APMaps in two main steps:

1. We computed the z-score Z_{P_n} relative to the MSA pattern P_n considering the MD map from the corresponding MSA patient I_{P_n} and the mean μ and standard deviation σ, calculated voxelwise, considering the MD maps of the 89 healthy individuals.

$$Z_{P_n} = \frac{I_{P_n} - \mu}{\sigma} \tag{4.1}$$

   Since we were interested in the increase of MD values due to the pathology, we considered only positive values of Z_{P_n}.

2. We obtained the ZB-APMap from the healthy individual x mimicking the MSA pattern P_n by adding the standard deviation of the healthy individuals σ, multiplied by the z score, to the original MD map of each healthy individual $OPMap_x$, as defined in (4.2).

$$\text{ZB-APMap}_{x,P_n} = \text{OPMap}_x + Z_{P_n}\sigma \tag{4.2}$$

We obtained a total of 29 patterns corresponding to the number of MSA patients, indicated by P following the patient's number (e.g. P01 is the pattern from patient #1).

4.3.3.1.1.3 CNN Implementation

We implemented the 3D CNN described in Section 3.2.4. In this case, we performed a 10-fold CV on the entire set of ZB-APMaps/OPMaps and used the MSA/HC set for testing. We evaluated two types of classification:

- *One-vs-One*: binary classification (ZB-APMaps vs. OPMaps), considering each pattern singly. We trained the network with each set of ZB-APMaps/OPMaps, obtaining 29 networks trained with the patterns from the 29 MSA patients.
- *Multiclass*: discrimination among 30 classes, i.e. 29 classes from the MSA patterns (one per pattern) and one class for the OPMaps. To obtain the final prediction on the MSA/HC set, we associated the positive class with the 29 classes from the patterns.

We assessed performances with accuracy, sensitivity, and specificity by computing the median value and IQR over the ten folds.

4.3.3.1.2 Results

Fig. 4.10 provides the performances of the One-vs-One classification according to the pattern used in training. Accuracy varied from a minimum of 0.55 for training pattern P18 to a maximum of 0.89 for P09. It oscillated between 0.75 and 0.80 for most patterns.
Moreover, specificity remained high (> 0.90), whereas the sensitivity oscillated around 0.70.

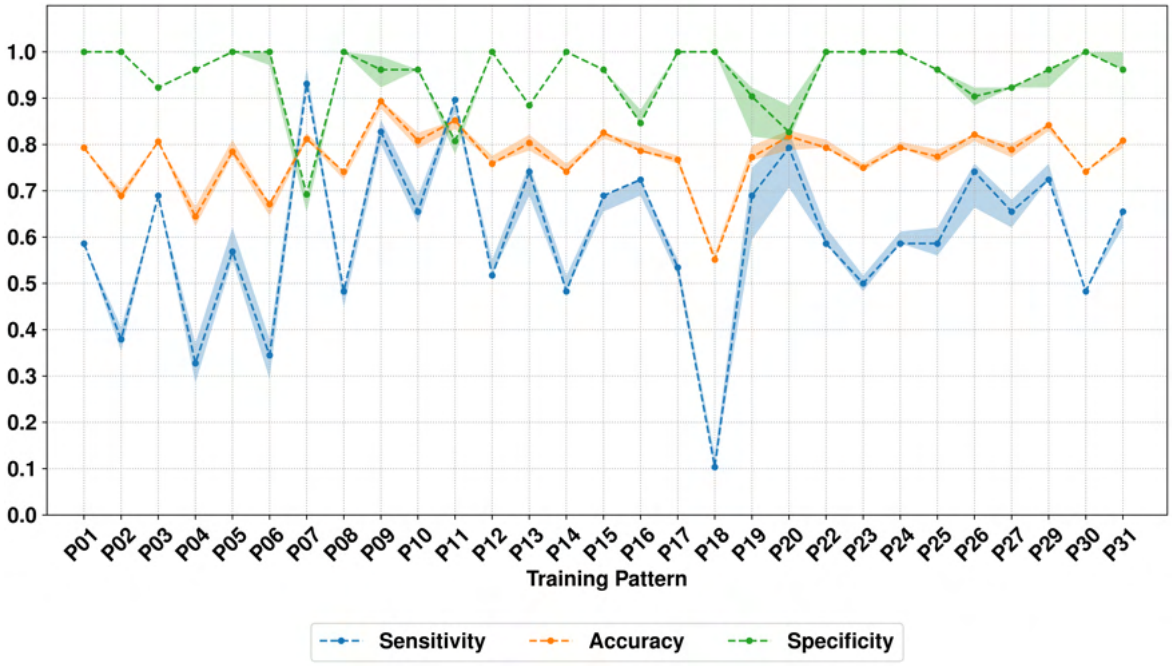


Figure 4.10: *ZB-APMaps for MSA Classification - One-vs-One Classification.* CNN performance provided as median and IQR on the MSA/HC set according to the pattern of ZB-APMaps used for training. CNN: Convolutional Neural Network; HC: Healthy Controls; IQR: Interquartile Range; MSA: Multiple System Atrophy; ZB-APMaps: Z-score-Based Altered Parametric Maps

Table 4.4 details the performances according to the type of classification, given as the median (IQR) for each metric. Multiclass classification achieved the best scores for all metrics. One-vs-One classification showed high specificity (0.96) and low sensitivity (0.62). We can also note that multiclass classification obtained the highest sensitivity.

Classification	Accuracy	Sensitivity	Specificity
One-vs-One	0.79 (0.07)	0.62 (0.21)	0.96 (0.08)
Multiclass	0.89 (0.02)	1.00 (0.00)	0.79 (0.04)

Table 4.4: *ZB-APMaps for MSA Classification.* CNN performance provided as median (IQR) on the MSA/HC set according to the type of classification. Best performances are highlighted in italic. CNN: Convolutional Neural Network; HC: Healthy Controls; IQR: Interquartile Range; MSA: Multiple System Atrophy; ZB-APMaps: Z-score-Based Altered Parametric Maps

4.3.3.1.3 Discussion

In this Section, we explored the discrimination ability of a 3D CNN trained with the ZB-APMaps, whole-brain MSA-inspired APMaps to distinguish between patients with MSA

and healthy controls. Overall, the proposed approach led to the creation of altered brain MRI parametric maps containing relevant information about MSA patterns, valuable enough for the CNN to reach satisfying performances on the MSA/HC set. The best accuracy of 0.89 was competitive with respect to previous studies (e.g. accuracy = 0.94 [41]) and outperformed the approach based on the CB-APMaps (best accuracy = 0.84). We registered a performance improvement, moving from a region-specific to a whole-brain approach for creating APMaps.

One-vs-One classification showed that some patterns were more versatile than others in that they seemed to provide enough information for the network to discriminate between MSA patients and HC. That is not as surprising as we acknowledged that some similarities exist among these patients (see clustering results in Section 4.3.2.2).
Multiclass performances were higher for sensitivity and accuracy but poorer for sensitivity compared to One-vs-One. One possible explanation is that, in multiclass classification, the CNN had already learned during training all MSA patterns through the ZB-APMaps, thus performing exceptionally on the MSA patients (maximum sensitivity).

No matter how encouraging these results are, their main limitation resides in the fact that the MSA patients used for creating the ZB-APMaps were the same used for testing the network. Indeed, the different performances obtained by the One-vs-One classification proved that each pattern enclosed information more or less beneficial to detect similarities in unseen data. The restricted number of healthy individuals used to create the ZB-APMaps could also be a point of improvement to increase the variability of normal subjects, and thus of altered data, in training.

4.3.3.2 Multi-Pattern Approach

Building on the promising results of the one-pattern approach (in Section 4.3.3.1), we moved to a more complex condition with the multi-pattern approach, getting closer to the intrinsic heterogeneity of pathological data. To this end, we trained the network with more than one pattern, hence the name *multi-pattern approach*.
One crucial difference compared to the one-pattern approach is the increase in sample size for MSA patients (29 vs. 58) and HI (89 vs. 470). This step was essential to enrich the inter-individual variability and the representation of each pattern.

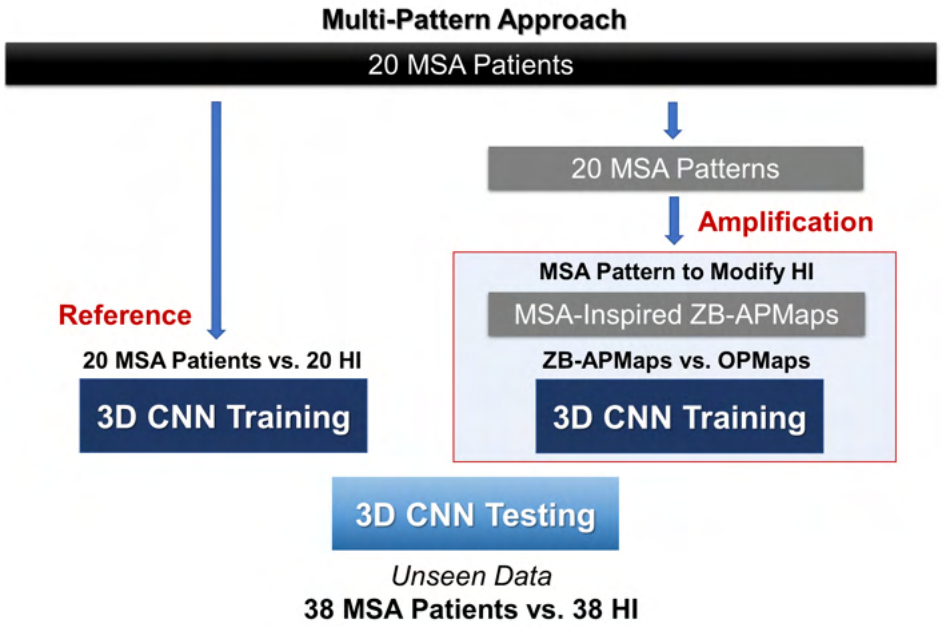


Figure 4.11: *ZB-APMaps for MSA Classification - Multi-Pattern Approach.* We compared CNN performances, evaluated on a separate testing set composed of the MD maps from 38 MSA patients and 38 HI, by considering the CNN trained with: 1) 20 MSA patients and 20 HI (randomly chosen); 2) ZB-APMaps and OPMaps in a variable number depending on the degree of amplification for each MSA pattern. CNN: Convolutional Neural Network; HI: Healthy Individuals; MD: Mean Diffusivity; MSA: Multiple System Atrophy; OPMaps: Original Parametric Maps; ZB-APMaps: Z-score-Based Altered Parametric Maps

The aim of the multi-pattern approach was two-folded, as summarized in Fig. 4.11:

- Determining whether the CNN could achieve at least similar performances with a comparable sample size between the two sets (ZB-APMaps/OPMaps and MSA/HI), thus establishing a *reference*;
- Improving performances by increasing the representation of each MSA pattern or, in other words, *amplifying* each pattern *n* times to discover whether the higher number of ZB-APMaps in training could ameliorate performances on the testing set.

4.3.3.2.1 Material and Methods

4.3.3.2.1.1 Datasets

The image processing described in Section 3.2.2 was applied to all datasets. We changed the image resolution from 3×3×3 mm^3 to 2×2×2 mm^3, allowing for better image quality and the possibility of keeping smaller cerebral structures.
Concerning data from healthy individuals, we gathered a total of 470 subjects as follows:

- 69 out of the 89 healthy subjects described in Section 3.2.1, excluding patients younger than 40 years.
- 26 healthy controls, age-matched to the 29 MSA patients (reported in Section 4.3.1.1.1).
- 57 healthy subjects belonging to an in-house database;
- Data used in the preparation of this study were obtained from:
  - The Parkinson's Progression Markers Initiative (PPMI) database (`www.ppmi-info.org/access-data-specimens/download-data`). For up-to-date information on the study, visit `www.ppmi-info.org`. PPMI - a public-private partnership - is funded by the Michael J. Fox Foundation for Parkinson's Research and funding partners (listing available at `www.ppmi-info.org/about-ppmi/who-we-are/study-sponsors`).
  - The Alzheimer's Disease Neuroimaging Initiative (ADNI) database (`adni.loni.usc.edu`). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in the analysis or writing of this work. A complete listing of ADNI investigators can be found at: `http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf`. The ADNI started 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 combining serial MRI, positron emission tomography (PET) and other biological markers with clinical and neuropsychological assessment can be exploited to measure the progression of mild cognitive impairment (MCI) and early Alzheimer's disease (AD). For up-to-date information, please visit `www.adni-info.org`.

  We accessed these databases to gather DTI data acquired from healthy subjects with the following parameters:

  - Age $\geq$ 50 years;
  - According to data availability, subjects classified as normal controls either at baseline (for PPMI) or screening (for ADNI);
  - Regarding image acquisition, we considered magnetic field at 3 T, all available manufacturers for the MRI machines, and all sequences for DTI acquisition.
  - In the presence of multiple acquisitions, we selected the most recent or the one with better image quality.

We selected a total of 258 subjects for ADNI and 60 for PPMI.

In addition to the 29 MSA patients mentioned in Section 4.3.1.1.1, we considered another set of MSA patients (n=29) (for more information, please refer to a recent study [227]). Table 4.5 details the age range, mean, and SD regarding healthy individuals and patients.

		MSA Patients	Healthy Individuals
	Count	58	470
Age (years)	**Mean**	62.2	67.8
	SD	7.9	7.7
	Min	42.0	41.4
	Max	78.0	89.0

Table 4.5: Summary statistics about the age distribution of healthy individuals and patients. SD: Standard Deviation

4.3.3.2.1.2 Variants of ZB-APMaps

We computed the z-score with the method described in Section 4.3.3.1.1.2, considering the mean μ and standard deviation σ of all MD maps from the healthy individuals.
To explore the effect of different types of ZB-APMaps, we applied a threshold on the z-score-based APmaps as follows:

- We set values higher than 1.5 multiplied by the image to be modified I_x equal to the corresponding value I_x. We called this threshold *T1.5*;

$$\text{ZB-APMap}_{x,P_n} > 1.5I_x \rightarrow \text{ZB-APMap}_{x,P_n} = I_x \tag{4.3}$$

- We set values higher than the MSA patient's image I_{P_n} equal to the corresponding value I_{P_n}. We called this threshold *TPat*.

$$\text{ZB-APMap}_{x,P_n} > I_{P_n} \rightarrow \text{ZB-APMap}_{x,P_n} = I_{P_n} \tag{4.4}$$

Both thresholds allowed for limiting aberrant values, the former focusing on the healthy individual and the latter on the MSA patient.

4.3.3.2.1.3 CNN Implementation

Regarding CNN implementation, Fig. 4.11 illustrates the salient parts of our method. From the total set of MSA patients, we randomly selected 20 patients used exclusively to train the network, whereas the remaining 38 served for testing. We randomly selected the

same number of healthy individuals for the testing set (38 MSA patients vs. 38 HI).
We can identify two main branches in the multi-pattern approach:

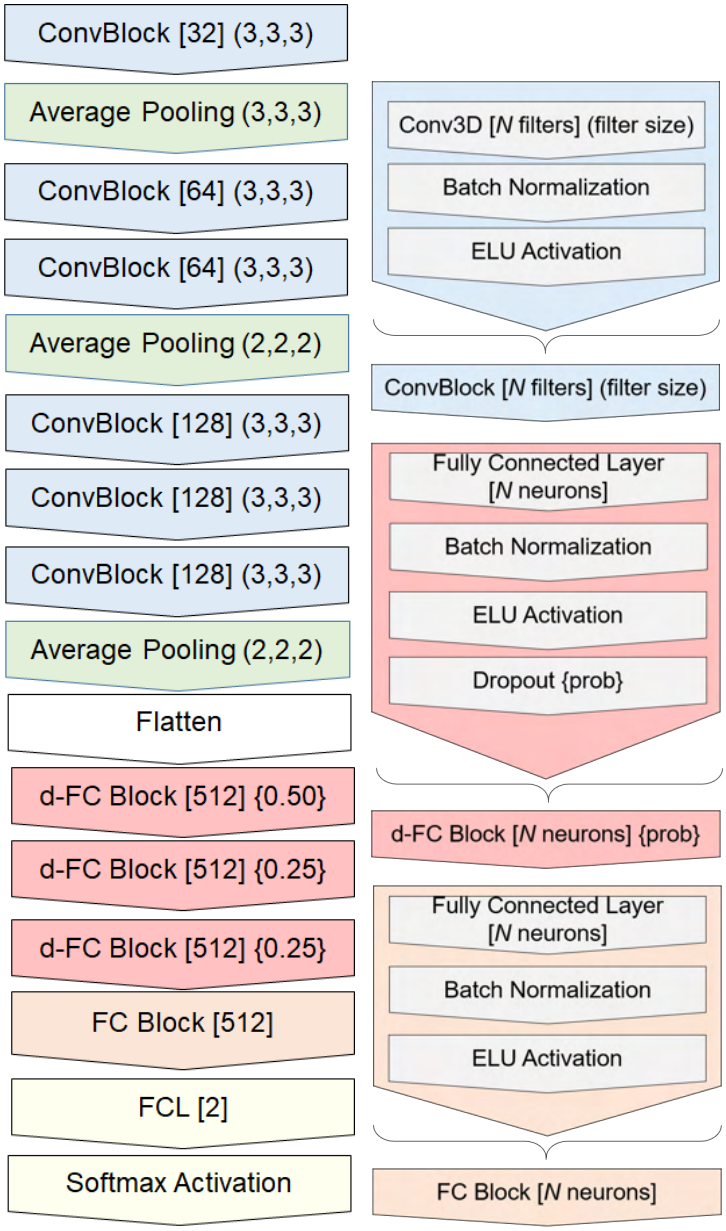
1. *Reference.* We trained the CNN with 21 sets, each composed of the MD maps from 20 MSA patients and 21 different sets of 20 HI, randomly selected from the healthy individuals with no overlap. That established a reference for the CNN performance when training directly on the MSA patients and HI.

2. *Amplification.* Network training began using 20 ZB-APMaps and 20 OPMaps, meaning only one ZB-APMap per pattern (i.e. no amplification with respect to the original number of 20 MSA patients). Pattern amplification consisted in increasing the representation of each pattern from two to ten times. That led to training the network from 40 ZB-APMaps vs. 40 OPMaps to 200 ZB-APMaps vs. 200 OPMaps. The selection of HI from the available set was kept random, considering each healthy individual only once (either as a ZB-APMap or an OPMap). We repeated the random sampling for ZB-APMaps and OPMaps 30 times to account for HI's variability.

We assessed CNN performance on the unseen set reserved for testing, i.e. 38 MSA patients vs. 38 HI.

We devised a 3D CNN similar to the one described in Section 3.2.4 but adapted to the chosen image resolution. Building blocks and implementation details remained unchanged, except for the number of training epochs set empirically to 30 for the amplification approach. We provide the proposed CNN structure in Fig 4.12.
The main differences compared to the architecture presented in Section 3.2.4 are:

- Filter size equal to 3×3×3 instead of 2×2×2 for the first Average Pooling layer;
- Two ConvBlocks with 64 convolutional filters, instead of one block, after the first Average Pooling layer.



(a) Architecture (b) Building blocks

Figure 4.12: 3D CNN proposed for the multi-pattern approach. FC layers receive as input a one-dimensional layer obtained with the flatten operation. BN: Batch Normalization; CNN: Convolutional Neural Network; ELU: Exponential Linear Unit; FC: Fully Connected; FCL: Fully Connected Layer; prob: dropout probability. Figure *b* reproduced from [182] (Giulia Maria Mattia, 2021). CC BY-NC-SA 4.0

4.3.3.2.2 Results

We provide performances of the CNN trained with 20 MSA and 21 randomly selected sets of HI in Fig. 4.13. We can notice that the accuracy varied a little around 0.90, contrary to the other two metrics, which differed according to the HI set.

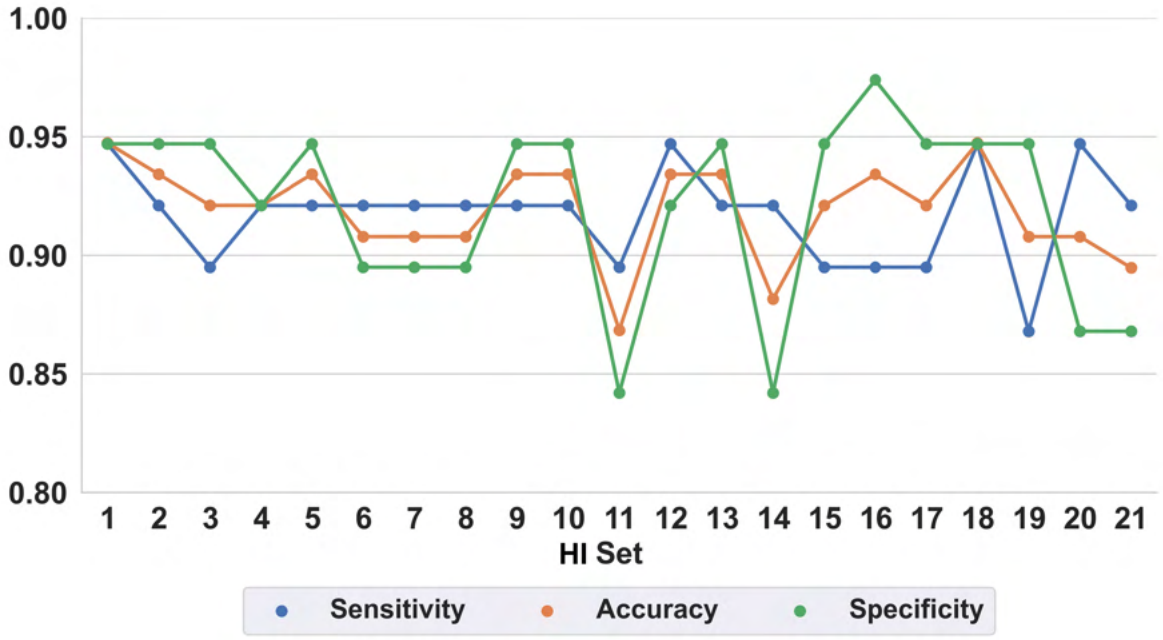


Figure 4.13: *ZB-APMaps for MSA Classification - Multi-Pattern Approach - Reference.* CNN performance according to the set of HI and the same 20 MSA patients used for training, evaluated on the testing set composed of the MD maps from 38 MSA patients and 38 HI. HI: Healthy Individuals; MD: Mean Diffusivity; MSA: Multiple System Atrophy; ZB-APMaps: Z-score-Based Altered Parametric Maps

We can observe from Table 4.6 that CNN performance was higher when training with the original set of MSA/HI (Reference) compared to the training set composed of 20 ZB-APMaps and 20 OPMaps (Multi-pattern).
The best accuracy was achieved by the TPat variant, whereas for the sensitivity, T1.5 reached the best score of 0.80.
Specificity was comparable between no threshold and TPat variants, amounting to around 0.95. The lowest specificity was instead equal to 0.82 for T1.5.

CNN Training	ZB-APMaps Threshold	Accuracy	Sensitivity	Specificity
Reference (20 MSA vs. 20 HI)	-	*0.92 (0.02)*	*0.92 (0.02)*	0.92 (0.04)
Multi-Pattern (20 ZB-APMaps vs. 20 OPMaps)	None	0.83 (0.05)	0.71 (0.09)	0.95 (0.04)
	T1.5	0.81 (0.05)	0.80 (0.08)	0.82 (0.08)
	TPat	0.85 (0.06)	0.75 (0.12)	*0.95 (0.03)*

Table 4.6: *ZB-APMaps for MSA Classification - Multi-Pattern Approach.* Comparison between CNN performances considering the reference (training with 20 MSA patients and 20 HI) and the multi-pattern approach (training with 20 ZB-APMaps and 20 OPMaps) according to the different thresholds applied for the creation of ZB-APMaps. Results are provided as mean (SD) obtained on the testing set, composed of the MD maps from 38 MSA patients and 38 HI. Best performances are highlighted in italic. CNN: Convolutional Neural Network; HI: Healthy Individuals; MD: Mean Diffusivity; MSA: Multiple System Atrophy; SD: Standard Deviation; T1.5: Threshold equal to 1.5 multiplied by the healthy individual's image value; TPat: Threshold equal to the MSA patient's image value; ZB-APMaps: Z-score-Based Altered Parametric Maps

Fig. 4.14 illustrates performance comparison according to the threshold on ZB-APMaps and the number of samples per class in training.

Accuracy increased with the number of samples for T1.5, whereas was pretty steady for the other variants, keeping a value between 0.80 and 0.90. T1.5 reached the highest accuracy (0.88) with 120 ZB-APMaps in training.
Sensitivity decreased slightly with the increasing number of samples, amounting around 0.80 for T1.5 and 0.75 for the other two thresholds.
The version with no threshold achieved the highest specificity (> 0.95) for all numbers of samples.

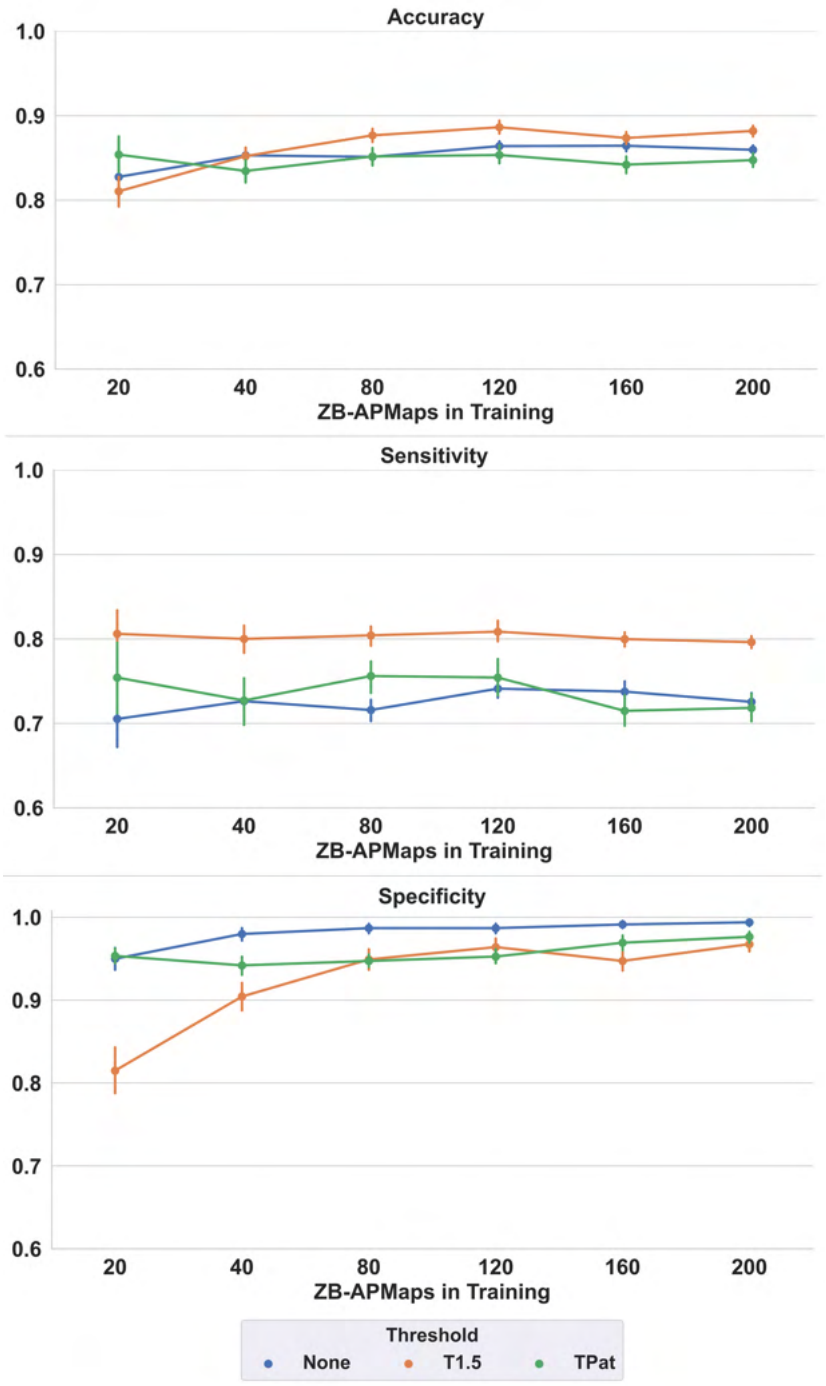


Figure 4.14: *ZB-APMaps for MSA Classification Multi-Pattern Approach - Amplification.* Performance metrics given as mean and SD over the 30 repetitions, obtained on the testing set composed of the MD maps from 38 MSA patients and 38 healthy individuals according to the threshold applied to ZB-APMaps and the number of MSA patients in training (same number of HI). CNN: Convolutional Neural Network; HI: Healthy Individuals; MD: Mean Diffusivity; MSA: Multiple System Atrophy; SD: Standard Deviation; T1.5: Threshold equal to 1.5 multiplied by the HI's image value; TPat: Threshold equal to the MSA patient's image value; ZB-APMaps: Z-score-Based Altered Parametric Maps

4.3.3.2.3 Discussion

The multi-pattern approach enabled us to explore the discriminating power of ZB-APMaps used as input to a 3D CNN in the case of different MSA patterns provided in training. To establish a comparison, we considered the network trained with data belonging to MSA patients. Furthermore, we progressively increased the number of ZB-APMaps per MSA pattern to discover whether that would lead to better discrimination between MSA patients and healthy individuals.

Regarding the reference network trained with the MSA/HI set of 20 samples per class, we obtained a mean accuracy of 0.92, in line with the results of previous studies about the discrimination between MSA patients and HC [41,206]. Although the training set comprised fewer MSA patients than the test set, these results were encouraging as most HI and patients were correctly classified (sensitivity and specificity > 0.90). Despite the different sets of HI, performances were still good enough, showing no particular bias relative to the random selection of healthy individuals.
Concerning the multi-pattern approach with an equal sample size to the reference approach (see Table 4.6), the former was less effective in differentiating MSA patients from HI.
Worth noticing is that despite comparable accuracy of around 0.83, performances of the ZB-APMaps variants differed according to the threshold, especially regarding specificity and sensitivity. That suggests that the content enclosed in the ZB-APMaps or retrieved by the CNN was not as informative as pathological data. However, we must remember that the MD map of each healthy individual could encompass peculiar characteristics leading to an MSA pattern not entirely mimicking the original. Indeed, we were interested in creating different versions of the same pattern, and not just a simple copy, by exploiting each subject's characteristics.
Increasing the representation of each pattern in training by including a variable number of ZB-APMaps mimicking the same pattern did not ameliorate performances compared to the reference. Nevertheless, we found a slight improvement with a higher representation per pattern (from 0.83 with no amplification to 0.88 with amplification by ten for the ZB-APMaps variant with no threshold). However, none of the variants considering the amplification outperformed the CNN trained with the original data from the MSA patients.

These findings show us how complex the process of refining altered data is and how small the benefit we gain from it can be. We would certainly need to further confirm these results by applying our method to other diseases or exploring different modification strategies.
Nevertheless, there are some advantages to the ZB-APMaps, such as the high tailoring degree

and the straightforward application of the z-score to the whole brain from MSA patients to healthy individuals.

One strength of the proposed approach relies upon the use of healthy subjects from multiple centers, increasing the feasibility when data are limited. Although we did not perform any harmonization between pathological and normal data, we identified no deterioration in performance that may be attributable to this aspect. We believe that mixing all samples independently from the acquisition center somehow compensated for the noisy components undoubtedly present. Furthermore, using a quantitative parametric map aided this cause owing to the universal physical significance.

4.3.4 Conclusion

This experimental part aimed to prove the many uses APMaps can provide for MSA classification. Most remarkable is that the overall performances reached by training a 3D CNN with altered parametric maps and testing on a set of MSA patients and HC showed not only satisfactory performance (accuracy > 0.84) but also competitive results with previous work [41, 206]. Let us discuss some crucial points while Table 4.7 summarizes the advantages and drawbacks of our experiments.

Pathology-agnostic APMaps surprised us with their capacity to provide the network with relevant information (best accuracy = 0.88). This approach brings a glimpse of hope as relying on a priori knowledge of a disease (in our case, MD increase in specific brain regions) may offer a valuable alternative to detect similar traits in pathological data. This aspect gains importance when there is a paucity of data characterizing, for instance, rare diseases such as MSA.
Worth reminding is that the creation of pathology-agnostic APMaps respects the physical significance of the chosen parametric maps, thereby granting meaning and validity to these altered data, regardless of their general character.

As the ultimate goal is to retrieve pathological patterns, we deepened our approach by creating pathology-oriented APMaps, including specific MSA features. Similarly to pathology-agnostic APMaps, CB-APMaps contained region-specific alterations coming, instead, from the pathology. Their main limitation is that they do not enclose a more global pattern by focusing on a single region. However, there is an upside to a higher level of interpretability since we assume that the network searches for specific regional alterations. That is when ZB-APMaps came into play with the added value of whole-brain alterations.
Considering only the ZB-APMaps with one pattern in training, we achieved great performances (best accuracy = 0.85) when testing on the pathological set, giving us a hint about

the presence of shared features among these MSA patients. Typical pathological data are characterized by high heterogeneity, which hinders the network from perfectly classifying all samples, as it constitutes an average prototype of each class to optimize performances. We explored this condition with the multi-pattern approach leading to improved performance (best accuracy = 0.89 vs. 0.92, using MSA patients and HI in training data). Furthermore, amplifying each pattern *n* times brought a slight improvement without a drastic change.

Regardless of these promising findings, further work is needed to fully understand the implications of this approach. To our knowledge, this was the first study advancing the use of altered data to classify pathological data using a 3D CNN. Future research should be undertaken to establish whether we can apply this method to other diseases without performance deterioration.
At this point, some questions can be raised: How to choose the most suitable type of APMaps? Can our approach compete with more sophisticated methods such as GANs? To what extent does CNN performance depend on the considered disease? Is this generalization ability extensible to unseen cohorts? Further work is needed to elucidate these aspects.
The limitation of the restricted sample size is another aspect to account for that will be at the core of the following sections.

CNN Training Data			Best Accuracy	Pros	Cons
Pathology-Oriented APMaps	Pathology-Agnostic APMaps		0.88	✓ Alterations independent of any specific pathological pattern	✗ *Region-specific* ✗ *Identification of regional alterations rather than specific pathological patterns* ✗ *A priori knowledge of the disease required*
	ZB-APMaps	CB-APMaps	0.84	✓ Specific pathological traits ✓ Production of new pathological images considering the reference pattern (e.g. cluster mean)	✗ *Region-specific* ✗ *Same patients used for creating CB-APMaps and for testing* ✗ *Same pattern repeated*
		One-Pattern	0.89	✓ Whole-brain alterations ✓ Discriminating power of a single pattern	✗ *Same patients used for creating ZB-APMaps and for testing* ✗ *Single pattern in training (low variability)*
		Multi-Pattern: Without Pattern Amplification	0.85	✓ Whole-brain alterations ✓ Higher image resolution ✓ Separate sets of patients for training and testing ✓ Multicentric approach ✓ Multiple patterns in training *(higher variability)*	✗ *Difficulty in determining the most suitable variant of ZB-APMaps* ✗ *No considerable gain in performances with pattern amplification*
		Multi-Pattern: With Pattern Amplification	0.89		

Table 4.7: *APMaps for MSA Classification.* Main pros and cons for each type of APMaps used for CNN training with the best-obtained accuracy for comparison. APMaps: Altered Parametric Maps; CB-APMaps: Cluster-Based Altered Parametric Maps; CNN: Convolutional Neural Network; HI: Healthy Individuals; MSA: Multiple System Atrophy; ZB-APMaps: Z-score-Based Altered Parametric Maps

4.4 Impact of Small Sample Size on MSA Classification

One of the foremost concerns of DL-based applications is the inability of a deep network to generalize when trained with a small training set. In the previous chapters, we also coped with this aspect by evaluating the discriminating power of altered data, either pathology-agnostic or pathology-oriented, obtaining encouraging results (best accuracy slightly lower than 0.90 to discern MSA patients from HC). These results were unforeseen in light of the limited sample size and the use of altered brain data for training the network. That is why we decided to take a step back and refocus on the analysis of MSA by directly considering the original pathological data as input to the CNN. We proposed to investigate this scenario by examining the behavior of different CNN architectures when faced with a restricted number of samples for the classification between MSA patients and HI.

Strictly correlated to small sample sizes and pathological data is the issue of data heterogeneity. We already looked into this aspect, first with the CB-APMaps based on clusters of MSA patients with similar MD distribution in the cerebellum (Section 4.3.2) and then with the multi-pattern approach, encompassing different degrees of representation for each pattern (Section 4.3.3.2). We have concretely seen that this heterogeneity may complicate pattern retrieval for a deep network, as each patient conveys peculiar information about the disease. To address this issue, we proposed to cluster patients according to the degree of alteration compared to healthy subjects and then use each cluster to train and test the network. This strategy led to having more control over training data, thus enabling a better interpretation of CNN behavior. Fig. 4.15 illustrates the salient points of this section.

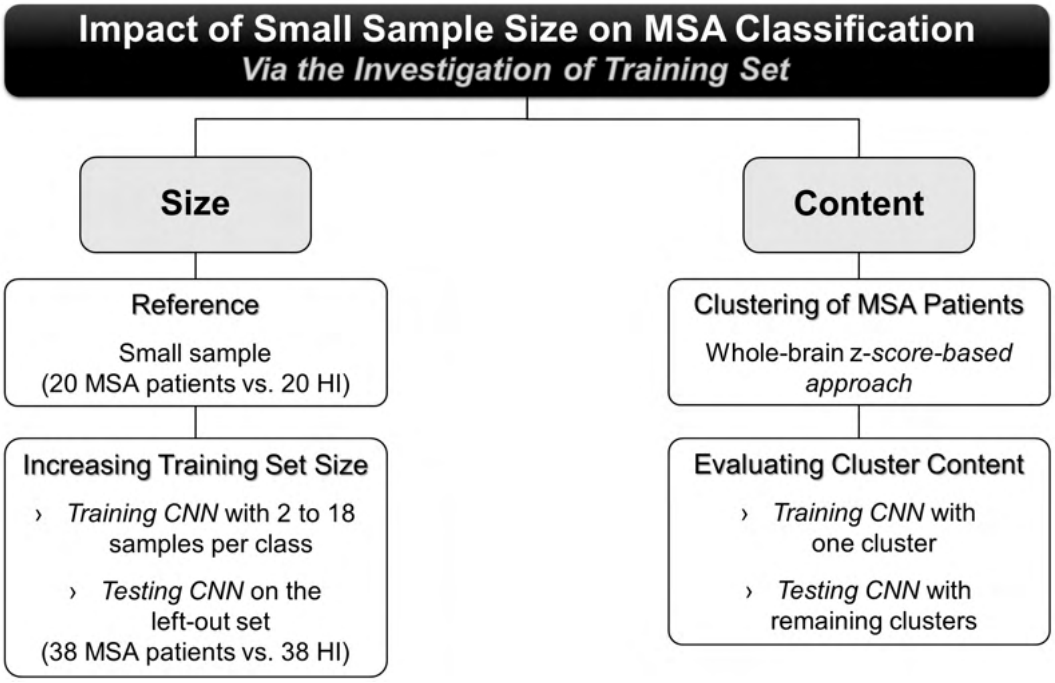


Figure 4.15: *Impact of Small Sample Size on MSA Classification.* We investigated the effect of a small sample size for classifying a rare disease, such as MSA, in two steps: 1) We investigated CNN performances by gradually increasing the number of samples in training from 2 to 18 per class, considering a set of 20 patients and 20 HI. We tracked performances on a left-out set of MSA patients and HI. 2) We fed the network with different training content based on a prior clustering of MSA patients while testing the remaining others. We tracked performances on a left-out set of MSA patients and HI. CNN: Convolutional Neural Network; HI: Healthy Individuals; MSA: Multiple System Atrophy

4.4.1 Investigation of Training Set Size

Notably discussed in deep learning applications is the necessity to use a high number of training samples for a system to work well or, in other words, to be capable of generalization. Recent studies have shown that deep learning methods can be efficient even with fewer training samples [161, 162]. These findings are encouraging, especially for fields burdened by a paucity of data, such as the medical domain.

In this work, we investigated the influence of training set size for three different CNN architectures to discern between patients with MSA and healthy individuals. As already mentioned, MSA is a rare neurodegenerative disease, thus qualifying as the perfect candidate to analyze this aspect since there will always be a gap in the amount of data proportional to its prevalence.
Before getting into the matter, we provide a small summary with a schematic diagram in Fig. 4.16:

1. *Reference.* Keeping a small number of samples (20 MSA patients vs. 20 HI), we trained the networks to establish a reference performance by testing on a left-out set of patients and HI (38 MSA patients vs. 38 HI);

2. *Increasing training set size.* Considering our small sample (20 MSA patients vs. 20 HI), we created 30 random subsets of varying sizes (2 to 18 samples per class) and evaluated performances on the same left-out set for comparison.

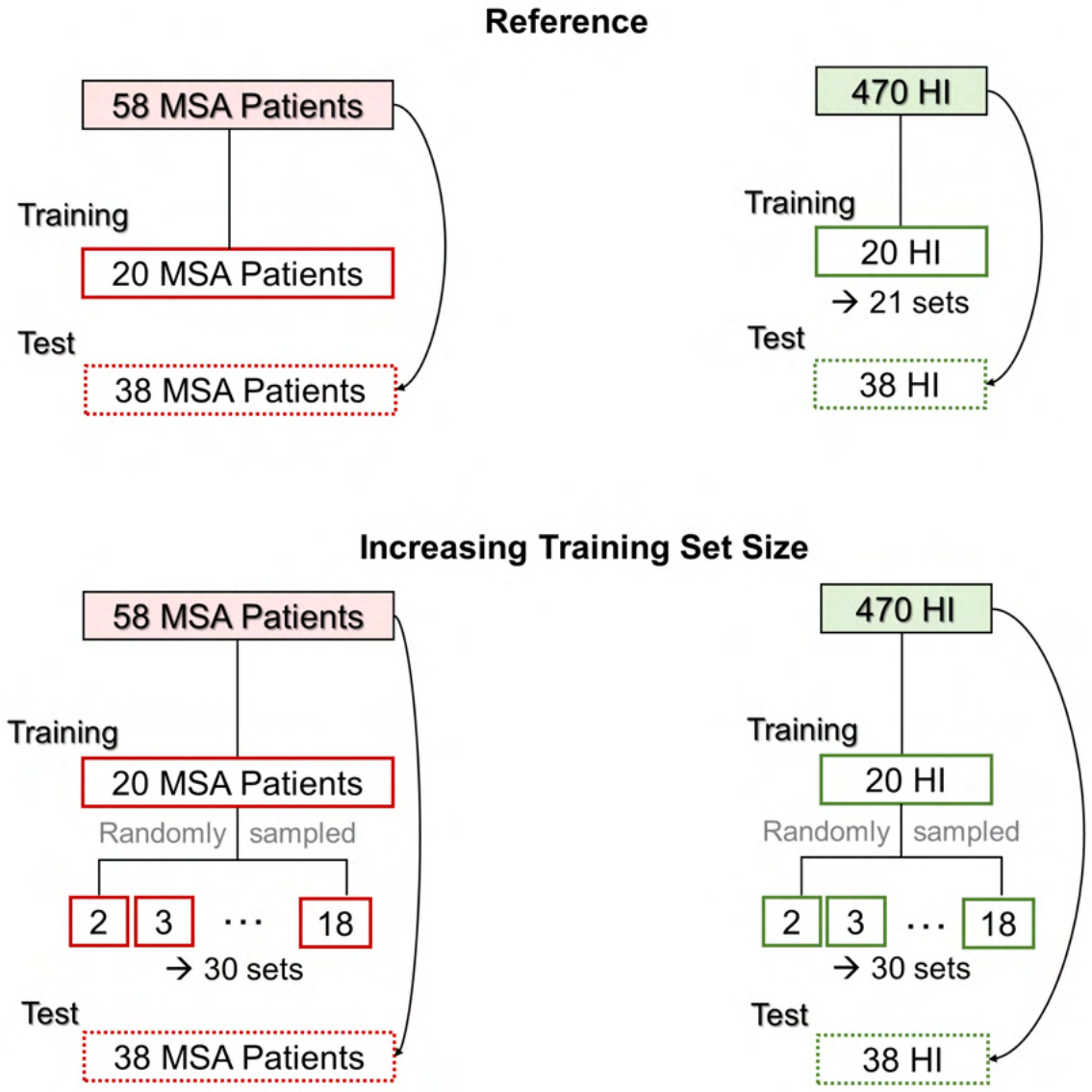


Figure 4.16: *Investigation of Training Set Size.* Diagrams representing dataset split for training and testing with two strategies: 1) *Reference*. We randomly sampled data from MSA patients and HI, to establish the reference performance obtained by training the network with 20 MSA patients and 21 different sets of HI; 2) *Increasing training set size*. We randomly selected an increasing number of samples per class from the set of 20 MSA patients and 20 HI, obtaining 30 subsets for each sample size. For both strategies, we tested the networks on the same set of 38 MSA patients and HI to allow for comparison. All samplings were performed randomly. Choosing such a small sample size (only 20 examples per class) places this approach in a realistic situation, e.g. in the case of a rare disease such as MSA. CNN: Convolutional Neural Network; HI: Healthy Individuals; MSA: Multiple System Atrophy

4.4.1.1 Material and Methods

4.4.1.1.1 Datasets

We used the MD maps from the 58 MSA patients and 470 healthy subjects presented in Section 4.3.3.2.1.1. Image processing remained unchanged.

4.4.1.1.2 CNN Implementation

For this experimental part, we introduced different CNN architectures to determine whether this could change the outcome. We proposed our implementation of two famous architectures, GoogLeNet [71] and ResNet [51], in addition to our model inspired instead by VGGNet [92]. We named each model after the corresponding well-known CNN architecture.

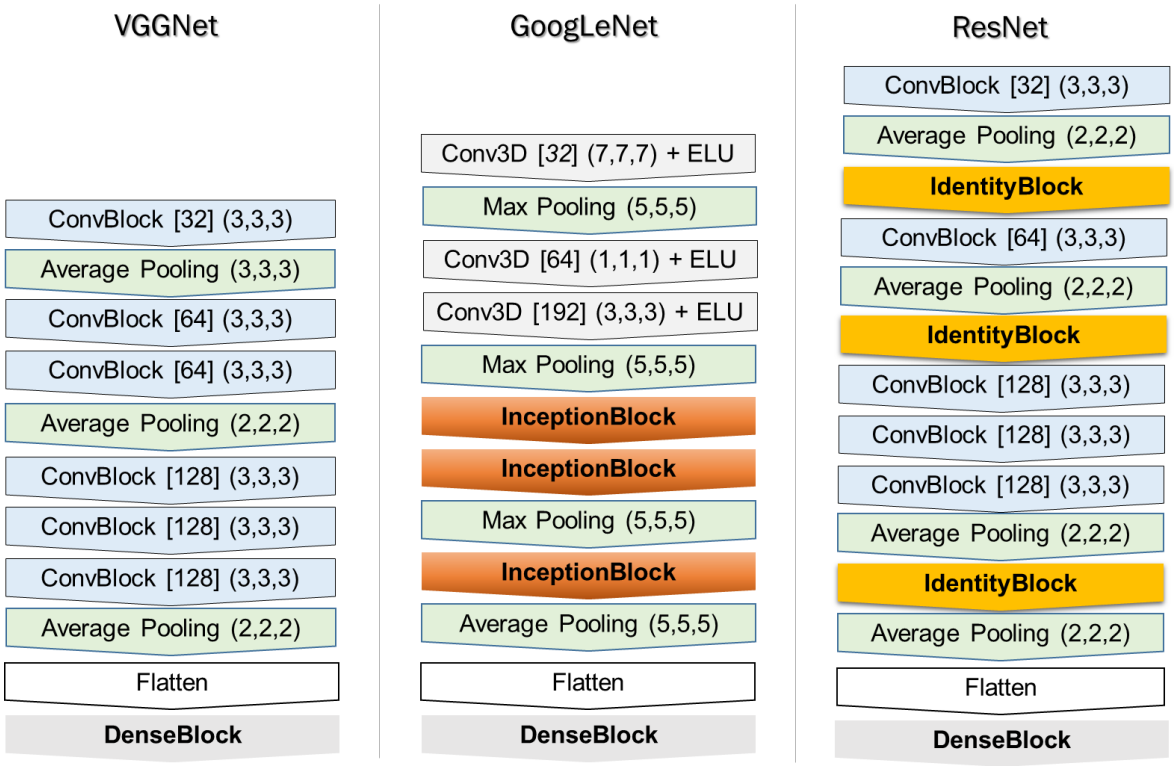


Figure 4.17: Proposed CNN architectures named after the corresponding well-known model. Details about each building block are available in Fig. 3.7b. Average Pooling: Average Pooling layer; Conv3D: Convolutional layer: ConvBlock: Convolutional layer Block; CNN: Convolutional Neural Network; DenseBlock: block containing fully connected layers; ELU: Exponential Linear Unit; Flatten: operation to reshape in a one-dimensional vector; IdentityBlock: block characteristic of ResNet; InceptionBlock: block characteristic of GoogLeNet; Max Pooling: Max Pooling layer; [filter number]; (filter size); dropout probability

Fig. 4.17 offers a comparison of CNN structures, whereas the building blocks are available in Fig. 4.18 and 3.7b. We detail the filter number of convolutional layers for GoogLeNet and ResNet in Table 4.8.
Implementation details remained unchanged from those reported in Section 3.2.4.

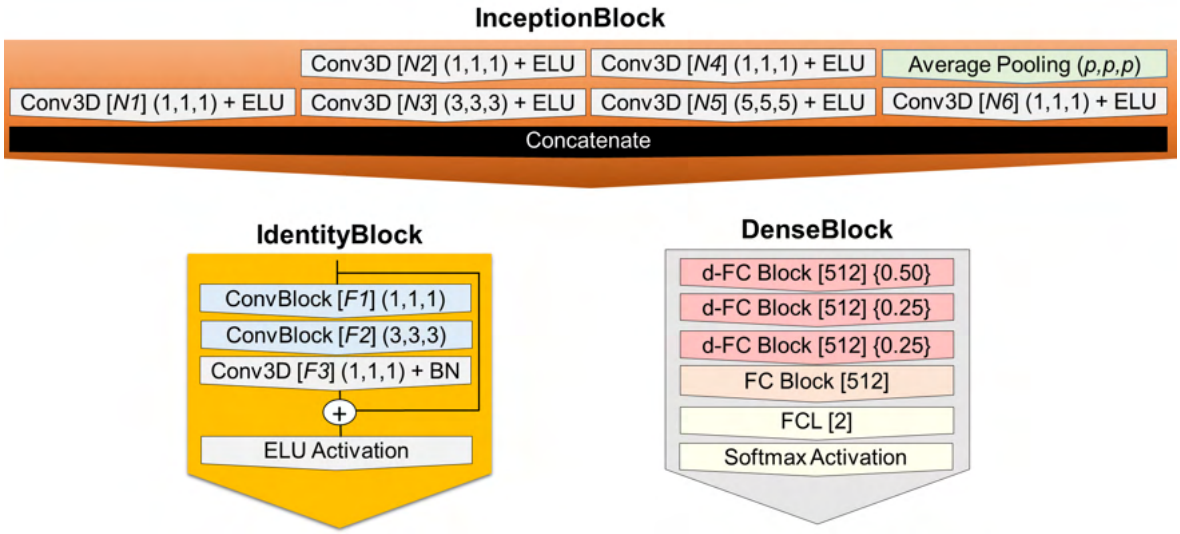


Figure 4.18: Building blocks for CNN architectures. The first two convolutional layers of the InceptionBlock present the same filter number for both the proposed versions of GoogLeNet. p is equal to 3 and 2 for image resolution equal to 2 mm and 3 mm per direction per voxel, respectively. BN: Batch Normalization; Conv3D: Convolutional layer; CNN: Convolutional Neural Network; ELU: Exponential Linear Unit; FC: Fully Connected; FCL: Fully Connected Layer; prob: dropout probability.

Table 4.8: Details on the number of convolutional filters for GoogLeNet (InceptionBlock) and ResNet (IdentityBlock)

(a) InceptionBlock

	Conv3D Layer		
Filters	**#1**	**#2**	**#3**
N1	64	128	192
N2	96	128	96
N3	128	192	208
N4	16	32	16
N5	32	96	48
N6	32	64	64

(b) IdentityBlock

	Conv3D Layer		
Filters	**#1**	**#2**	**#3**
F1	64	128	256
F2	64	128	256
F3	32	64	128

As represented in Fig. 4.16, we organized CNN implementation as follows:

1. *Reference.* This part established a reference as we trained the networks on the set of MSA patients and HI, considering the maximum number of samples per class that we set to 20.

    - *Training.* We trained the CNN with the MD maps from 20 MSA patients and 21 different sets of 20 HI, randomly selected from the HI set with no overlap.
    - *Testing.* We tested the networks on the remaining 38 MSA patients and 38 randomly selected HI left out from the beginning. We computed accuracy, sensitivity, and specificity for each of the 21 sets used in training.

2. *Increasing training set size.* To progressively increase the training set size, we proceeded as follows:

    - *Training.* We randomly selected a set of 20 MSA patients and 20 HI. Keeping as reference only 20 samples per class makes this approach realistic considering the quantity of data available for rare diseases. We created 30 non-overlapping random subsets to obtain 2 to 18 samples per class. We trained the CNN to perform binary classification between MSA patients and HI with each of these sets.
    - *Testing.* We tested the networks on the remaining 38 MSA patients and 38 randomly selected HI. We provide the mean and SD of accuracy, sensitivity, and specificity over the 30 subsets for each training set size.

4.4.1.2 Results

We investigated the effect of training set size by considering three CNN architectures, each inspired by a well-known model.

Fig. 4.19 provides a comparison for each metric between the three models, trained with 20 MSA patients and 21 different sets of 20 HI. We can observe that performances did not differ much according to the model. Sensitivity and accuracy presented few variations around 0.95, whereas there was higher variability for specificity according to the HI set.

Training accuracy was higher than 0.95 for all training set sizes and CNNs.
Fig. 4.20 offers a comparison between CNN architectures for each metric. The worst-performing model was GoogLeNet, although it reached an accuracy of 0.90 with 18 samples. VGGNet and ResNet presented quite similar behavior. The latter achieved higher sensitivity and accuracy than the former for most sample sizes. Overall, the mean accuracy was around 0.70 with just two samples per class, whereas it was higher than 0.80 with only five samples per class.
We can notice that increasing the training set size improved performances as well. From 10 patients in training, the accuracy kept rising over 0.90. The sensitivity was instead slightly lower than the specificity.

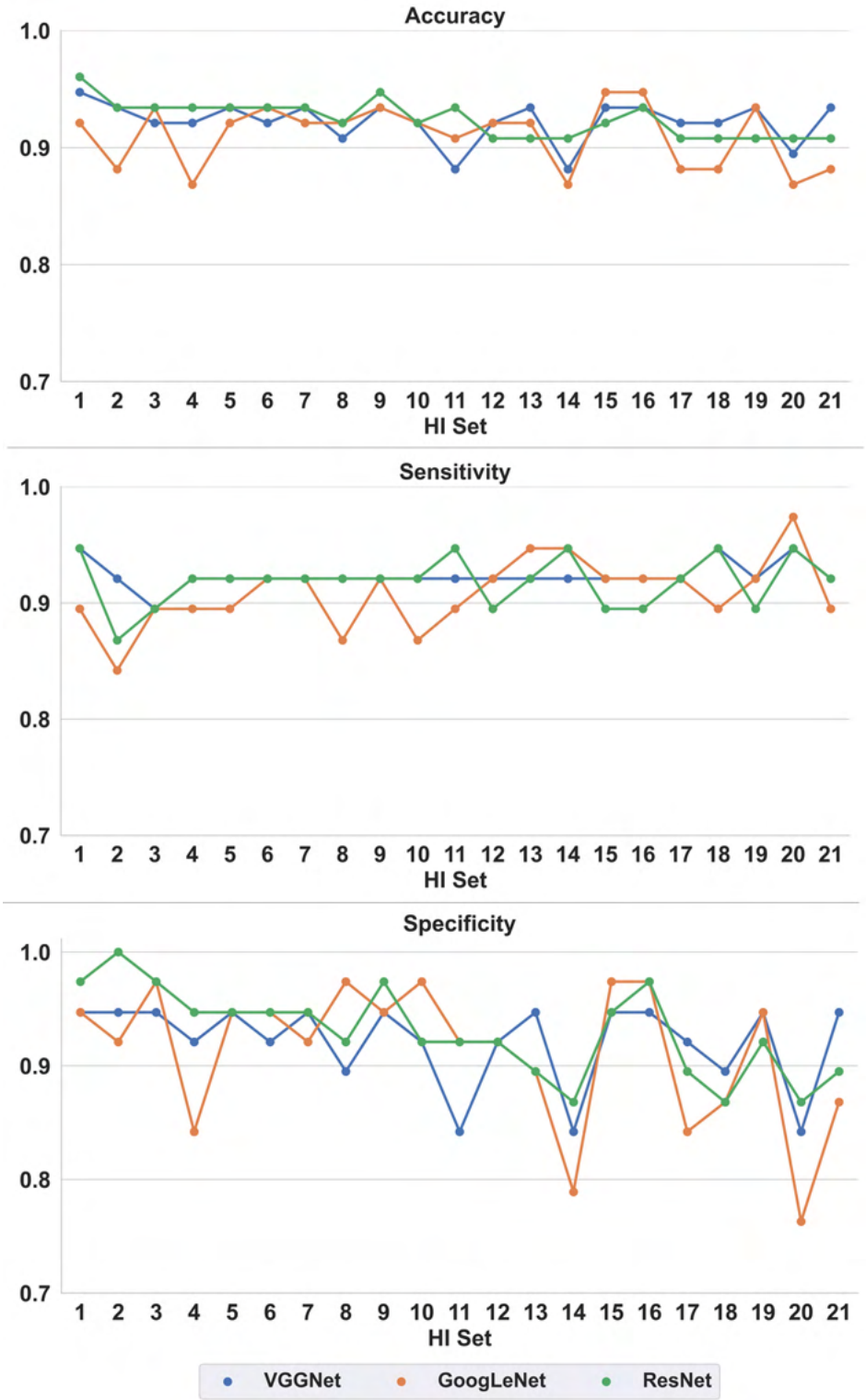


Figure 4.19: *Investigation of Training Set Size - Reference.* Performance comparison between CNN models according to the set of HI used in training with the fixed set of 20 MSA patients, evaluated on the test set composed of the MD maps from 38 MSA patients and 38 HI. CNN: Convolutional Neural Network; HI: Healthy Individuals; MD: Mean Diffusivity; MSA: Multiple System Atrophy

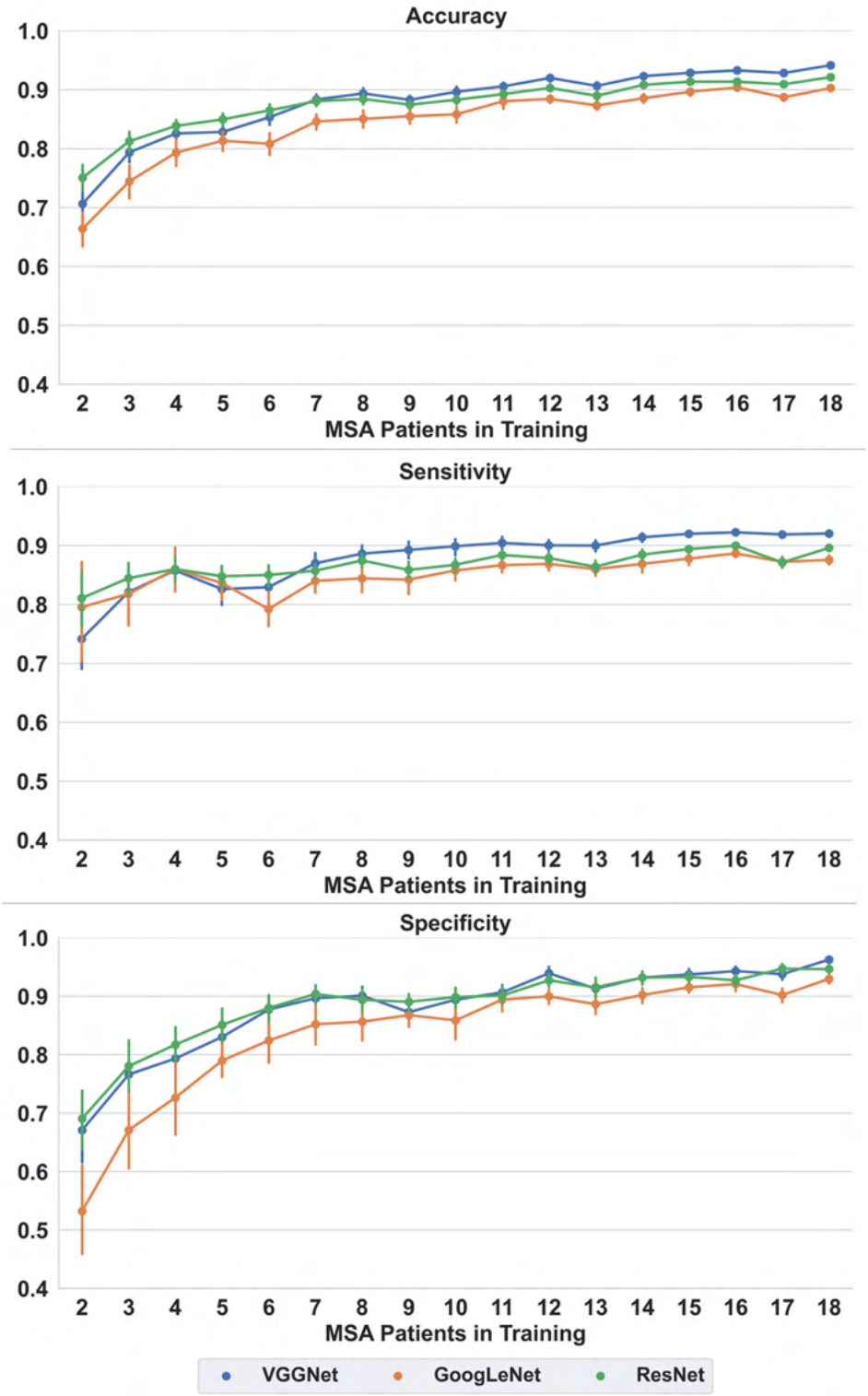


Figure 4.20: *Investigation of Training Set Size - Increasing Training Set Size.* Performance comparison between CNN models on the test set (38 MSA patients vs. 38 HI) provided as mean and SD considering 50 subsets according to training set size (the number of HI was equal to the number of MSA patients). CNN: Convolutional Neural Network; HI: Healthy Individuals; MSA: Multiple System Atrophy; SD: Standard Deviation

4.4.1.3 Discussion

This work shed light on the importance of training set size in the case of a limited amount of data, a debated point in the deep learning community. We proposed to analyze a rare disease, such as MSA, varying the quantity of training data to determine how the models' performance changed accordingly. To keep in line with a realistic situation with only a few samples available, we considered as a reference 20 MSA patients and 20 HI, randomly selected from the datasets at our disposal. To further validate our approach, we implemented three different CNN models for comparison.

First, we looked at the performances when changing the set of HI in training, keeping the same MSA patients (Fig. 4.19). Despite some variations, there was no considerable difference among these sets. However, the most unstable metric was the specificity, i. e. the ability to correctly classify HI, allegedly suggesting that some healthy subjects may present common characteristics with patients. One explanation could be related to the fact that older people can present with signs of reduced tissue microstructural integrity inevitably due to aging.
Surprisingly, we reached an accuracy of 0.90 with only ten samples per class for all models (Fig. 4.20). Considering the reference approach, we can also observe that the performance with 18 samples per class was comparable to that obtained by training the models with 20 MSA patients and 20 HI (see Fig. 4.13).

Remarkable is the absence of significant differences in the trends presented by the proposed CNNs. Given the infinite possibilities in the choice of CNN architectures, we may attribute this to the specific characteristics of MSA patients' data. The latter comprise features setting them apart from healthy subjects, thus making them more easily discernible. Nonetheless, that is not always the case for other pathologies, hence the interest in extending our approach to different diseases.
These findings are encouraging as we achieved incredibly satisfactory performances on unseen data despite the small sample size. The random sampling strategy for training also contributed to providing a robust estimate.
Worth noticing is that the higher the number of samples, the smaller the standard deviation for all metrics. As the models received more and more information with increasing training set size, the level of uncertainty progressively decreased.

4.4.2 Investigation of Training Set Content

All the previous work has shown us how sensitive a CNN can be depending on the information delivered by the training data. For instance, consider the application of CB-APMaps and the different performances obtained according to the cluster used for creating the altered maps (Section 4.3.2). This behavior offered us a lead to follow: we can calibrate training content to track CNN performance by grouping patients before feeding data to the network. Moreover, one can argue that using non-original pathological data could have somehow impacted the outcome. Therefore, in this case, we focused on the original data of MSA patients to see if performance could benefit from such an approach. Another difference is that we created CB-APMaps by considering the distribution of MD values in a specific region. Instead, we based the clustering proposed in this experimental part on a whole-brain approach, considering z-score values.
We can summarize our approach for the investigation of training set content in two fundamental steps:

1. Clustering patients according to z-score values from the MD maps of MSA patients with respect to HI, selecting those within a significance level of 5%, thus with a significant deviation from the average;

2. Training the models with each cluster while testing the others to see how performances changed.

4.4.2.1 Material and Methods

4.4.2.1.1 Clustering of MSA Patients

We performed clustering on the MD maps from the 58 MSA patients, whose data and preprocessing steps are available in Section 4.4.1.1.1.
To group patients according to the degree of alteration, we followed the steps highlighted in Fig. 4.21:

1. Computation of the z-score of each patient considering the mean and standard deviation of the healthy individuals, as in (4.1);

2. Application of a threshold on z-score values corresponding to a significance level of 5% for a one-tailed positive distribution (i.e. > 1.645). We were interested only in positive increases due to reduced microstructural integrity [41, 168];

3. Computation of the median and count of z-score values above the threshold;

4. Application of k-means considering the median and counts for each patient.

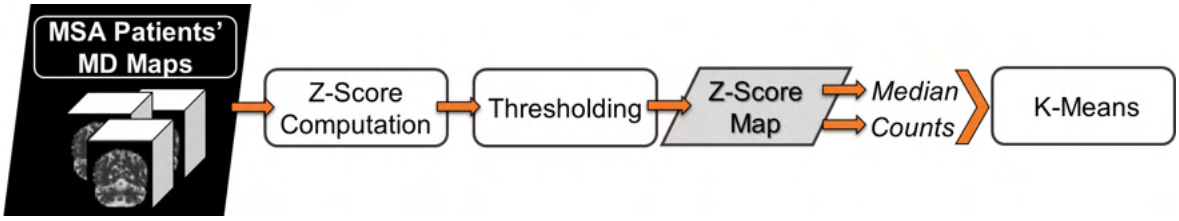


Figure 4.21: *Investigation of Training Set Content.* Main steps to perform clustering on MSA patients. We applied a threshold to z-score values to consider only significant deviations from the mean of healthy individuals. We then performed k-means by considering the median and count of z-score values above the threshold. MSA: Multiple System Atrophy; SD: Standard Deviation

4.4.2.1.2 CNN Implementation

We employed the 3D CNNs described in Section 4.4.1.1.2. For this application, we organized the implementation as follows:

- *Training.* We trained each network with every cluster of MSA patients and 30 random samplings from the HI data in the same number of MSA patients to obtain a balanced set.

- *Testing.* We isolated a set of HI for each cluster of MSA patients to function as a separate testing set. We tested each CNN considering the clusters not used for training. We provide the mean and SD over the 30 random samplings in terms of accuracy, specificity, and sensitivity.

4.4.2.2 Results

4.4.2.2.1 Clustering of MSA Patients

We chose the number of clusters $k = 3$ to allow for comparison between three different degrees of alteration. Fig. 4.22 depicts a scatter plot with the number of significant voxels as a function of the median z-score values for $k = 3$. We named clusters Mild, Intermediate, and Severe going from lower to higher median values and corresponding counts according to an increasing degree of alteration.

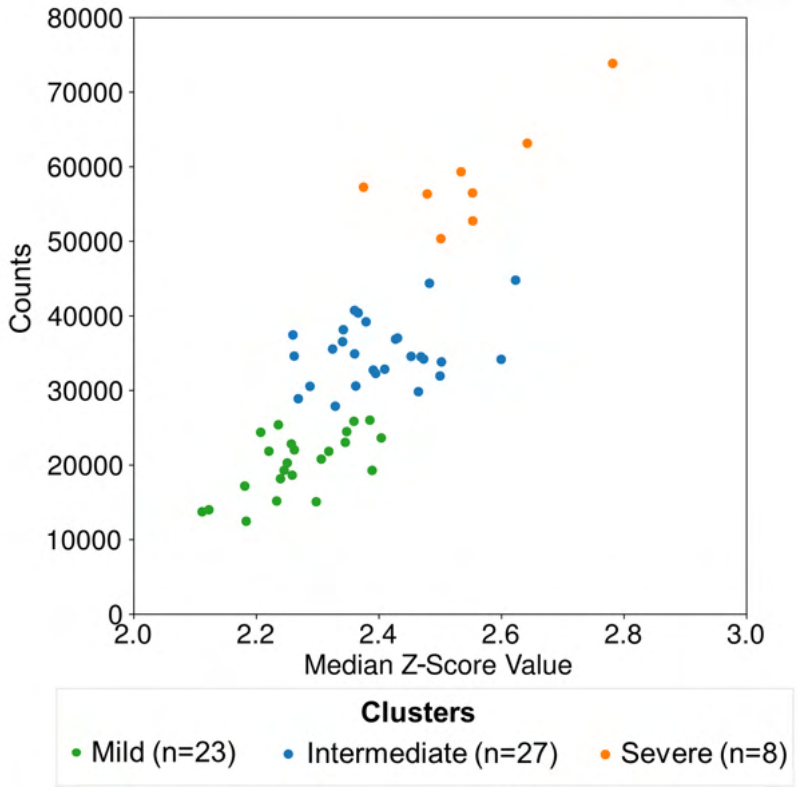


Figure 4.22: *Investigation of Training Set Content.* Scatter plot showing clusters of MSA patients according to median value and count of the z-score above the threshold. MSA: Multiple System Atrophy

4.4.2.2.2 CNN Performance

We provide in Fig. 4.23 VGGNet performances according to the cluster of MSA patients considered in training. In light of the comparable performances, we provide the results for GoogLeNet and ResNet in Appendix B.

Training performance was higher than 0.90 fo all models and clusters. Focusing on accuracy, we can observe that it is high when considering the CNN trained with the Mild cluster on both the Intermediate and Severe clusters. The Intermediate cluster performed well on the Mild with accuracy equal to 0.78 and 1.00 on the Severe. Instead, considering the Severe, the maximum accuracy was only 0.76 on the Intermediate cluster.

Sensitivity showed scores comparable to accuracy, whereas specificity was overall high (> 0.90).

Given no considerable variation in the standard deviation over the 30 repetitions for every tested model (< 0.15), we provide these results in Appendix B.

Figure 4.23: *Investigation of Training Set Content.* CNN performances according to the cluster of MSA patients used for training and testing. Mean for each metric provided considering the 30 random samplings from the set of healthy individuals used for training. Blue and red represent respectively poor and high performances. Notice how the accuracy scored by the CNN trained with the Mild clusters was excellent on both the Intermediate and Mild clusters. CNN: Convolutional Neural Network MSA: Multiple System Atrophy; SD: Standard Deviation

4.4.2.3 Discussion

In this study, we investigated the performance of three different CNNs feeding as input training sets with varying degrees of significant modifications of MSA patients compared to healthy individuals. Exploiting the z-score computed for each MSA patient, we applied k-means clustering to identify groups of patients according to increasingly significant alterations compared to HI (clusters denoted as Mild, Intermediate, and Severe). We tracked the models' performance according to the cluster used in training while testing the others.

Our hypothesis lies in the assumption that a CNN trained to discern patients with milder alterations from healthy subjects would be able to distinguish patients with more severe alterations as well. On the other hand, the opposite would not be the case since more evident alterations become very specific. Our findings seemed to confirm this hypothesis. The Mild cluster led to the best performances for the Intermediate and Severe clusters. The Intermediate cluster achieved good performances on the Mild and Severe, although the accuracy on the Mild was around 0.80. The Severe cluster showed accuracy inferior to 0.50 on the Mild and equal to 0.76 on the Intermediate.
These results gave us insight into the importance of making the network learn from earlier stages of a disease. They even suggest that when the training data enclose more evident modifications, it becomes more difficult for the network to discern healthy controls from patients with milder alterations, as they are more similar to healthy individuals. Furthermore, we found that adopting a different CNN architecture did not change this trend (see Appendix B).
Another point is that each cluster comprised a different number of MSA patients. The smallest was the Severe, with only eight patients, compared with the others, including more than 20 patients. We previously observed that CNN performances with just eight patients in training were still good (accuracy above 0.90, see Fig. 4.20). Therefore, we considered CNN training for the Severe cluster acceptable in this condition.

Despite the restricted sample size, we found a behavior for the networks which could guide future developments. For instance, the proposed clustering approach based on the z-score offers a straightforward way to identify groups of patients with varying degrees of alterations. We could ameliorate this approach further by focusing on regional changes to target more peculiar traits. This type of reasoning may help devise a strategy to create training sets conveying targeted knowledge content to the network, hence better performance and easier interpretation. In this case, we could not apply a majority voting strategy as there was no left-out set for each cluster. Nevertheless, given a higher number of MSA patients, we could compensate for misclassifications by exploiting all the networks trained with the different clusters to combine predictions.
With the clustering strategy proposed in this approach, we were blind to the spatial localization of the most significant alterations, as we inputted the k-means only global features (i.e. counts and median value). It would be interesting to search for a spatial correspondence to turn this to our advantage.

4.4.3 Conclusion

In this investigation, we aimed to study the effect of a small sample size on the classification of a rare disease such as MSA using 3D CNNs. To this end, we progressively varied the sample size for CNN training from 2 to 18 samples per class. We tested CNN performances using a hold-out set to assess the generalization ability. Next, we evaluated the impact of different training content to account for data heterogeneity typical of pathological data. After grouping patients based on z-score features, we used these clusters to train and test the networks and track performances.

These experiments surprisingly revealed great performances even with a small sample size for discriminating MSA patients from HI. We coupled these findings with the evaluation of the training set content. The latter confirmed that if a network is trained with images enclosing milder alterations and can distinguish them from the healthy condition, it would be more probably capable of discerning more severe alterations, unlike the opposite case. This outcome may help develop automated systems to optimize the classification of the early stages of a disease.

Future research might explore the application of these approaches to other pathologies or a differential diagnosis, as we are well aware that the findings discussed so far are strictly related to MSA classification. That is a promising beginning to better grasp CNN's functioning and master its use.

5 CNN for Coma Classification

In this Section, we present an application of 3D CNNs to discern healthy individuals from comatose patients, resulting in the publication of a research article [228]. This study represents one of the first attempts to combine the informative content of mMRI data with the discriminating capacity offered by deep learning techniques.

5.1 Introduction

Among the major causes of death and disability around the world figures acute brain injury inducing coma after Cardiac Arrest (CA) [229]. Over the last decade, no significant change has been made to the treatment of coma patients, even though the research in this field continues to progress [230]. One explanation may be found in the difficulty of accurately describing the damage to brain connectomes due to CA [231]. A better understanding of the processes implicated in functional and structural disruptions of the brain during coma is of paramount importance to give these patients the best care, such as adopting promising precision medicine approaches and increasing our knowledge of these mechanisms.

Multimodal MRI has shown great promise for the investigation of neural processes with fMRI and structural Magnetic Resonance Imaging (sMRI). The latter has been exploited to predict the neurological outcome of coma patients using FA [232] or gray matter morphometry [233]. Putative signatures of consciousness have also been identified using fMRI, either with static or dynamic resting-state connectivity [234–237]. Recent fMRI studies have highlighted the role of posterior parietal (PreCuneus (PreCun) and Posterior Cingulate Cortex (PCC)) and frontal (mesial PreFrontal Cortex (mPFC)) cortices as implicated in the presumed brain mesocircuit responsible for the emergence and maintain of consciousness [238]. Nevertheless, it is still cumbersome to use multimodal MRI in everyday clinical practice due to the complex and time-consuming interpretation process. That is corroborated by the fact that several studies focused either on a small dataset, or employed just one MRI modality (sMRI or fMRI), often considering hypothesis-driven brain regions [232–237, 239].

In this context, AI tools may be of great help to assist in the analysis of mMRI data to overcome subjective readability and identify meaningful signatures. Convolutional neural networks have been widely used to solve different tasks (see Section 1.3.3) but not yet to discriminate between healthy controls and patients in anoxoischemic coma.

The present proof-of-concept study aims at developing and using a CNN to inspect 3D mMRI data for an early assessment of cerebral damage due to anoxoischemic coma. Besides evaluating CNN performances, we tried to gain insight into CNN functioning by applying a recently developed visualization technique to highlight the most relevant regions of interest. We offered a thorough analysis of CNN misclassifications to investigate the added value for the patient's neuroprognostication.

5.2 Material and Methods

5.2.1 Study Design

This prospective study was carried out between march 2018 and may 2020 at the intensive care unit of the University Hospital based in Toulouse (France). Blinded to neuroimaging data, physicians treated patients following the current guidelines. Patients' assessment was performed in normothermic conditions, at least two days (4 ± 2 days) after complete withdrawal of sedation. To be included, patients underwent a behavioral assessment with the Glasgow Coma Scale (GCS) and presented a diagnosis of coma induced by CA (GCS score ≤6 at the moment of admission and motor responses <6). Following the Coma Recovery Scale Revised (CRS-R), the neurological outcome was assessed three months after the hospital admission for each patient. Healthy volunteers were recruited if presenting normal neurological examination and no prior neurological or psychiatric disorder.
The Ethics Committee of the University Teaching Hospital of Toulouse, France (2018-A31) approved this study. All participants, or legal surrogates of the patients, gave written informed consent to take part in the study (Clinical trial identifier: NCT03482115).

5.2.2 Population

Inclusion criteria comprised a diagnosis of coma due to a primary anoxoischemic brain injury (GCS score ≤6 at the moment of admission and motor responses <6). Patients were excluded if presenting head motion >3 mm in translation and 3° in rotation during MRI acquisition. For more details, please refer to previous work [236].

5.2.3 Clinical Outcome

Patient follow-up was carried out until death or three months after CA. The CRS-R was employed as the main outcome measure, which was used for diagnosing the Minimally Conscious State (MCS) according to current guidelines for the evaluation of consciousness disorders in patients reporting severe brain injury [229, 230].

MCS was further classified as "+" or "−" according to a patient's command-following response as previously indicated [230]. A favorable outcome was assigned in the case of "$MCS+$" or "$MCS-$" whereas an unfavorable outcome corresponded to death or Vegetative State/Unresponsive Wakefulness Syndrome (VS/UWS).

5.2.4 MRI Data Acquisition

MRI acquisition was performed with a 3 T scanner (Intera Achieva; Philips, Best, the Netherlands), including vital measures monitoring supervised by a senior intensivist. MRI protocol included 11 min of resting-state functional Magnetic Resonance Imaging (rs-fMRI), 3D T1-weighted, and DWI. An estimation of GM volume was computed by applying voxel-based morphometry on 3D T1-weighted images [233]. White matter integrity was examined through FA and MD maps computed from DTI models [232]. Functional connectivity analysis with a Region Of Interest (ROI) vs. whole-brain approach [236] was performed defining cortical frontal (mPFC) and posterior parietal (PreCun and PCC) as ROIs and using Statistical Parametric Mapping (SPM) (version 12, `http://www.fil.ion.ucl.ac.uk/spm/`). Realignment, slice-time correction, coregistration to corresponding T1-weighted image, and normalization to standard stereotaxic anatomical MNI space were applied to fMRI images as previously detailed [234–236].

5.2.5 3D CNN Implementation

We implemented the 3D CNN represented in Fig. 4.12 and described in Section 4.3.3.1.1.3. In this study, we performed 10-time repeated 10-fold CV to reduce performance bias. To establish the discriminating power of each MR index, we fed the set of controls and patients relative to each MR index as input to the network.

5.2.6 Visual Interpretation

To discover the most salient regions for CNN prediction, we employed the visualization technique described in Section 1.2.5.1.5 [107]. To obtain the visualization maps, we computed the absolute difference between the average of all the maps per class previously normalized by considering only correctly classified samples of the training set for each MR index.

To facilitate visual interpretation, we applied a thresholding step, considering half of the maximum for each visualization map.

5.2.7 Statistical Analysis

We computed performance metrics (accuracy, sensitivity, specificity, Positive Predictive Value (PPV), and Negative Predictive Value (NPV)) using standard formulas and the AUC value for each model (see [228] for more information).
PPV and NPV can be defined using the confusion matrix represented in Fig. 1.15, Section 1.2.3.4 according respectively to (1.20) and (1.21) in Section 1.2.3.4.
Regarding the analysis of misclassified patients, we considered the known outcome three months after the primary brain injury. We calculated the FN good outcome rate as the percentage of FN with good outcome over the total FN count.
Furthermore, we adopted the majority voting technique by assigning the most scored prediction to each sample [240, 241]. This strategy allowed us to establish the potential benefits of merging the outcome from all MR indexes.

An overview of the methods is available in Fig. 5.1.

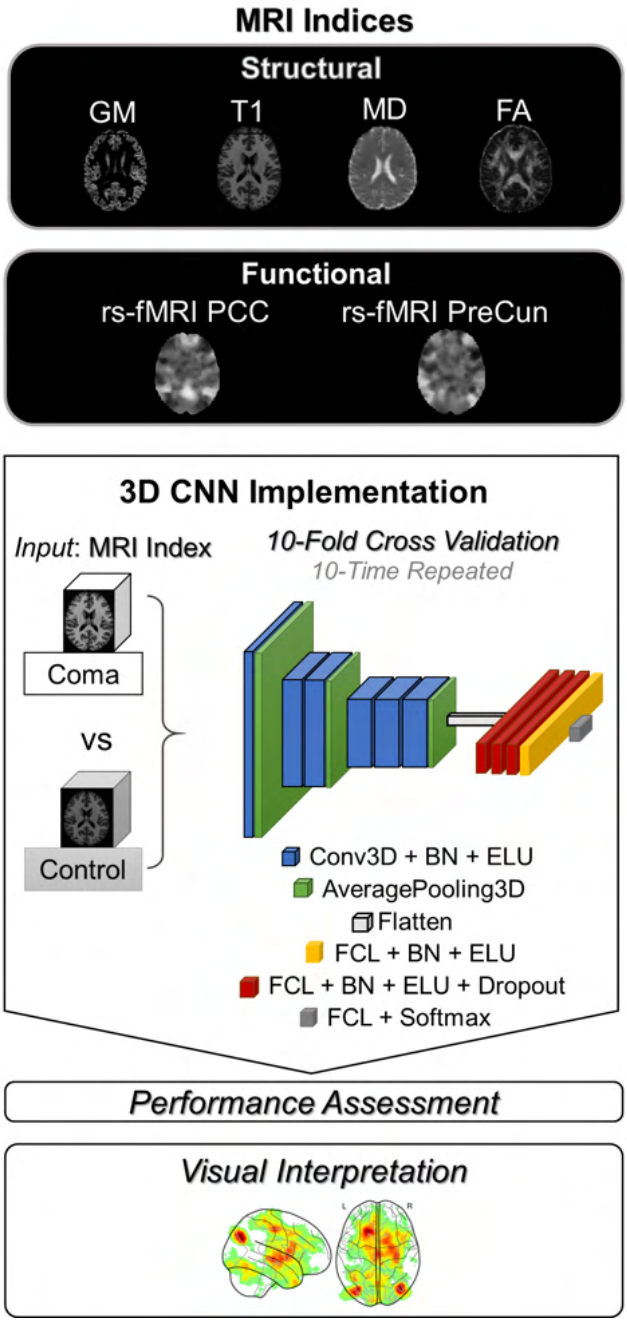


Figure 5.1: *CNN for Coma Classification - Methods Overview.* MRI indices providing functional and structural information were fed as input to a 3D CNN to discern coma patients (n=29) from healthy controls (n=34). We evaluated the performance of each MR index by adopting a 10-time repeated 10-fold cross-validation. We describe CNN architecture with its building blocks. In addition to performance assessment using standard evaluation metrics, we accompanied CNN results by highlighting the most relevant voxels for prediction. AveragePooling3D, Average Pooling Layer; BN: Batch Normalization; CNN: Convolutional Neural Network; Conv3D: Convolutional layer; ELU: Exponential Linear Unit; FCL: Fully Connected Layer; Flatten: operation to reshape the output from convolutional layers in a 1D array; L: Left; Softmax: Softmax activation; R: Right. Adapted from [228]

5.3 Results

5.3.1 Population

At hospital admission, 35 patients in anoxoischemic coma were identified. Five did not meet at least one inclusion criterion, and one withdrew consent. A total of 29 patients aged 62.0 (range: 51.6-75.0) years, comprising 15 women, constituted the final cohort. The latter included 34 healthy volunteers of age 61.0 (range: 51.0-72.1) years (additional details in the supplementary material of [228]).

5.3.2 Model Performance

Table 5.1 reports performance metrics for each MR index.

MR Index	AUC (IC 95)	Accuracy	Sensitivity
GM	0.84 (0.13, 0.81-0.86)	0.84 (0.13, 0.81-0.86)	0.72 (0.24, 0.67-0.76)
T1	0.82 (0.15, 0.79-0.85)	0.82 (0.15, 0.79-0.85)	0.77 (0.25, 0.72-0.82)
MD	0.89 (0.13, 0.86-0.91)	0.89 (0.13, 0.86-0.91)	0.82 (0.23, 0.78-0.87)
FA	0.92 (0.11, 0.89-0.94)	0.92 (0.11, 0.89-0.94)	0.86 (0.20, 0.83-0.90)
rs-fMRI PCC	*0.96 (0.08, 0.94-0.98)*	*0.96 (0.08, 0.95-0.98)*	*0.95 (0.12, 0.93-0.97)*
rs-fMRI PreCun	0.90 (0.12, 0.88-0.93)	0.90 (0.12, 0.88-0.93)	0.88 (0.20, 0.84-0.91)

MR Index	Specificity	PPV	NPV
GM	0.96 (0.10, 0.94-0.98)	0.95 (0.13, 0.92-0.98)	0.82 (0.15, 0.79-0.85)
T1	0.87 (0.18, 0.83-0.91)	0.86 (0.19, 0.82-0.90)	0.84 (0.16, 0.81-0.87)
MD	0.95 (0.13, 0.92-0.97)	0.95 (0.13, 0.92-0.97)	0.88 (0.15, 0.85-0.91)
FA	0.97 (0.11, 0.95-0.99)	0.97 (0.10, 0.95-0.99)	0.91 (0.13, 0.88-0.94)
rs-fMRI PCC	*0.97 (0.09, 0.95-0.99)*	*0.97 (0.08, 0.96-0.99)*	*0.96 (0.09, 0.95-0.98)*
rs-fMRI PreCun	0.93 (0.14, 0.90-0.96)	0.93 (0.13, 0.91-0.96)	0.92 (0.13, 0.89-0.94)

Table 5.1: *CNN for Coma Classification - Model Performance.* Mean (SD, 95% CI) achieved by each evaluation metric obtained with the CNN trained with the corresponding MRI index. The best scores are highlighted in italic. CI: Confidence Interval; CNN: Convolutional Neural Network; FA: Fractional Anisotropy; GM: Gray Matter volume; MD: Mean Diffusivity; MR: Magnetic Resonance; MRI: Magnetic Resonance Imaging; NPV: Negative Predictive Value; PCC: Posterior Cingulate Cortex; PPV: Positive Predictive Value; PreCun: Precuneus; rs-fMRI: resting-state functional MRI; SD: Standard Deviation; T1: T1-weighted MRI. Adapted from [228]

In general, functional indices outperformed structural indices. We obtained overall satis-

fying performances independently of the MR index (accuracy over 0.80). The best accuracy of 0.96 was achieved by the rs-fMRI PCC index, whereas the T1 index obtained the worst accuracy amounting to 0.82. AUC scores were comparable to accuracy.
Specificity was high across indices (> 0.85). By contrast, sensitivity varied according to the MR index: especially low for GM and T1, about 0.75, but higher than 0.80 for the remaining indices. NPV scores were poorer than PPV.

5.3.3 Classification Errors

We provide CNN predictions for all subjects along with the majority voting in Fig. 5.2. Controls were all correctly classified using majority voting. The latter increased the sensitivity, resulting in a smaller number of FN compared to most MR indices.

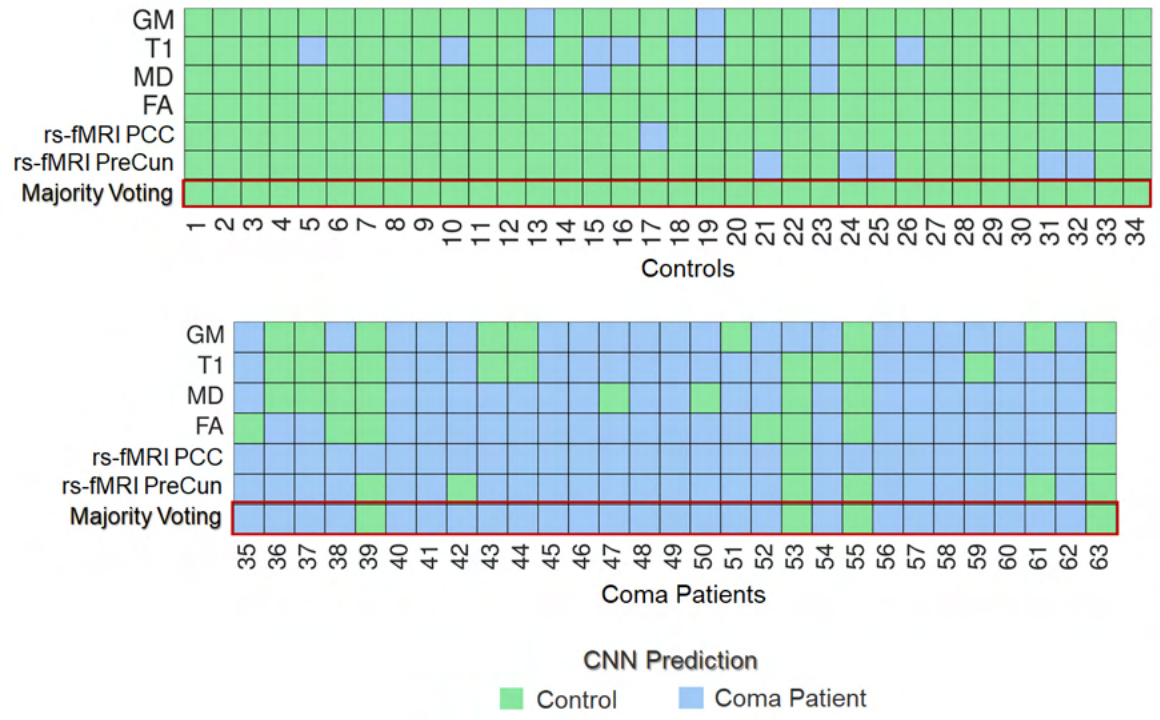


Figure 5.2: *CNN for Coma Classification - Classification Errors.* CNN prediction output reported for each control and coma patient. We applied the majority voting strategy (considering as final prediction the most frequent CNN output) to compensate for single MR index errors. Controls were all correctly classified and the performance on coma patients improved with only four misclassified (only second to rs-fMRI PCC with two misclassified). CNN: Convolutional Neural Network; FA: Fractional Anisotropy; GM: Gray Matter volume; MD: Mean Diffusivity; MR: Magnetic Resonance; MRI: Magnetic Resonance Imaging; PCC: Posterior Cingulate Cortex; PreCun: Precuneus; rs-fMRI: resting-state functional MRI; SD: Standard Deviation; T1: T1-weighted MRI. Adapted from [228]

Table 5.2 presents the number of FN and the FN good outcome rate per MR index.

MR Index	FN Good Outcome Rate (%)	Good Outcome FN	Total FN
GM	44%	4	9
T1	64%	7	11
MD	56%	5	9
FA	50%	3	6
rs-fMRI PCC	*100%*	*2*	*2*
rs-fMRI PreCun	67%	4	6

Table 5.2: *CNN for Coma Classification - Classification Errors.* Details about misclassified patients (FN) or each MRI index. We associated each FN with the corresponding outcome at three months after the primary severe brain injury to discover whether we could find a relationship between patients who recovered from coma and controls. The best results are highlighted in italic. CNN: Convolutional Neural Network; FA: Fractional Anisotropy; GM: Gray Matter volume; MD: Mean Diffusivity; MR: Magnetic Resonance; MRI: Magnetic Resonance Imaging; PCC: Posterior Cingulate Cortex; PreCun: Precuneus; rs-fMRI: resting-state functional MRI; SD: Standard Deviation; T1: T1-weighted MRI. Adapted from [228]

T1, GM, and MD indices achieved the highest number of FN (around ten) compared to six from FA and rs-fMRI PreCun and only two for rs-fMRI PCC.
To explore whether there could be a relationship between the CNN's prediction and the patient's neurological outcome at three months, we found the percentage of FN presenting a favorable outcome. We hypothesized that patients with a positive outcome may have presented functional or structural similarities with healthy subjects at the moment of the MRI acquisition, thus somehow foreseeing their survival. Our results showed that around 50% of FN turned out to have a favorable outcome for all indices except rs-fMRI PCC with 100% (just two FN all with a favorable outcome).

5.3.4 Visual Interpretation

As a representative example, Fig. 5.3 provides the average maps obtained from the training set considering a single repetition. We can notice that each MR index highlights different brain areas featured with high activation values. For instance, we can recognize the brainstem and subcortical cerebral structures in FA maps or associative cortical regions (e.g. mPFC) in functional indices.

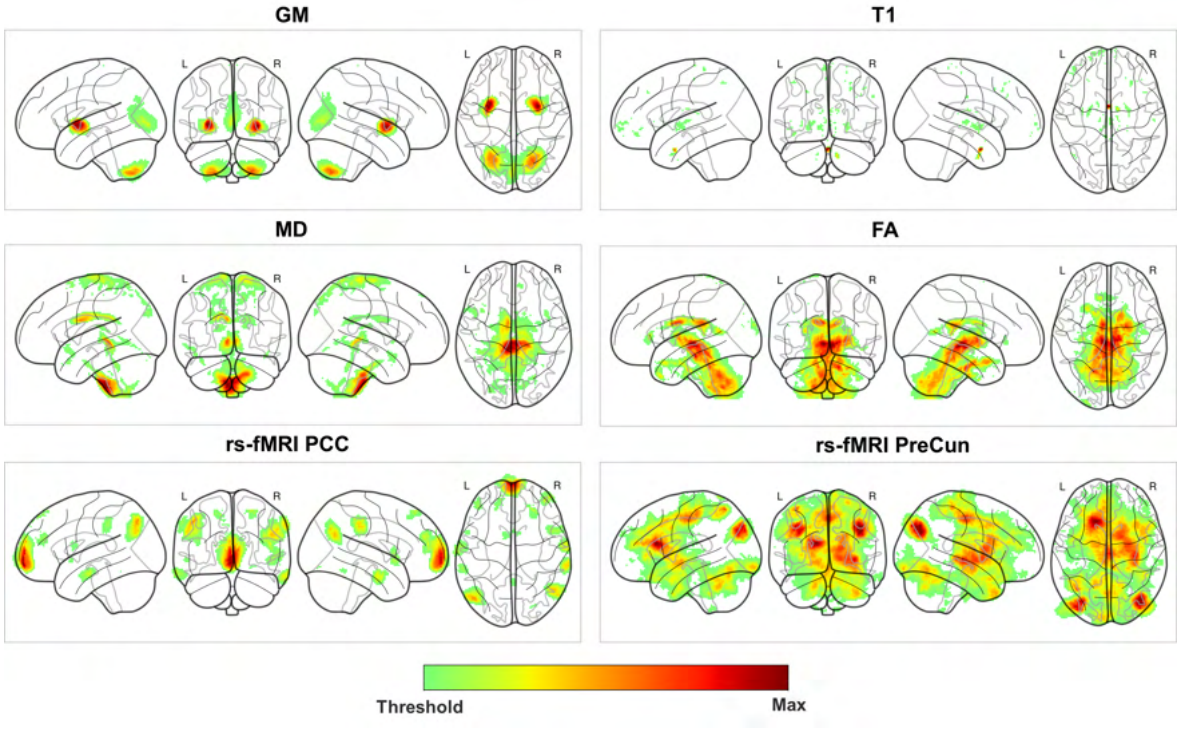


Figure 5.3: *CNN for Coma Classification - Visual Interpretation.* Visualization maps for each MR index showing the absolute difference between the average of correctly classified samples per class on the training set. To highlight salient parts, we applied a threshold equal to the maps at half the maximum value (Max). CNN: Convolutional Neural Network; FA: Fractional Anisotropy; GM: Gray Matter volume; L: Left; MD: Mean Diffusivity; MR: Magnetic Resonance; MRI: Magnetic Resonance Imaging; PCC: Posterior Cingulate Cortex; PreCun: Precuneus; R: Right; rs-fMRI: resting-state functional MRI; SD: Standard Deviation; T1: T1-weighted MRI. Adapted from [228]

5.4 Discussion

In this study, we proposed using a 3D CNN to discriminate between healthy subjects and patients in coma based on different MRI indices. To the best of our knowledge, this work represents one of the first attempts to apply deep learning to the detection of weak signals from 3D MRI data belonging to comatose patients with encouraging results. AI-assisted techniques can aid the analysis of the variegated information from MRI (both structural and functional) to offer an efficient and timely tool ready and easy to use for physicians taking care of patients in postanoxic coma.

Although performances were good for all MR indices, rs-fMRI PCC was the best scoring an accuracy of 0.96 on the test set from the 10-fold CV repeated ten times. Moreover, we found that mMRI can compensate for the errors from a single MRI index, proven by the majority voting strategy. Instead, structural indices exhibited overall poorer performances than functional ones. Recent works about the functional segregation within the posteromedial parietal cortex [242] and its role in conscious processing [243] pointed out that fMRI data from PCC appear less discriminating than PreCun. However, we still do not know to which extent acute severe brain injury affects functional or structural whole-brain connectivity, given that the relationships between these components have not yet been elucidated [244, 245].

As an investigatory objective, we searched for a connection between the falsely classified patients and their outcomes three months after the CA. Indeed, neuroprognostication via AI and neuroimaging data is one of the most promising research fields, given prior development and validation of methods to organize and merge information from raw MRI data. Our findings are in line with previous reports [233–236], electing rs-fMRI as a potential source of meaningful information able to predict neurological outcomes of patients in coma. Furthermore, in a previous study from our group, fMRI data seemed to be more useful for neuroprognostication than sMRI data [236]. Our analysis may pave the way for more sophisticated approaches to improve the prognosis of patients in coma.

Another aspect we covered was CNN's interpretability to cast some light on the predictions of these so-called black boxes. Analyzing the output from convolutional filters with a recently developed technique [107], we obtained for each MR index a map indicating the most activated voxels. Among the MR indices, FA maps revealed the activation of subcortical structures, whose damages include by consciousness abolition. The maps from rs-fMRI included the PCC as a region of interest which was coherent with previous studies reporting the potential role of frontoparietal disconnections as a coma biomarker [234–236].

Despite the limited number of samples, which burdens the medical field in general, we obtained encouraging results to consider CNNs a valuable discrimination tool exploiting spatial information from raw MRI data. In the future, we plan to increase our sample size to validate our method.

One strength of the proposed approach is the possibility to benefit from prediction errors to see whether there are factors contributing to a good prognosis for the patient, thus being able to anticipate it. In addition, we emphasized the added value of mMRI with the majority voting a posteriori. That is just the starting point to devise future developments for an automatic selection of the best discriminating MR indices based on a CNN.

Coma has been currently defined as a "disconnection syndrome" due to the combined damages provoked by primary and secondary severe brain insults [231, 238, 246–248]. Higher-order cognitive processes seem possible thanks to the information conveyed by multiple cerebral systems exploiting long-range functional interaction intrinsically related to brain structural connectivity [248, 249]. In light of this complexity, we hope our work will favor and enhance the use of deep learning methods to allow for knowledge discovery and patients' neuroprognostication improvement.

Conclusions and Future Work

This research project has enlightened us about crucial aspects which sometimes fade into the background, such as the importance of the information provided by training data to deep networks. Indeed, the beauty of a convolutional neural network lies in its ability to exploit data and learn their underpinning representations. However, if data are biased or incomplete, CNN performance cannot be optimal with consequent biases in interpretation.
For instance, think about a network that should distinguish nurses from doctors. When asked about the picture of a female doctor, the predicted class was instead a nurse. Why? Because training data comprised only female nurses and male doctors, hence the bias. We may infer that the network associated the nurse class with typical female characteristics (e.g. long hair) and the doctor class with male features (e.g. short hair, beard). On the contrary, we would have preferred it to associate traits that make us recognize a doctor, such as a white coat, or a nurse, such as a uniform. But if we do not provide meaningful information to the network, how can it possibly work well? That was just an emblematic example to give you an idea of what we are dealing with.

In this doctoral dissertation, the main objectives were to better understand CNN behavior and support the diagnosis of neurological disorders.
Always keeping in mind the importance of training content, we focused first on a pathology-agnostic approach by introducing targeted regional modifications to brain MRI data, thus creating the APMaps. We chose to alter mean diffusivity maps, a quantitative index informing about water diffusion, and used it as a biomarker for several diseases. Training the CNN with these calibrated altered images, we concretely showed its sensitivity to input features, such as the intensity and size of the altered regions. The most remarkable result was that the two regions, not discriminant alone, combined led to improved performances (accuracy from 0.65 to 0.90). To our knowledge, that represents the first attempt to better grasp CNN behavior via controlled modifications of the input data in the particular case of 3D brain MRI.
We could have made different choices, such as varying the CNN architecture or designing new methods for interpreting them. Instead, by focusing on the training data, we possessed the considerable advantage of knowing the differences between the two classes to facilitate interpretation.
As a perspective, we can envisage applying this method to different brain or body regions to characterize CNN performance according to the specific input features. It may help discover the peculiarities of detecting anomalies in specific regions and tune the network appropri-

ately. In general, we encourage trying to alter MRI data given a few prior considerations (e.g. the physical meaning of the MR sequence and what is expected from an abnormal condition) to create a baseline performance, which could be exploited as ground truth. That represents a considerable advantage in the case of many pathological conditions whose ground truth is unknown or incomplete.
These findings constituted the basis for moving to the more complex analysis of multiple system atrophy data. MSA is a rare neurodegenerative disease, presenting in the early stage with similar clinical symptoms but a more rapid progression than Parkinson's disease. Therefore, an accurate and early diagnosis is paramount for choosing the most suitable patient care path.

Rare diseases imply a paucity of data that could have prevented us from using CNNs, known as data-hungry methods. Fortunately, we can find increasingly continuous efforts by the scientific community to demonstrate the validity of deep networks with restricted sample sizes. Our work on MSA classification was devoted to making our contribution in this area, always keeping in mind the need for better CNN interpretability.

Full of enthusiasm for the results obtained with the APMaps, we exploited these pathology-agnostic region-specific APMaps for detecting similar traits in MSA patients. That required some a priori knowledge about the regions of interest in the disease. The cerebellum and putamen undergoing disease-related changes in MSA were the regions modified in the APMaps. However, at this stage, we did not conceive the alteration patterns of the APMaps to resemble the MSA patterns. Even so, by training a CNN to detect general and more uniform region-specific MD increases, we reached an accuracy of 0.88 on the set of MSA patients and HC. Furthermore, the population of healthy subjects used to create the APMaps and the pathological cohort were completely independent.
Despite this incredibly unexpected result, we felt we could still improve the creation of altered data by including features directly extracted from the disease. So we got back to modifying data with region-specific traits based on MSA features by creating the CB-APMaps. To do so, we clustered MSA patients according to the distribution of MD values in the cerebellum and transformed the same region in the healthy subjects to match this intensity distribution. Our best accuracy was 0.84, again a promising result. Nevertheless, this approach was still limited because it did not encompass whole-brain alterations.
That is why we devised a whole-brain approach to creating the ZB-APMaps. Using brain MRI data from healthy subjects made it feasible to increase our sample size by accessing online and in-house databases. We did not just test the case with a comparable number of samples between APMaps and MSA patients' data, but we tried to increase the representation of each pattern by amplifying it, i.e. considering more than one ZB-APMap per pattern.

Although the best performance achieved an accuracy of 0.88, we acknowledged that the ZB-APMaps were not as informative as the original pathological data (accuracy = 0.92).
Despite these encouraging performances, further effort is needed to understand how the different variants of APMaps impact CNN performance and why some work better than others. Indeed, it represents an open line of future research. For instance, applying this method to other diseases could inform us about its generalization ability and dependence on the specific features of the pathology.
These approaches allowed us to face the heterogeneity of patterns characterizing MSA pathology. So we took a step back and examined how the CNN reacted to different degrees of severity determined by clustering MSA patients based on whole-brain changes. This approach showed us that milder alterations seem to detect more severe alterations, unlike the opposite. That is extremely important considering that the early stages of a disease can be very challenging to discriminate from the healthy condition, hence the difficulty in correctly classifying them. Nevertheless, it is paramount to have some information about disease progression (e.g. if the alteration becomes more prominent in a single region or extends to others) because it would change the interpretation of CNN results or even determine the validity of the proposed method.
In response to the small amount of data for MSA, we proposed to study CNN behavior by varying the number of MSA patients given as input. This work showed that even with only ten patients, we could achieve satisfying performances in line with the reference performance using 20 MSA patients in training (accuracy = 0.90 vs. 0.92). Even though we considered various repetitions with different patients and HI, we found the performances did not vary much, as proven by the standard deviation decreasing with an increasing number of patients in training. These results are promising enough to encourage experimenting with CNNs, even if the sample size is limited. Performing similar experiments with other pathological data could increase our understanding of the importance and impact of training content on CNN performance while testing different architectures.
Last but not least is the possibility of finding hints for CNN pattern retrieval by analyzing misclassified patients. If these patients present features in common, we could identify traits that seem less relevant to the network's decision-making process. Hence, the misclassified patients could offer a starting point for ameliorating CNN performance by targeting their peculiarities.
By summarizing all the previous studies, a concrete application would be to test the feasibility of Parkinson's disease data by targeting the substantia nigra, which is a region of interest in PD. Given the small dimension of this region, we could first create pathology-agnostic APMaps, similarly as in Chapter 3, to establish a threshold for the intensity increase to ob-

tain good CNN performances. Afterward, we could exploit these maps to train a CNN and test PD patients' data. In addition, creating pathology-oriented maps may favor a better understanding of the influence of different patterns on CNN performance. Finally, progressively increasing the number of PD patients in training could lead to determining the minimal quantity of data necessary to achieve the classification task.
Furthermore, we plan to extend these studies to aid the differential diagnosis between PD, MSA, and other parkinsonian syndromes. It would be interesting to access different cohorts of pathological data, acquired for clinical and research purposes to discover whether similar behavior could be found.

The implications of creating altered brain MRI data via such a controlled approach able to provide meaningful content in the case of a rare neurodegenerative disease hold a great deal of promise, especially in deep learning applications. The paucity of data should not prevent us from getting the most from deep methods but rather prompt us to turn this to our advantage by exploiting the available wealth of knowledge. We hope our approach will inspire other scientists to investigate the importance of training data for inspecting CNN behavior.

Thanks to the applicability of CNN, we used them to distinguish coma patients from healthy controls with a multimodal approach. That represented a stepping stone to contribute to the field of neuroprognostication.
The main concern of clinicians regarding patients in a comatose state is the difficulty of predicting a patient's future. Relentless efforts are in progress to exploit all the information from neuroimaging and clinical findings and create an integrative and reliable approach to guide us in this complex world. Our findings are just the first bright step leading in this direction.

This work has amazed us with encouraging findings and comforted us when confirming our hypotheses. However, it has also raised awareness that there is room for improvement to favor the acceptance and use of these powerful tools.
The ultimate goal is always the patient's well-being to ensure a reliable and early diagnosis and a better quality of life. This noble aim requires the collaboration of health practitioners, researchers, and engineers to integrate their skills and knowledge and find an efficient way to merge their expertise. This transition will probably cause a shift in the role of radiologists, whose value and contribution will benefit from these advancements while maintaining their relevance.

After all, the beautiful machines we call black boxes are a human's work, and, as all human's work, they jealously guard their mysterious side.

List of Abbreviations

AD Alzheimer's Disease

ADC Apparent Diffusion Coefficient

ADNI Alzheimer's Disease Neuroimaging Initiative

AGI Artificial General Intelligence

AI Artificial Intelligence

ANN Artificial Neural Network

ANOVA Analysis Of Variance

APMaps Altered Parametric Maps

BN Batch Normalization

BOLD Blood-Oxygen-Level-Dependent

CA Cardiac Arrest

CAM Class Activation Mapping

CB-APMaps Cluster-Based Altered Parametric Maps

CNN Convolutional Neural Network

CRS-R Coma Recovery Scale Revised

CSF Cerebrospinal Fluid

CT Computed Tomography

CV Cross Validation

DCE-MRI Dynamic Contrast-Enhanced Magnetic Resonance Imaging

DICOM Digital Imaging and COmmunications in Medicine

DL Deep Learning

D-Putamen Dilated Putamen

DTI Diffusion Tensor Imaging

DWI Diffusion-Weighted Imaging

E-Cerebellum Eroded Cerebellum

ELU Exponential Linear Unit

ES Expert System

FA Fractional Anisotropy

FCL Fully Connected Layer

FID Free Induction Decay

fMRI functional Magnetic Resonance Imaging

FN False Negative

FP False Positive

GAN Generative Adversarial Network

GCS Glasgow Coma Scale

GM Gray Matter

GOFAI Good Old-Fashioned Artificial Intelligence

GPU Graphical Processing Unit

HC Healthy Controls

HI Healthy Individuals

HLM Hierarchical Learning Machine

HRF Hemodynamic Response Function

IEEE Institute of Electrical and Electronics Engineers

ILSVRC ImageNet Large Scale Visual Recognition Challenge

IQR Interquartile Range

ISBI International Symposium on Biomedical Imaging

KB Knowledge Base

LIME Local Interpretable Model-agnostic Explanations

MCS Minimally Conscious State

MD Mean Diffusivity

ML Machine Learning

MLP Multilayer Perceptron

mMRI multimodal Magnetic Resonance Imaging

MNI Montreal Neurological Institute

mPFC mesial PreFrontal Cortex

MR Magnetic Resonance

MRI Magnetic Resonance Imaging

MSA Multiple System Atrophy

MSA-C MSA Cerebellar variant

MSA-P MSA Parkinsonian variant

MSE Mean Squared Error

NifTi Neuroimaging informatics Technology initiative

NPV Negative Predictive Value

OPMaps Original Parametric Maps

PCC Posterior Cingulate Cortex

PD Parkinson's Disease

PET Positron Emission Tomography

PPMI Parkinson's Progression Markers Initiative

PPV Positive Predictive Value

PreCun PreCuneus

PSP Progressive Supranuclear Palsy

ReLU Rectified Linear Unit

RF Radio Frequency

RNN Recurrent Neural Network

ROI Region Of Interest

rs-fMRI resting-state functional Magnetic Resonance Imaging

SAR Specific Absorption Rate

SGD Stochastic Gradient Descent

sMRI structural Magnetic Resonance Imaging

SNc Substantia Nigra pars compacta

SPM Statistical Parametric Mapping

SVM Support Vector Machine

TE Echo Time

TN True Negative

TP True Positive

TR Repetition Time

VGG Visual Geometry Group

VS/UWS Vegetative State/Unresponsive Wakefulness Syndrome

WHO World Health Organization

WM White Matter

ZB-APMaps Z-score-Based Altered Parametric Maps

Bibliography

[1] J. W. Henson and R. G. Gonzalez, *Neuroimaging*, vol. 104. Elsevier B.V., 1 2012.

[2] E. George, J. P. Guenette, and T. C. Lee, *Introduction to Neuroimaging*, vol. 131. Elsevier Inc., 4 2018.

[3] E. M. Larsson and J. Wikström, *Overview of neuroradiology*, vol. 145. Elsevier B.V., 1 2018.

[4] P. Gantet and I. Berry, *Magnétostatique et RMN Cours et QCM UE3 PACES*. 2019.

[5] D. C. Preston, "Magnetic resonance imaging (mri) of the brain and spine: Basics." `https://case.edu/med/neurology/NR/MRI%20Basics.htm`, 2006. Accessed: 2022-07-17.

[6] M. Bergamino, L. Bonzano, F. Levrero, G. L. Mancardi, and L. Roccatagliata, "A review of technical aspects of t1-weighted dynamic contrast-enhanced magnetic resonance imaging (dce-mri) in human brain tumors," *Physica Medica*, vol. 30, pp. 635–643, 9 2014.

[7] D. Bihan, "Looking into the functional architecture of the brain with diffusion mri," *Nature Reviews Neuroscience*, vol. 4, pp. 469–480, 2003.

[8] G. S. Chilla, C. H. Tan, C. Xu, and C. L. Poh, "Diffusion weighted magnetic resonance imaging and its recent trend-a survey.," *Quantitative imaging in medicine and surgery*, vol. 5, pp. 407–22, 6 2015.

[9] C. Beaulieu and P. S. Allen, "Determinants of anisotropic water diffusion in nerves," *Magnetic Resonance in Medicine*, vol. 31, pp. 394–400, 4 1994.

[10] Y. Assaf and O. Pasternak, *Diffusion tensor imaging (DTI)-based white matter mapping in brain research: A review*, vol. 34. Springer, 1 2008.

[11] M. Viallon *et al.*, *State-of-the-art MRI techniques in neuroradiology: principles, pitfalls, and clinical applications*, vol. 57. Springer Verlag, 5 2015.

[12] P. Eustache, F. Nemmi, L. Saint-Aubert, J. Pariente, and P. Péran, "Multimodal magnetic resonance imaging in Alzheimer's disease patients at prodromal stage," *Journal of Alzheimer's Disease*, vol. 50, pp. 1035–1050, 2016.

[13] P. Péran *et al.*, "Magnetic resonance imaging markers of Parkinson's disease nigrostriatal signature," *Brain*, vol. 133, pp. 3423–3433, 2010.

[14] P. Péran *et al.*, "Mri supervised and unsupervised classification of Parkinson's disease and multiple system atrophy," *Movement Disorders*, vol. 33, pp. 600–608, 4 2018.

[15] E. M. Hillman, "Coupling mechanism and significance of the bold signal: A status report," *Annual Review of Neuroscience*, vol. 37, pp. 161–181, 7 2014.

[16] G. H. Glover, "Overview of functional magnetic resonance imaging," *Neurosurgery Clinics of North America*, vol. 22, pp. 133–139, 4 2011.

[17] E. Amaro and G. J. Barker, "Study design in fmri: Basic principles," *Brain and Cognition*, vol. 60, no. 3, pp. 220–232, 2006.

[18] T. Yousaf, G. Dervenoulas, and M. Politis, *Advances in MRI Methodology*, vol. 141. Academic Press Inc., 1 2018.

[19] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[20] Y. LeCun, *Quand la machine apprend*. Odile Jacobs, 7381-4931-x ed., 10 2019.

[21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[22] T. J. Huang, "Imitating the brain with neurocomputer a "new" way towards artificial general intelligence," *International Journal of Automation and Computing 2017 14:5*, vol. 14, pp. 520–531, 5 2017.

[23] L. J. Gugerty, "Newell and Simon's logic theorist: Historical background and impact on cognitive modeling," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 50, pp. 880 – 884, 2006.

[24] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65 6, pp. 386–408, 1958.

[25] M. Minsky and S. A. Papert, *Perceptrons: An Introduction to Computational Geometry*. The MIT Press, 09 2017.

[26] K. Fukushima, "Cognitron: A self-organizing multilayered neural network," *Biological Cybernetics 1975 20:3*, vol. 20, pp. 121–136, 9 1975.

[27] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics 1980 36:4*, vol. 36, pp. 193–202, 4 1980.

[28] D. Hubel and T. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of Physiology*, vol. 160, pp. 106–54, 1962.

[29] D. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[30] Y. L. Cun, "Learning process in an asymmetric threshold network," *Disordered Systems and Biological Organization*, pp. 233–240, 1986.

[31] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning 1995 20:3*, vol. 20, pp. 273–297, 1995.

[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "Imagenet: A large-scale hierarchical image database," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, pp. 84–90, 5 2017.

[34] P. Jackson, *Introduction to Expert Systems*. USA: Addison-Wesley Longman Publishing Co., Inc., 3rd ed., 1998.

[35] J. S. Zielinski and S. D. J. McArthur, "An introduction to intelligent knowledge based systems," *Intelligent knowledge based systems in electrical power engineering*, pp. 5–12, 1997.

[36] "Advantages and disadvantages of expert systems." `https://www.ilearnlot.com/expert-system-advantages-disadvantages/34332/`. Accessed: 2022-05-29.

[37] K. Mainzer, *Systems Become Experts*. Springer, Berlin, Heidelberg, 2020.

[38] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter, *Probabilistic Networks and Expert Systems*. Springer-Verlag, 1999.

[39] T. M. Mitchell, *Machine Learning*. McGraw-Hill, 1997.

[40] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, pp. 179–188, 9 1936.

[41] F. Nemmi *et al.*, "A totally data-driven whole-brain multimodal pipeline for the discrimination of Parkinson's disease, multiple system atrophy and healthy control," *NeuroImage : Clinical*, vol. 23, 2019.

[42] L. Baecker, R. Garcia-Dias, S. Vieira, C. Scarpazza, and A. Mechelli, "Machine learning for brain age prediction: Introduction to methods and clinical applications," *EBioMedicine*, vol. 72, 10 2021.

[43] S. P. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, 1982.

[44] B. Everitt, *The Cambridge dictionary of statistics*. Cambridge University Press, 2006.

[45] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 11 1987.

[46] S. Theodoridis and K. Koutroumbas, "Pattern recognition," *Pattern Recognition*, 2009.

[47] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, pp. 67–82, 1997.

[48] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics 1943 5:4*, vol. 5, pp. 115–133, 12 1943.

[49] D. O. Hebb, *The Organization of Behavior: A Neuropsycological Theory*. Wiley, 1949.

[50] I. Goodfellow *et al.*, "Generative adversarial nets," in *Advances in Neural Information Processing Systems* (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, eds.), vol. 27, Curran Associates, Inc., 2014.

[51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 12 2015.

[52] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2016.

[53] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[54] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, pp. 267–288, 1 1996.

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

[56] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. abs/1207.0580, 2012.

[57] D. Silver *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, pp. 354–359, 10 2017.

[58] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.

[59] H. E. Kim, A. Cosa-Linan, N. Santhanam, M. Jannesari, M. E. Maros, and T. Ganslandt, "Transfer learning for medical image classification: a literature review," *BMC Medical Imaging*, vol. 22, 12 2022.

[60] M. A. Morid, A. Borjali, and G. D. Fiol, "A scoping review of transfer learning research on medical image analysis using imagenet," *Computers in Biology and Medicine*, vol. 128, 1 2021.

[61] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International Conference on Artificial Intelligence and Statistics*, 2010.

[62] A. Cauchy, "Méthode générale pour la résolution de systèmes d'équations simultanées," *Compte rendu des séances de l'académie des sciences*, vol. 83, pp. 536–538, 1847.

[63] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[64] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, pp. 1–17, 1 1964.

[65] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, 7 2011.

[66] G. Hinton, "Neural networks for machine learning.," *Coursera, video lectures.*, 2012.

[67] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.

[68] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of Statistical Planning and Inference*, vol. 90, pp. 227–244, 10 2000.

[69] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *ArXiv*, vol. abs/1502.03167, 2015.

[70] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural Networks: Tricks of the Trade*, 1998.

[71] C. Szegedy *et al.*, "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2014.

[72] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, 1998.

[73] J. R. Vincent de Ladurantaye and J. Vanden-Abeele, "Hierarchical model of vision." `https://commons.wikimedia.org/wiki/File:Hierarchical_Model_of_Vision.jpg`. Source: `https://www.intechopen.com/books/visual-cortex-current-status-and-perspectives/models-of-information-processing-in-the-visual-cortex`, Sep 6, 2019. Licence: CC BY-SA 3.0 `https://creativecommons.org/licenses/by-sa/3.0`, via Wikimedia Commons.

[74] M. W. Eysenck and M. T. Keane, *Cognitive psychology : a student's handbook*. New York, NY : Psychology Press, 6 ed., 2010.

[75] OpenStax College, "1424 visual streams." https://commons.wikimedia.org/wiki/File:1424_Visual_Streams.jpg. Source: Anatomy & Physiology, Connexions Web site. http://cnx.org/content/col11496/1.6/, Jun 19, 2013. Licence: CC BY 3.0 https://creativecommons.org/licenses/by/3.0, via Wikimedia Commons.

[76] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *ArXiv*, vol. abs/1311.2901, 2013.

[77] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Computing Research Repository*, vol. abs/1409.1556, 2015.

[78] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, pp. 157–166, 1994.

[79] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller, "Striving for simplicity: The all convolutional net," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.

[80] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Los Alamitos, CA, USA), pp. 3431–3440, IEEE Computer Society, jun 2015.

[81] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," vol. 9351, pp. 234–241, Springer Verlag, 2015.

[82] A. Holzinger, C. Biemann, C. S. Pattichis, and D. B. Kell, "What do we need to build explainable ai systems for the medical domain?," *ArXiv*, vol. abs/1712.09923, 2017.

[83] Y. Bengio, A. C. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 1798–1828, 2012.

[84] R. Bellman, R. Corporation, and K. M. R. Collection, *Dynamic Programming*. Rand Corporation research study, Princeton University Press, 1957.

[85] R. Bellman and K. M. R. Collection, *Adaptive Control Processes: A Guided Tour*. Princeton Legacy Library, Princeton University Press, 1961.

[86] M. G. Core, H. C. Lane, M. van Lent, D. Gomboc, S. Solomon, and M. Rosenberg, "Building explainable artificial intelligence systems," in *AAAI Conference on Artificial Intelligence*, 2006.

[87] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje, "Not just a black box: Learning important features through propagating activation differences," *ArXiv*, vol. abs/1605.01713, 2016.

[88] D. C. Elton, "Self-explaining ai as an alternative to interpretable ai," pp. 95–106, Springer International Publishing, 2020.

[89] T. P. Lillicrap and K. P. Kording, "What does it mean to understand a neural network?," *ArXiv*, vol. abs/1907.06374, 2019.

[90] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital Signal Processing*, vol. 73, pp. 1–15, 2018.

[91] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 80–89, 2018.

[92] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *CoRR*, vol. abs/1312.6034, 2014.

[93] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why should i trust you?": Explaining the predictions of any classifier," in *KDD '16*, 2016.

[94] J. R. Zilke, E. L. Mencía, and F. Janssen, "Deepred - rule extraction from deep neural networks," in *Discovery Science*, 2016.

[95] R. Andrews, J. Diederich, and A. B. Tickle, "Survey and critique of techniques for extracting rules from trained artificial neural networks," *Knowl.-Based Syst.*, vol. 8, pp. 373–389, 1995.

[96] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2921–2929, 2015.

[97] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2016.

[98] D. Smilkov, N. Thorat, B. Kim, F. B. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," *ArXiv*, vol. abs/1706.03825, 2017.

[99] S. Bach *et al.*, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," in *PloS one*, 2015.

[100] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 3319–3328, JMLR.org, 2017.

[101] P.-J. Kindermans *et al.*, "The (un)reliability of saliency methods," in *Explainable AI*, 2018.

[102] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller, "How to explain individual classification decisions," *J. Mach. Learn. Res.*, vol. 11, pp. 1803–1831, 2009.

[103] G. Montavon, S. Bach, A. Binder, W. Samek, and K.-R. Müller, "Explaining nonlinear classification decisions with deep taylor decomposition," *Pattern Recognition*, vol. 65, pp. 211–222, 2017.

[104] P.-J. Kindermans *et al.*, "Learning how to explain neural networks: Patternnet and patternattribution," in *ICLR*, 2017.

[105] S. Haufe *et al.*, "On the interpretation of weight vectors of linear models in multivariate neuroimaging," *NeuroImage*, vol. 87, pp. 96–110, 2014.

[106] E. Villain, *Utilisation de l'intelligence artificielle pour l'aide au diagnostic des patients atteints de pathologies neuro dégénératives*. PhD thesis, Université Toulouse III Paul Sabatier, 12 2021.

[107] E. Villain, G. M. Mattia, F. Nemmi, P. Péran, X. Franceries, and M. V. L. Lann, "Visual interpretation of cnn decision-making process using simulated brain mri," in *2021 IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS)*, pp. 515–520, 2021.

[108] "Dicom." `https://www.dicomstandard.org/`. Accessed: 2022-09-23.

[109] "Nifti: - neuroimaging informatics technology initiative." `https://nifti.nimh.nih.gov/`. Accessed: 2022-09-23.

[110] E. B. Johnson *et al.*, "Recommendations for the use of automated gray matter segmentation tools: Evidence from huntington's disease," *Frontiers in Neurology*, vol. 8, p. 519, 10 2017.

[111] X. Chen, H. Zhang, L. Zhang, C. Shen, S. W. Lee, and D. Shen, "Extraction of dynamic functional connectivity from brain grey matter and white matter for mci classification," *Human Brain Mapping*, vol. 38, pp. 5019–5034, 10 2017.

[112] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C. W. Lin, "Deep learning on image denoising: An overview," *Neural Networks*, vol. 131, pp. 251–275, 11 2020.

[113] C. Pierpaoli, "Quantitative brain mri," *Topics in Magnetic Resonance Imaging*, vol. 21, p. 63, 4 2010.

[114] X. Liu, K. Chen, T. Wu, D. Weidman, F. Lure, and J. Li, "Use of multimodality imaging and artificial intelligence for diagnosis and prognosis of early stages of Alzheimer's disease," *Translational Research*, vol. 194, pp. 56–67, 4 2018.

[115] K. R. Laukamp *et al.*, "Fully automated detection and segmentation of meningiomas using deep learning on routine multiparametric mri," *European Radiology*, vol. 29, pp. 124–132, 1 2019.

[116] K. Pinker, T. H. Helbich, and E. A. Morris, "The potential of multiparametric mri of the breast," *British Journal of Radiology*, vol. 90, 2017.

[117] V. Sawlani *et al.*, "Multiparametric mri: practical approach and pictorial review of a useful tool in the evaluation of brain tumours and tumour-like lesions," *Insights into Imaging*, vol. 11, 12 2020.

[118] S. Ghafoor, I. A. Burger, and A. H. Vargas, "Multimodality imaging of prostate cancer," *Journal of Nuclear Medicine*, vol. 60, pp. 1350–1358, 10 2019.

[119] G. Litjens *et al.*, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, 2017.

[120] G. Currie, K. E. Hawk, E. Rohren, A. Vial, and R. Klein, "Machine learning and deep learning in medical imaging: Intelligent imaging," *Journal of Medical Imaging and Radiation Sciences*, vol. 50, pp. 477–487, 12 2019.

[121] J. C. Gore, "Artificial intelligence in medical imaging," *Magnetic Resonance Imaging*, vol. 68, pp. A1–A4, 5 2020.

[122] J. H. Cole *et al.*, "Predicting brain age with deep learning from raw imaging data results in a reliable and heritable biomarker," *NeuroImage*, vol. 163, pp. 115–124, 12 2017.

[123] C. Davatzikos, "Why voxel-based morphometric analysis should be used with great caution when characterizing group differences," *NeuroImage*, vol. 23, pp. 17–20, 2004.

[124] K. J. Friston, A. P. Holmes, K. J. Worsley, J.-P. Poline, C. D. Frith, and R. S. J. Frackowiak, "Statistical parametric maps in functional imaging: A general linear approach," *Human Brain Mapping*, vol. 2, pp. 189–210, 1 1994.

[125] C. Davatzikos, "Machine learning in neuroimaging: Progress and challenges," *NeuroImage*, vol. 197, pp. 652–656, 8 2019.

[126] S. Haller, K. O. Lovblad, P. Giannakopoulos, and D. V. D. Ville, "Multivariate pattern recognition for diagnosis and prognosis in clinical neuroimaging: State of the art, current challenges and future trends," *Brain Topography*, vol. 27, pp. 329–337, 3 2014.

[127] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 10 2001.

[128] A. Sarica, A. Cerasa, and A. Quattrone, "Random forest algorithm for the classification of neuroimaging data in Alzheimer's disease: A systematic review," *Frontiers in Aging Neuroscience*, vol. 9, 10 2017.

[129] E. Feczko *et al.*, "Subtyping cognitive profiles in autism spectrum disorder using a functional random forest algorithm," *NeuroImage*, vol. 172, pp. 674–688, 5 2018.

[130] X. A. Bi, Q. Shu, Q. Sun, and Q. Xu, "Random support vector machine cluster analysis of resting-state fmri in Alzheimer's disease," *PLoS ONE*, vol. 13, 3 2018.

[131] L. Squarcina *et al.*, "Automatic classification of autism spectrum disorder in children using cortical thickness and support vector machine," *Brain and Behavior*, vol. 11, 8 2021.

[132] M. A. Mazurowski, M. Buda, A. Saha, and M. R. Bashir, "Deep learning in radiology: An overview of the concepts and a survey of the state of the art with focus on mri," *Journal of Magnetic Resonance Imaging*, vol. 49, pp. 939–954, 4 2019.

[133] G. Zaharchuk, E. Gong, M. Wintermark, D. Rubin, and C. P. Langlotz, "Deep learning in neuroradiology," *American Journal of Neuroradiology*, vol. 39, pp. 1776–1784, 10 2018.

[134] G. Martí-Juan, G. Sanroma-Guell, and G. Piella, "A survey on machine and statistical learning for longitudinal analysis of neuroimaging data in Alzheimer's disease," *Computer Methods and Programs in Biomedicine*, vol. 189, 6 2020.

[135] M. R. Ahmed, Y. Zhang, Z. Feng, B. Lo, O. T. Inan, and H. Liao, "Neuroimaging and machine learning for dementia diagnosis: Recent advancements and future prospects," *IEEE Reviews in Biomedical Engineering*, vol. 12, pp. 19–33, 2019.

[136] S. S. Mohseni Salehi, D. Erdogmus, and A. Gholipour, "Auto-context convolutional neural network (auto-net) for brain extraction in magnetic resonance imaging," *IEEE Transactions on Medical Imaging*, vol. 36, no. 11, pp. 2319–2330, 2017.

[137] H. Chen, Q. Dou, L. Yu, J. Qin, and P. A. Heng, "Voxresnet: Deep voxelwise residual networks for brain segmentation from 3d mr images," *NeuroImage*, vol. 170, pp. 446–455, 4 2018.

[138] P. Moeskops, M. Veta, M. W. Lafarge, K. A. Eppenhof, and J. P. Pluim, "Adversarial training and dilated convolutions for brain mri segmentation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10553 LNCS, pp. 56–64, 2017.

[139] P. Natekar, A. Kori, and G. Krishnamurthi, "Demystifying brain tumour segmentation networks: Interpretability and uncertainty analysis," *ArXiv*, vol. abs/1909.01498, 2019.

[140] S. Lee, J. Lee, J. Lee, C.-K. Park, and S. Yoon, "Robust tumor localization with pyramid grad-cam," *ArXiv*, vol. abs/1805.11393, 2018.

[141] R. Gaurav *et al.*, "Deep learning-based neuromelanin mri changes of isolated rem sleep behavior disorder," *Movement Disorders*, vol. 37, pp. 1064–1069, 5 2022.

[142] Z. Akkus, A. Galimzianova, A. Hoogi, D. L. Rubin, and B. J. Erickson, "Deep learning for brain mri segmentation: State of the art and future directions," *Journal of Digital Imaging*, vol. 30, pp. 449–459, 8 2017.

[143] K. Shal and M. S. Choudhry, "Evolution of deep learning algorithms for mri-based brain tumor image segmentation," *Critical Reviews in Biomedical Engineering*, vol. 49, pp. 77–94, 2021.

[144] Q. Dou *et al.*, "Automatic detection of cerebral microbleeds from mr images via 3d convolutional neural networks," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1182–1195, 2016.

[145] A. Panda, R. Naskar, S. Rajbans, and S. Pal, "A 3d wide residual network with perceptual loss for brain mri image denoising," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–7, 2019.

[146] Z. Zhou *et al.*, "Super-resolution of brain tumor mri images based on deep learning," *Journal of Applied Clinical Medical Physics*, p. e13758, 9 2022.

[147] A. Chartsias, T. Joyce, M. V. Giuffrida, and S. A. Tsaftaris, "Multimodal mr synthesis via modality-invariant latent representation," *IEEE Transactions on Medical Imaging*, vol. 37, no. 3, pp. 803–814, 2018.

[148] C. Han *et al.*, "Gan-based synthetic brain mr image generation," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pp. 734–738, 2018.

[149] X. Han, "Mr-based synthetic ct generation using a deep convolutional neural network method," *Medical Physics*, vol. 44, pp. 1408–1419, 4 2017.

[150] S. M. Plis *et al.*, "Deep learning for neuroimaging: a validation study," *Frontiers in Neuroscience*, vol. 8, 2014.

[151] S. M. G. Vieira, W. H. L. Pinaya, and A. Mechelli, "Using deep learning to investigate the neuroimaging correlates of psychiatric and neurological disorders: Methods and applications," *Neuroscience Biobehavioral Reviews*, vol. 74, pp. 58–75, 2017.

[152] J. Wen *et al.*, "Convolutional neural networks for classification of Alzheimer's disease: Overview and reproducible evaluation," *Medical Image Analysis*, vol. 63, p. 101694, 2020.

[153] H. W. Loh *et al.*, "Application of deep learning models for automated identification of Parkinson's disease: A review (2011-2021)," *Sensors*, vol. 21, 11 2021.

[154] M. Khosla, K. Jamison, A. Kuceyeski, and M. R. Sabuncu, "Ensemble learning with 3d convolutional neural networks for functional connectome-based prediction," *NeuroImage*, vol. 199, pp. 651–662, 2019.

[155] S. Esmaeilzadeh, Y. Yang, and E. Adeli, "End-to-end Parkinson disease diagnosis using brain mr-images by 3d-cnn," *ArXiv*, vol. abs/1806.05233, 2018.

[156] S. Korolev, A. Safiullin, M. Belyaev, and Y. Dodonova, "Residual and plain convolutional neural networks for 3d brain mri classification," *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pp. 835–838, 2017.

[157] L. Yuan, X. Wei, H. Shen, L.-L. Zeng, and D. Hu, "Multi-center brain imaging classification using a novel 3d cnn approach," *IEEE Access*, vol. 6, pp. 49925–49934, 2018.

[158] E. Trivizakis *et al.*, "Extending 2-d convolutional neural networks to 3-d for advancing deep learning cancer classification with application to mri liver tumor differentiation," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, pp. 923–930, 5 2019.

[159] S. P. Singh, L. Wang, S. Gupta, H. Goli, P. Padmanabhan, and B. Gulyàs, "3d deep learning on medical images: A review," *Sensors (Switzerland)*, vol. 20, pp. 1–24, 9 2020.

[160] L. Wang, Y. Wang, and Q. Chang, "Feature selection methods for big data bioinformatics: A survey from the search perspective," *Methods*, vol. 111, pp. 21–31, 12 2016.

[161] L. Brigato and L. Iocchi, "A close look at deep learning with small data," in *2020 25th International Conference on Pattern Recognition (ICPR)*, (Los Alamitos, CA, USA), pp. 2490–2497, IEEE Computer Society, 1 2021.

[162] M. Olson, A. Wyner, and R. Berk, "Modern neural networks generalize on small data sets," in *Advances in Neural Information Processing Systems* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), vol. 31, Curran Associates, Inc., 2018.

[163] T. Wang *et al.*, "A review on medical imaging synthesis using deep learning and its clinical applications," *Journal of applied clinical medical physics*, vol. 22, pp. 11–36, 1 2021.

[164] G. Kwon, C. Han, and D. shik Kim, "Generation of 3d brain mri using auto-encoding generative adversarial networks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11766 LNCS, pp. 118–126, 2019.

[165] D. Nie *et al.*, "Medical image synthesis with deep convolutional adversarial networks hhs public access," *IEEE Trans Biomed Eng*, vol. 65, pp. 2720–2730, 2018.

[166] K. Kazuhiro *et al.*, "Generative adversarial networks for the creation of realistic artificial brain magnetic resonance images," *Tomography*, vol. 4, pp. 159 – 163, 2018.

[167] M. E. Laino, P. Cancian, L. S. Politi, M. G. D. Porta, L. Saba, and V. Savevski, "Generative adversarial networks in brain imaging: A narrative review," *Journal of Imaging*, vol. 8, 2022.

[168] G. Barbagallo *et al.*, "Multimodal mri assessment of nigro-striatal pathway in multiple system atrophy and Parkinson disease," *Movement Disorders*, vol. 31, no. 3, pp. 325–34, 2016.

[169] B. A. Richards *et al.*, "A deep learning framework for neuroscience," *Nature Reviews Neuroscience*, vol. 22, pp. 1761–1770, 2019.

[170] J. Brettschneider, K. Tredici, V. Lee, and J. Trojanowski, "Spreading of pathology in neurodegenerative diseases: A focus on human studies," *Nature Reviews Neuroscience*, vol. 16, pp. 109–120, 2015.

[171] S. Vos, D. Jones, B. Jeurissen, M. Viergever, and A. Leemans, "The influence of complex white matter architecture on the mean diffusivity in diffusion tensor mri of the human brain," *NeuroImage*, vol. 59, pp. 2208–2216, 2012.

[172] H. Kim, S. Kim, H. S. Kim, C. Choi, and C. Lee, "Alterations of mean diffusivity in brain white matter and deep gray matter in Parkinson's disease," *Neuroscience Letters*, vol. 550, pp. 64–68, 2013.

[173] D. Yin, F. Valles, M. Fiandaca, J. Forsayeth, and K. Bankiewicz, "Striatal volume differences between non-human and human primates," *Journal of Neuroscience Methods*, vol. 176, pp. 200–205, 2009.

[174] G. Shepherd, *The Synaptic Organization of the Brain*, ch. 7. New York: Oxford University Press., 2004.

[175] M. Molinari and M. Leggio, *Cerebellum: Clinical Pathology*, pp. 737–742. Elsevier Ltd, 2010.

[176] N. Viñas-Guasch and Y. J. Wu, "The role of the putamen in language: a meta-analytic connectivity modeling study," *Brain Structure and Function*, vol. 222, pp. 3991–4004, 2017.

[177] S. Haber, "Corticostriatal circuitry," *Dialogues in Clinical Neuroscience*, vol. 18, pp. 7–21, 2016.

[178] D. Berg, J. Steinberger, C. W. Olanow, T. Naidich, and T. Yousry, "Milestones in magnetic resonance imaging and transcranial sonography of movement disorders.," *Movement Disorders*, vol. 26, no. 6, pp. 979–92, 2011.

[179] H. Shin, S. Kang, J. H. Yang, H. Kim, M.-S. Lee, and Y. Sohn, "Use of the putamen/-caudate volume ratio for early differentiation between Parkinsonian variant of multiple system atrophy and Parkinson disease," *Journal of Clinical Neurology (Seoul, Korea)*, vol. 3, no. 2, pp. 79–81, 2007.

[180] K. Seppi *et al.*, "Progression of putaminal degeneration in multiple system atrophy: A serial diffusion mr study," *NeuroImage*, vol. 31, pp. 240–245, 2006.

[181] A. Michell, S. Lewis, T. Foltynie, and R. Barker, "Biomarkers and Parkinson's disease," *Brain*, vol. 127 Pt 8, pp. 1693–705, 2004.

[182] G. M. Mattia, F. Nemmi, E. Villain, M. V. L. Lann, X. Franceries, and P. Péran, "Investigating the discrimination ability of 3d convolutional neural networks applied to altered brain mri parametric maps," *TechRxiv. Preprint*, 2021.

[183] F. Nemmi, M. Levardon, and P. Péran, "Brain-age estimation accuracy is significantly increased using multishell free-water reconstruction," *Human Brain Mapping*, vol. 43, pp. 2365–2376, 5 2022.

[184] T. Behrens *et al.*, "Characterization and propagation of uncertainty in diffusion-weighted mr imaging," *Magnetic Resonance in Medicine*, vol. 50, no. 5, pp. 1077–88, 2003.

[185] M. Brett, I. S. Johnsrude, and A. M. Owen, "The problem of functional localization in the human brain," *Nature Reviews Neuroscience*, vol. 3, pp. 243–249, 2002.

[186] A. Hammers *et al.*, "Three-dimensional maximum probability atlas of the human brain, with particular reference to the temporal lobe," *Human Brain Mapping*, vol. 19, no. 4, pp. 224–47, 2003.

[187] J. Wen *et al.*, "Convolutional neural networks for classification of Alzheimer's disease: Overview and reproducible evaluation," *Medical Image Analysis*, vol. 63, p. 101694, 2020.

[188] M. I. Qureshi, J. Oh, and B. Lee, "3d-cnn based discrimination of schizophrenia using resting-state fmri," *Artificial Intelligence in Medicine*, vol. 98, pp. 10–17, 2019.

[189] F. Chollet *et al.*, "Keras." `https://keras.io`, 2015.

[190] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2016.

[191] J. Brettschneider, K. Tredici, V. Lee, and J. Trojanowski, "Spreading of pathology in neurodegenerative diseases: A focus on human studies," *Nature Reviews Neuroscience*, vol. 16, pp. 109–120, 2015.

[192] A. Schrag and R. N. Taddei, *Depression and Anxiety in Parkinson's Disease*, vol. 133. Academic Press Inc., 1 2017.

[193] L. M. Chahine, A. W. Amara, and A. Videnovic, "A systematic review of the literature on disorders of sleep and wakefulness in Parkinson's disease from 2005 to 2015," *Sleep Medicine Reviews*, vol. 35, pp. 33–50, 10 2017.

[194] J. Parkinson, "An essay on the shaking palsy. 1817.," *The Journal of neuropsychiatry and clinical neurosciences*, vol. 14, 3 2002.

[195] L. Chougar, N. Pyatigorskaya, and S. Lehéricy, "Update on neuroimaging for categorization of Parkinson's disease and atypical Parkinsonism," *Current Opinion in Neurology*, vol. 34, pp. 514–524, 8 2021.

[196] T. Mitchell, S. Lehéricy, S. Y. Chiu, A. P. Strafella, A. J. Stoessl, and D. E. Vaillancourt, "Emerging neuroimaging biomarkers across disease stage in Parkinson disease: A review," *JAMA Neurology*, vol. 78, pp. 1262–1272, 10 2021.

[197] "Parkinson disease." `https://www.who.int/news-room/fact-sheets/detail/{Parkinson}-disease`. Accessed: 2022-07-29.

[198] A. Samii, J. G. Nutt, and B. R. Ransom, "Parkinson's disease," vol. 363, pp. 1783–1793, Elsevier B.V., 5 2004.

[199] L. Marsili, G. Rizzo, and C. Colosimo, "Diagnostic criteria for Parkinson's disease: From james Parkinson to the concept of prodromal disease," *Frontiers in Neurology*, vol. 9, 3 2018.

[200] N. R. Mcfarland, "Diagnostic approach to atypical Parkinsonian syndromes," *CONTINUUM Lifelong Learning in Neurology*, vol. 22, pp. 1117–1142, 8 2016.

[201] J. Richardson, J. Steele, and J. Olszewski, "Supranuclear ophthalmoplegia, pseudobulbar palsy, nuchal dystonia and dementia. a clinical report on eight cases of heterogenous system degeneration.," *Trans Am Neurol Assoc*, vol. 88, pp. 25–29, 1963.

[202] I. G. McKeith *et al.*, "Diagnosis and management of dementia with lewy bodies: Third report of the dlb consortium," *Neurology*, vol. 65, pp. 1863–1872, 2005.

[203] L. Chougar *et al.*, "Automated categorization of Parkinsonian syndromes using magnetic resonance imaging in a clinical setting," *Movement Disorders*, vol. 36, pp. 460–470, 2 2021.

[204] D. B. Archer *et al.*, "Development and validation of the automated imaging differentiation in Parkinsonism (aid-p): a multicentre machine learning study," *The Lancet Digital Health*, vol. 1, pp. e222–e231, 9 2019.

[205] A. Cherubini *et al.*, "Magnetic resonance support vector machine discriminates between Parkinson disease and progressive supranuclear palsy," *Movement Disorders*, vol. 29, pp. 266–269, 2 2014.

[206] H. Huppertz *et al.*, "Differentiation of neurodegenerative Parkinsonian syndromes by volumetric magnetic resonance imaging analysis and support vector machine classification," *Movement Disorders*, vol. 31, 2016.

[207] A. Schrag, Y. Ben-Shlomo, and N. P. Quinn, "Prevalence of progressive supranuclear palsy and multiple system atrophy: A cross-sectional study," *Lancet*, vol. 354, pp. 1771–1775, 11 1999.

[208] U. Wüllner, T. Schmitz-Hübsch, M. Abele, G. Antony, P. Bauer, and K. Eggert, "Features of probable multiple system atrophy patients identified among 4770 patients with Parkinsonism enrolled in the multicentre registry of the german competence network on parkinson's disease," *Journal of Neural Transmission*, vol. 114, pp. 1161–1165, 9 2007.

[209] Y. Ben-Shlomo, G. K. Wenning, F. Tison, and N. P. Quinn, "Survival of patients with pathologically proven multiple system atrophy: A meta-analysis," *Neurology*, vol. 48, pp. 384–393, 1997.

[210] A. Schrag, G. K. Wenning, N. Quinn, and Y. Ben-Shlomo, "Survival in multiple system atrophy," *Movement Disorders*, vol. 23, pp. 294–296, 1 2008.

[211] A. Fanciulli and G. K. Wenning, "Multiple-system atrophy," *New England Journal of Medicine*, vol. 372, pp. 249–263, 1 2015.

[212] H. J. Lee, D. Ricarte, D. Ortiz, and S. J. Lee, "Models of multiple system atrophy," *Experimental and Molecular Medicine*, vol. 51, pp. 1–10, 11 2019.

[213] G. K. Wenning *et al.*, "The movement disorder society criteria for the diagnosis of multiple system atrophy," *Movement Disorders*, vol. 37, pp. 1131–1148, 6 2022.

[214] F. Krismer *et al.*, "Morphometric mri profiles of multiple system atrophy variants and implications for differential diagnosis," *Movement Disorders*, vol. 34, pp. 1041–1048, 7 2019.

[215] L. Chougar, N. Pyatigorskaya, B. Degos, D. Grabli, and S. Lehéricy, "The role of magnetic resonance imaging for the diagnosis of atypical parkinsonism," *Frontiers in Neurology*, vol. 11, 7 2020.

[216] Q. Ren *et al.*, "Morphology and signal changes of the lentiform nucleus based on susceptibility weighted imaging in Parkinsonism-predominant multiple system atrophy," *Parkinsonism and Related Disorders*, vol. 81, pp. 194–199, 12 2020.

[217] M. Kim, J. H. Ahn, Y. Cho, J. S. Kim, J. Youn, and J. W. Cho, "Differential value of brain magnetic resonance imaging in multiple system atrophy cerebellar phenotype and spinocerebellar ataxias," *Scientific Reports*, vol. 9, pp. 1–7, 12 2019.

[218] G. Carré *et al.*, "Brain mri of multiple system atrophy of cerebellar type: a prospective study with implications for diagnosis criteria," *Journal of Neurology*, vol. 267, pp. 1269–1277, 5 2020.

[219] J. Pasquini, M. J. Firbank, R. Ceravolo, V. Silani, and N. Pavese, "Diffusion magnetic resonance imaging microstructural abnormalities in multiple system atrophy: A comprehensive review," *Movement Disorders*, 8 2022.

[220] C. Scherfler *et al.*, "Diagnostic potential of automated subcortical volume segmentation in atypical Parkinsonism," *Neurology*, vol. 86, pp. 1242–1249, 3 2016.

[221] M. Tsuda, S. Asano, Y. Kato, K. Murai, and M. Miyazaki, "Differential diagnosis of multiple system atrophy with predominant Parkinsonism and parkinson's disease using neural networks," *Journal of the Neurological Sciences*, vol. 401, pp. 19–26, 6 2019.

[222] Y. Kanatani, Y. Sato, S. Nemoto, M. Ichikawa, and O. Onodera, "Improving the accuracy of diagnosis for multiple-system atrophy using deep learning-based method," *Biology*, vol. 11, 7 2022.

[223] G. M. Mattia *et al.*, "Neurodegenerative traits detected via 3d cnns trained with simulated brain mri: Prediction supported by visualization of discriminant voxels," in

2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 1437–1442, 2021.

[224] J. R. England and P. Cheng, "Artificial intelligence for medical image analysis: A guide for authors and reviewers.," *AJR. American journal of roentgenology*, vol. 212 3, pp. 513–519, 2019.

[225] G. M. Mattia, E. Villain, O. Rascol, W. G. Meissner, X. Franceries, and P. Péran, "Multiple system atrophy classification via 3d convolutional neural network and simulated brain mri parametric maps," in *Proc. Intl. Soc. Mag. Reson. Med. 30*, 2022. https://index.mirasmart.com/ISMRM2022/PDFfiles/2675.html.

[226] R. C. Gonzalez and R. E. Woods, *Digital image processing*. Prentice Hall, third ed., 8 2007.

[227] W. G. Meissner *et al.*, "A phase 1 randomized trial of specific active α-synuclein immunotherapies pd01a and pd03a in multiple system atrophy," *Movement Disorders*, vol. 35, pp. 1957–1965, 11 2020.

[228] G. M. Mattia *et al.*, "Multimodal mri-based whole-brain assessment in patients in anoxoischemic coma by using 3d convolutional neural networks," *Neurocritical Care*, vol. 37, pp. 303–312, 8 2022.

[229] D. M. Greer, E. S. Rosenthal, and O. Wu, "Neuroprognostication of hypoxic-ischaemic coma in the therapeutic hypothermia era," *Nature Reviews Neurology*, vol. 10, pp. 190–203, 2014.

[230] C. Sandroni, A. Grippo, and J. P. Nolan, "Erc-esicm guidelines for prognostication after cardiac arrest: time for an update," *Intensive Care Medicine*, vol. 46, pp. 1901–1903, 10 2020.

[231] B. L. Edlow *et al.*, "Personalized connectome mapping to guide targeted therapy and promote recovery of consciousness in the intensive care unit," *Neurocritical Care 2020 33:2*, vol. 33, pp. 364–375, 8 2020.

[232] L. Velly *et al.*, "Use of brain diffusion tensor imaging for the prediction of long-term neurological outcomes in patients after cardiac arrest: a multicentre, international, prospective, observational, cohort study," *The Lancet Neurology*, vol. 17, pp. 317–326, 4 2018.

[233] S. Silva *et al.*, "Brain gray matter mri morphometry for neuroprognostication after cardiac arrest," *Critical care medicine*, vol. 45, pp. e763–e771, 8 2017.

[234] B. Malagurski *et al.*, "Neural signature of coma revealed by posteromedial cortex connection density analysis," *NeuroImage. Clinical*, vol. 15, pp. 315–324, 2017.

[235] B. Malagurski *et al.*, "Topological disintegration of resting state functional connectomes in coma," *NeuroImage*, vol. 195, pp. 354–361, 7 2019.

[236] P. Peran *et al.*, "Functional and structural integrity of frontoparietal connectivity in traumatic and anoxic coma," *Critical care medicine*, vol. 48, pp. E639–E647, 8 2020.

[237] S. Silva *et al.*, "Disruption of posteromedial large-scale neural communication predicts recovery from coma," *Neurology*, vol. 85, pp. 2036–2044, 12 2015.

[238] J. T. Giacino, J. J. Fins, S. Laureys, and N. D. Schiff, "Disorders of consciousness after acquired brain injury: the state of the science," *Nature reviews. Neurology*, vol. 10, pp. 99–114, 2 2014.

[239] B. L. Edlow, J. Claassen, N. D. Schiff, and D. M. Greer, "Recovery from disorders of consciousness: mechanisms, prognosis and emerging therapies," *Nature Reviews Neurology 2020 17:3*, vol. 17, pp. 135–156, 12 2020.

[240] L. Xu, A. Krzyżak, and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, pp. 418–435, 1992.

[241] J. A. Benediktsson, J. Chanussot, and M. Fauvel, "Multiple classifier systems in remote sensing: From basics to recent developments," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4472 LNCS, pp. 501–512, 2007.

[242] P. Fransson and G. Marrelec, "The precuneus/posterior cingulate cortex plays a pivotal role in the default mode network: Evidence from a partial correlation network analysis," *NeuroImage*, vol. 42, pp. 1178–1184, 9 2008.

[243] C. Koch, M. Massimini, M. Boly, and G. Tononi, "Posterior and anterior cortex - where is the difference that makes the difference?," *Nature Reviews Neuroscience 2016 17:10*, vol. 17, pp. 666–666, 7 2016.

[244] D. A. Gusnard and M. E. Raichle, "Searching for a baseline: Functional imaging and the resting human brain," *Nature Reviews Neuroscience*, vol. 2, pp. 685–694, 2001.

[245] C. D. Perri *et al.*, "Neural correlates of consciousness in patients who have emerged from a minimally conscious state: a cross-sectional multimodal imaging study," *The Lancet. Neurology*, vol. 15, pp. 830–842, 7 2016.

[246] C. M. Booth, R. H. Boone, G. Tomlinson, and A. S. Detsky, "Is this patient dead, vegetative, or severely neurologically impaired? assessing outcome for comatose survivors of cardiac arrest," *JAMA*, vol. 291, pp. 870–879, 2 2004.

[247] R. T. Seel *et al.*, "Assessment scales for disorders of consciousness: evidence-based recommendations for clinical practice and research," *Archives of physical medicine and rehabilitation*, vol. 91, pp. 1795–1813, 12 2010.

[248] S. Dehaene and J. P. Changeux, "Experimental and theoretical approaches to conscious processing," *Neuron*, vol. 70, pp. 200–227, 4 2011.

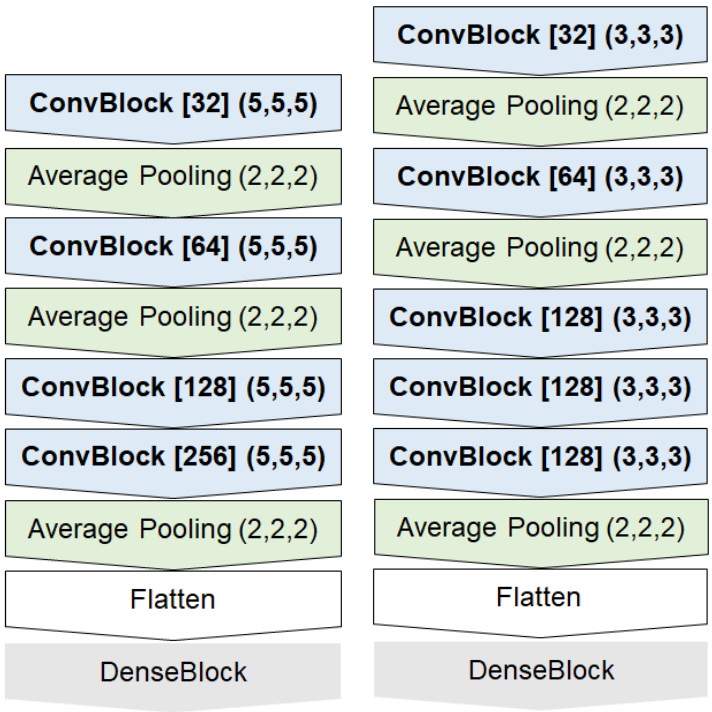
[249] S. Dehaene, J. P. Changeux, L. Naccache, J. Sackur, and C. Sergent, "Conscious, preconscious, and subliminal processing: a testable taxonomy," *Trends in Cognitive Sciences*, vol. 10, pp. 204–211, 5 2006.

Appendices

A APMaps for CNN Interpretability

A.1 Comparison with Previous Work

We illustrate the preliminary results regarding the comparison between the CNN model employed in a previous study from our group [106] and the one proposed in the current dissertation.
Fig. IV shows the two CNN architectures taken into consideration.



(a) Previous work [106] (b) Current work

Figure IV: Comparison between CNN architectures considering a previous work from our group [106]. The main differences are highlighted in bold. Building blocks are detailed in Section 3.2.4, Fig. 3.2.4b. Average Pooling: Average Pooling layer; CNN: Convolutional Neural Network; ConvBlock: Block performing the Convolution operation; DenseBlock: Block with Dense layers (fully connected layers); Flatten: reshaping of the outcome of the previous layers in a 1-D array

The main difference between the two models is in the convolutional layers, having filter size equal to 5×5×5 for the model proposed in [106] and 3×3×3 for the model proposed

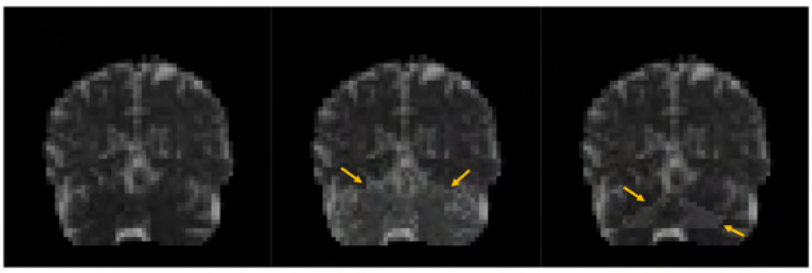
in the present dissertation. In addition, the latter presents three convolutional layers with different filter numbers instead of the two in the other architecture.
We trained the two networks with the implementation described in Section 3.2.4 performing a 10-fold CV.
Table III presents the accuracy scored on the test set achieved by the two models, considering monoregion APMaps with a 99% intensity increase. We can notice that the model adopted in this work outperformed the other by more than 10% in the case of the E-Cerebellum. One possible explanation could be related to the smaller filter size favoring the extraction of features within a smaller receptive field.

Table III: Comparison of CNN performances expressed as median accuracy (IQR) between a previous work from our group [106] and the work presented in this dissertation. Results were obtained by training the models with APMaps presenting an intensity increase equal to 99% for all regions. APMaps: Altered Parametric Maps; CNN: Convolutional Neural Network; D: Dilated; E: Eroded; IQR: Interquartile Range

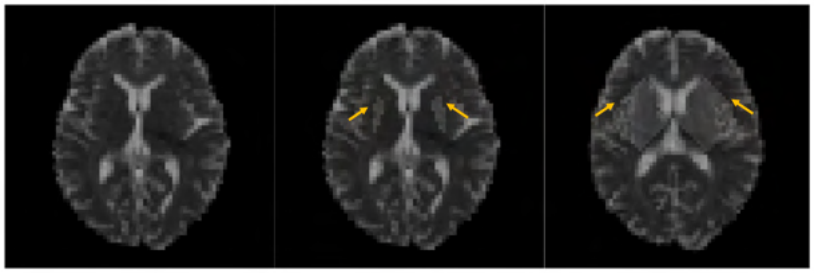
CNN Model	Target Region in CNN Training			
	Cerebellum	Putamen	D-Putamen	E-Cerebellum
Previous Work	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.69 (0.08)
Current Work	*1.00 (0.00)*	*1.00 (0.00)*	*1.00 (0.00)*	*0.81 (0.03)*

A.2 Monoregion-Trained CNNs

We provide some examples of APMaps in Fig. V.



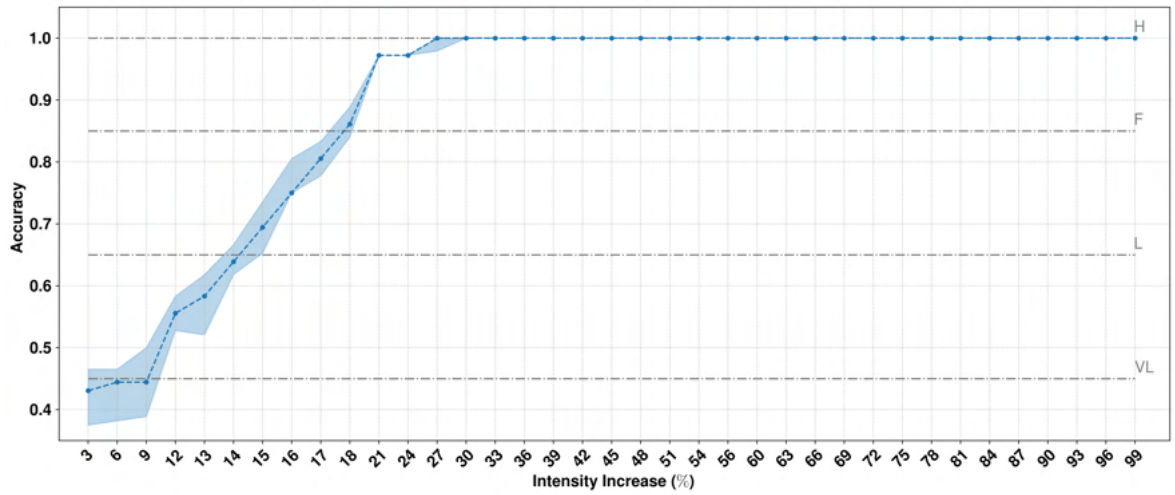
(a) OPMap, Cerebellum APMap, and E-Cerebellum APMap



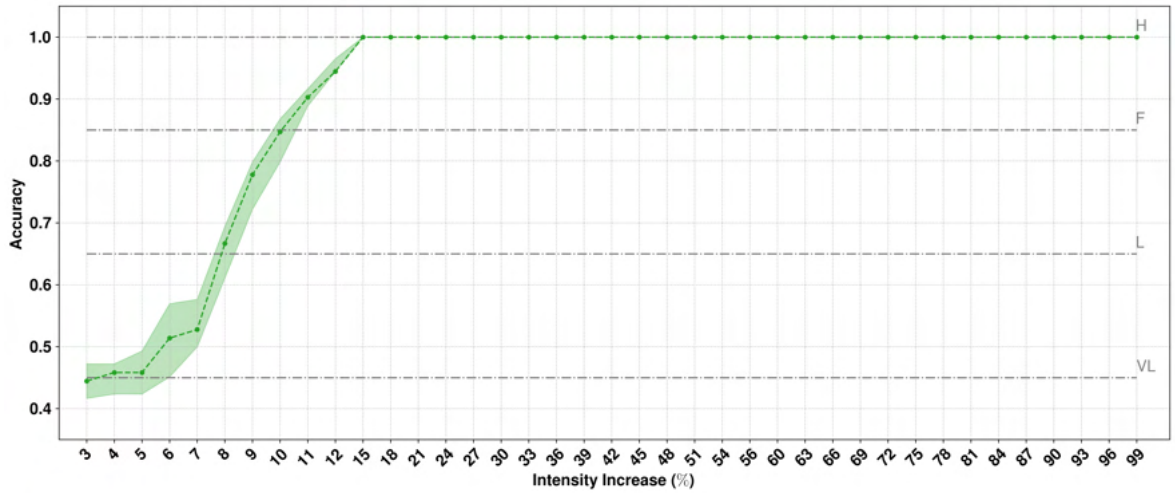
(b) OPMap, Putamen APMap, and D-Putamen APMap

Figure V: *Monoregion APMaps*. Examples of APMaps at 75% intensity increase. Arrows point to the modified regions. APMaps: Altered Parametric Maps; D: Dilated; E: Eroded; OPMaps: Original Parametric Maps. Reproduced from [182] (Giulia Maria Mattia, 2021). CC BY-NC-SA 4.0

To meet as precisely as possible the reference accuracy values, we considered additional intensity increases in steps of 1%, as provided in Fig. VI and VII.

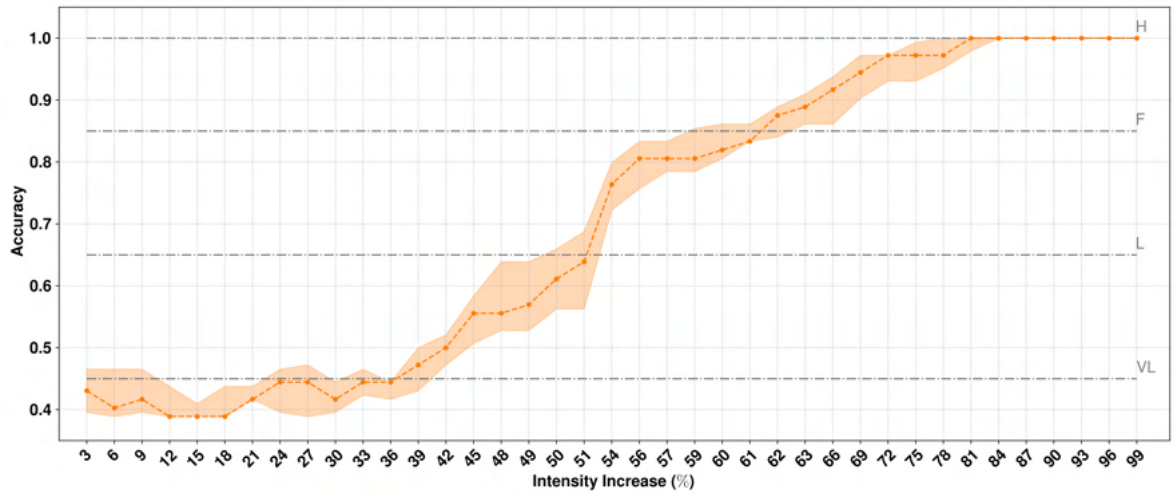


(a) Cerebellum

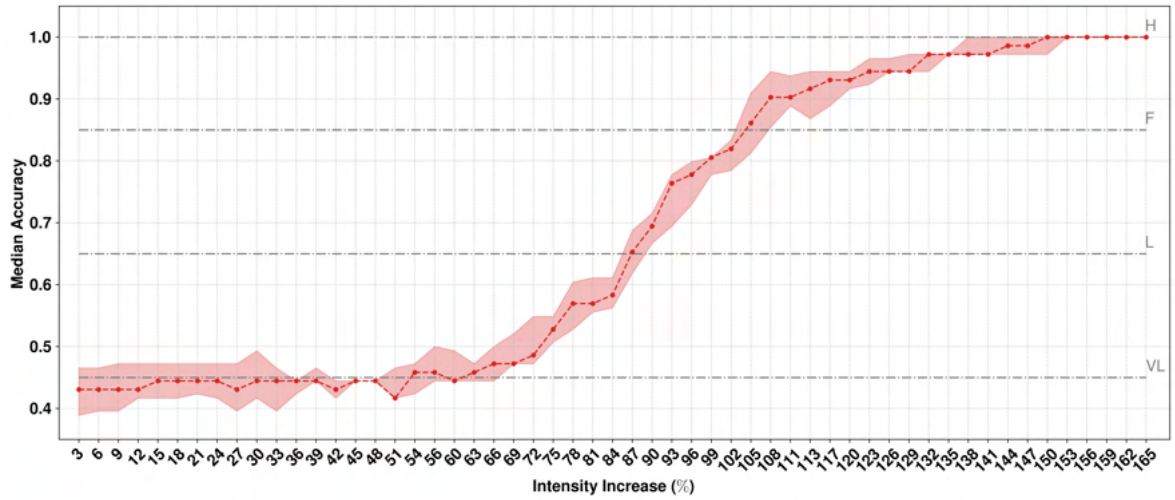


(b) D-Putamen

Figure VI: *Monoregion-Trained CNNs*. Accuracy on the hold-out set given as the median and IQR over a 10-fold CV according to intensity increase. APMaps: Altered Parametric Maps; CNN: Convolutional Neural Network; D: Dilated; F: Fair; H: High; IQR: Interquartile Range; L: Low; VL: Very Low. Reproduced from [182] (Giulia Maria Mattia, 2021). CC BY-NC-SA 4.0



(a) Putamen



(b) E-Cerebellum

Figure VII: *Monoregion-Trained CNNs*. Accuracy on the hold-out set given as the median and IQR over a 10-fold CV according to intensity increase. APMaps: Altered Parametric Maps; CNN: Convolutional Neural Network; CV: Cross Validation; D: Dilated; F: Fair; H: High; IQR: Interquartile Range; L: Low; VL: Very Low. Reproduced from [182] (Giulia Maria Mattia, 2021). CC BY-NC-SA 4.0

A.2.1 Comparison with other CNN Architectures

We proposed our implementation of two famous architectures, GoogLeNet and ResNet, for comparison to the model inspired by VGGNet. We named each model after the corresponding well-known CNN architecture.

Fig. VIII details the devised CNN structures. We provide the additional building blocks in Fig. 4.18, whereas the others are available in Fig. 3.7b.

We indicate the filter number of convolutional layers for GoogLeNet and ResNet in Table

4.8 (layers that are not indicated in Fig. VIII may be disregarded in Table 4.8).

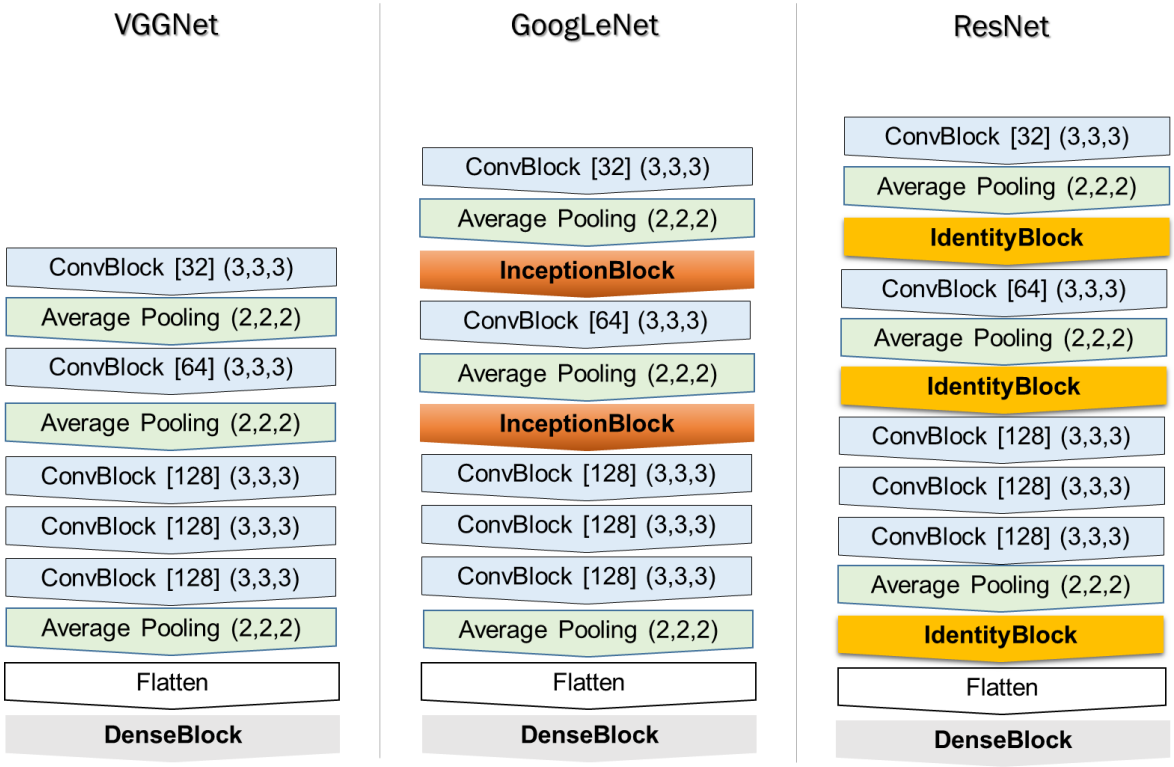
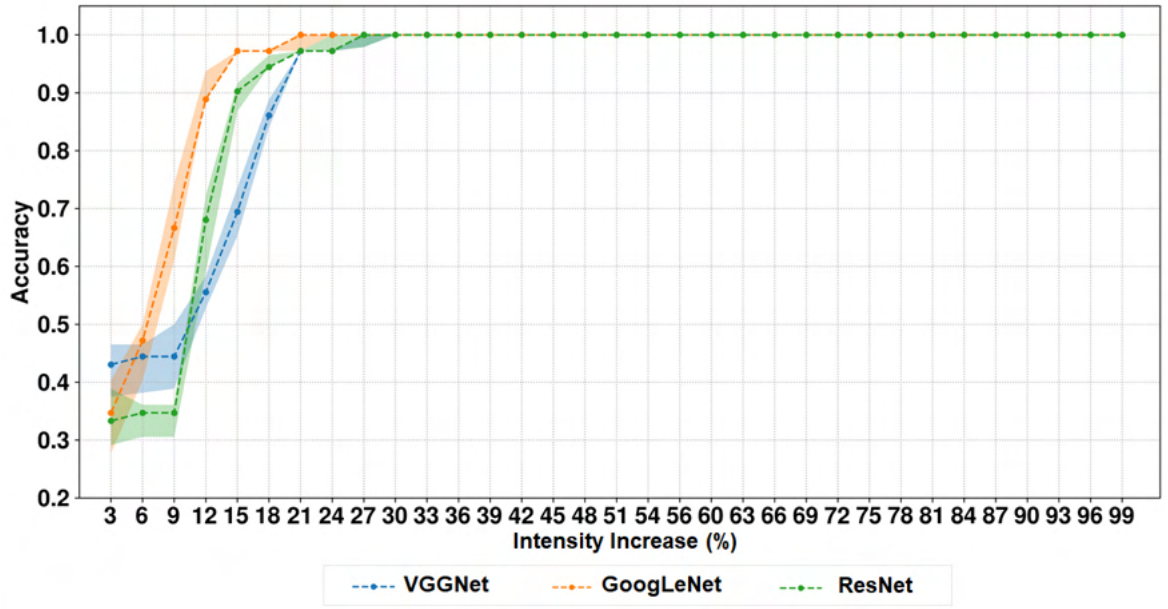
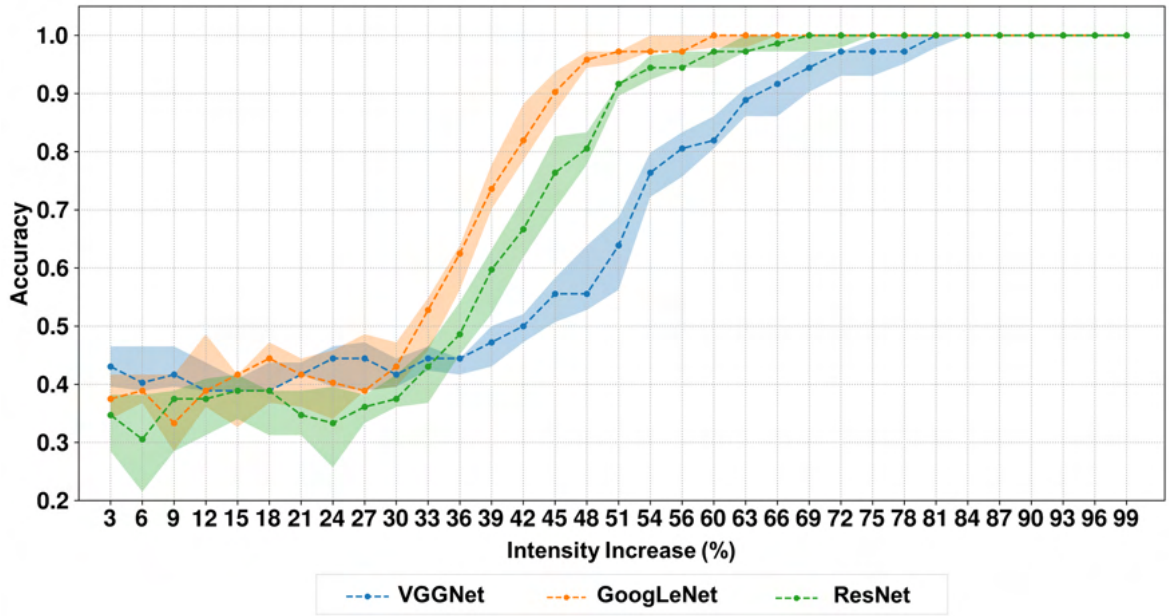


Figure VIII: Proposed CNN architectures named after the corresponding well-known model, considering as input images with a resolution equal to 3×3×3 mm^3 per voxel. Details about each building block are available in Fig. 3.2.4b. Average Pooling: Average Pooling layer; Conv3D: Convolutional layer: ConvBlock: Convolutional layer Block; CNN: Convolutional Neural Network; DenseBlock: block containing fully connected layers; ELU: Exponential Linear Unit; Flatten: operation to reshape in a one-dimensional vector; IdentityBlock: block characteristic of ResNet; InceptionBlock: block characteristic of GoogLeNet; Max Pooling: Max Pooling layer; [filter number]; (filter size)

For the two additional architectures, we kept the implementation described in Section 3.2.4, considering only Cerebellum and Putamen APMaps as input.
Fig. IX shows performance comparison. We can notice that all three CNNs showed comparable behavior, even with some differences in performance, with GoogLeNet being the most effective. That underlines that this type of investigation can be used to choose the most suitable model.



(a) Cerebellum



(b) Putamen

Figure IX: *Monoregion-Trained CNNs.* Accuracy on the hold-out set given as the median and IQR over a 10-fold CV according to intensity increase and CNN architecture. CNN: Convolutional Neural Network; CV: Cross Validation; IQR: Interquartile Range

A.2.2 APMaps and Training Set Size

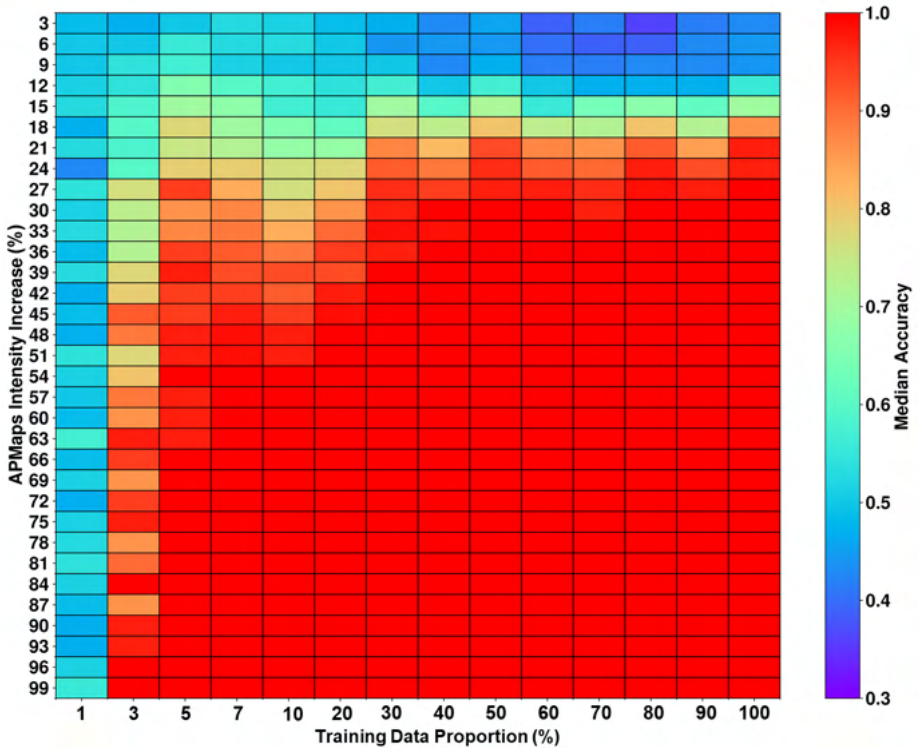
We explored CNN performance when varying the training set size according to the modified regions in the APMaps. In this case, we considered the cerebellum and putamen as regions of interest.

Adopting a 10-fold CV, we progressively increased the number of APMaps and OPMaps in training to reach the maximum possible quantity (i.e. 80% of the entire dataset reserved for training, see Section 3.2.4 for more details) and tested on the hold-out set.

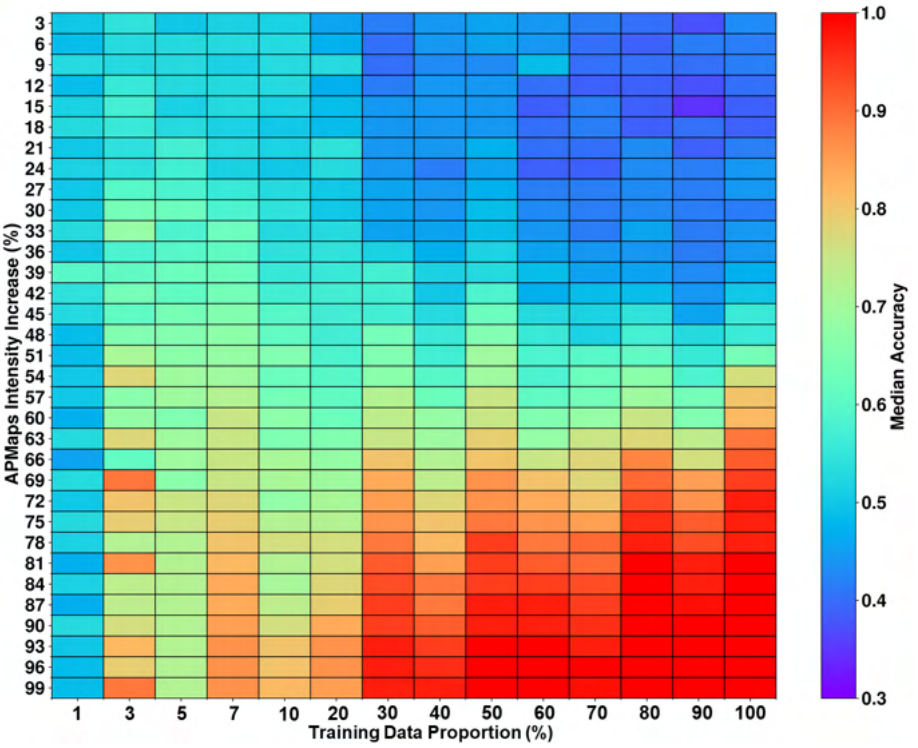
Fig. X provides the median accuracy over the ten folds according to training set size and intensity increase in the APMaps. There is a substantial difference between the Cerebellum and Putamen CNN. The former achieved high accuracy at a 27% intensity increase and only 5% of training data, whereas the latter needed over 80% intensity increase and 30% of training data.

From these findings, we can conclude that the greater and more intense the modified region, the less training data are necessary for good performances. That is in line with our expectations: as the differences between the classes to discern become more evident, learning gets easier for the network.

This implementation is simplified as APMaps were modified with a common method contrary to pathological data, which may present high inter-subject variability. Nevertheless, the proposed application highlighted how much the information delivered by training data could affect CNN performance.



(a) Cerebellum



(b) Putamen

Figure X: *Monoregion-Trained CNNs.* CNN performances on the hold-out set according to training set size and intensity increase, considering APMaps presenting alterations in the cerebellum or putamen. APMaps: Altered Parametric Maps; CNN: Convolutional Neural Network

A.3 Biregion-Trained CNNs

Table IV lists the significant combinations obtained from the post-hoc analysis of the one-way ANOVA. To perform this analysis, we considered the accuracy obtained by the different combinations of biregion-trained CNNs. To avoid repetitions, we grouped biregion CNNs presenting the same significant combinations, listed according to increasing accuracy levels from VL to H.

Table IV: *Biregion-trained CNNs.* Significant combinations given by a one-way ANOVA performed on the accuracy values achieved by biregion-trained CNNs. ANOVA: Analysis Of Variance; CNN: Convolutional Neural Network; D: Dilated; E: Eroded; F: Fair; H: High; L: Low; VL: Very Low. Reproduced from [182] (Giulia Maria Mattia, 2021). CC BY-NC-SA 4.0

Biregion-Trained CNN	**Significant Combinations**	
Cerebellum/Putamen, E-Cerebellum/Putamen, D-Putamen/Cerebellum	VL/VL*vs.*	VL/L *vs.*
	VL/L, VL/F, VL/H	VL/F, VL/H
	L/VL, L/L, L/F, L/H	L/L, L/F, L/H
	F/VL, F/L, F/F, F/H	F/VL, F/L, F/F, F/H
	H/VL, H/L, H/F, H/H	H/VL, H/L, H/F, H/H
	L/VL *vs.*	F/VL *vs.*
	VL/F, VL/H	VL/H
	L/L, L/F, L/H	L/H
	F/VL, F/L, F/F, F/H	F/F, F/H
	H/VL, H/L, H/F, H/H	H/VL, H/L, H/F, H/H
Cerebellum/Putamen, D-Putamen/Cerebellum	VL/F *vs.*	L/L *vs.*
	VL/H	VL/H
	L/F, L/H	L/H
	F/F, F/H	F/L, F/F, F/H
	H/VL, H/L, H/F, H/H	H/VL, H/L, H/F, H/H
Cerebellum/Putamen	L/F *vs.*	VL/F *vs.*
	VL/H	F/L
	L/H	
	F/H	
	H/VL, H/L, H/F, H/H	
D-Putamen/Cerebellum	L/F *vs.*	F/L *vs.*
		VL/H
	L/L	L/H
	F/VL	F/VL, F/H
		H/VL, H/L, H/F, H/H

A.4 Monoregion- vs. Biregion-Trained CNNs

Fig. XI presents biregion-trained CNN performance considering biregion APMaps with the two regions modified with the same intensity increase, compared to the results from the monoregion-trained CNN.
Cerebellum/Putamen CNN showed similar behavior to Cerebellum CNN. The same goes for the D-Putamen/Cerebellum CNN with respect to D-Putamen CNN. The performance of E-Cerebellum/Putamen CNNs slightly improved compared to their monoregion counterparts.

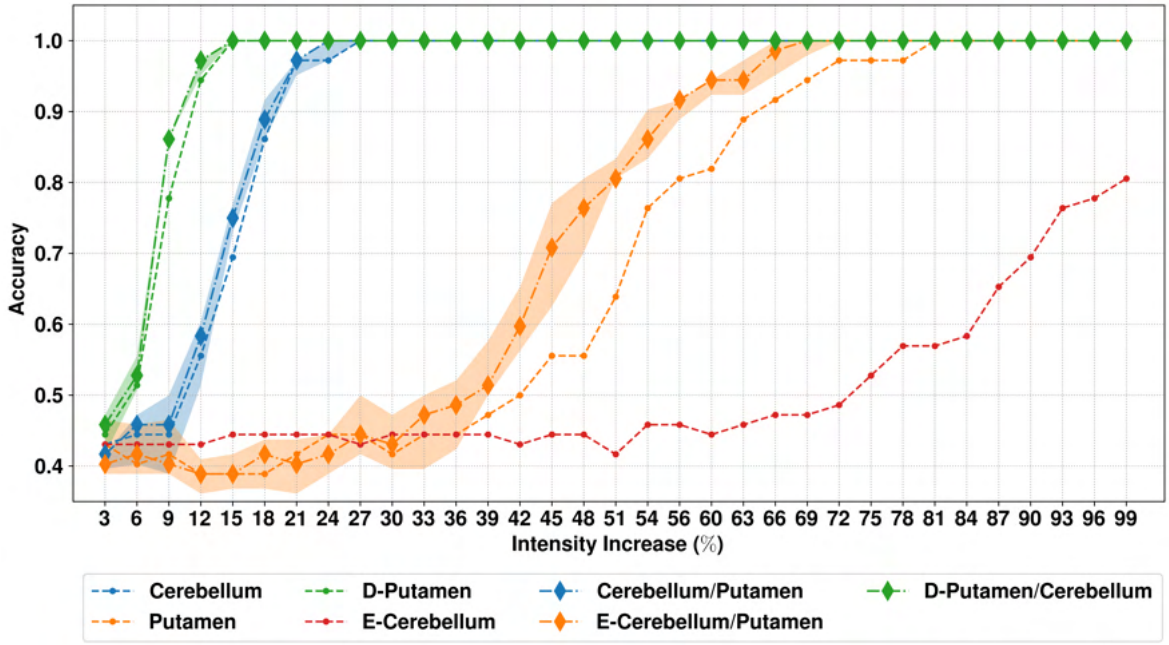


Figure XI: *Monoregion- vs. Biregion-Trained CNNs.* Performance comparison between monoregion- and biregion-trained CNNs. The latter were trained with biregion APMaps modified using the same intensity increase for both regions. Accuracy on the hold-out set is given as the median and IQR achieved with a 10-fold CV according to the intensity increase applied on APMaps. IQR of monoregion-trained CNNs is omitted for clarity. APMaps: Altered Parametric Maps; CNN: Convolutional Neural Network; CV: Cross Validation; D: Dilated; E: Eroded; IQR: Interquartile Range. Reproduced from [182] (Giulia Maria Mattia, 2021). CC BY-NC-SA 4.0

A.5 Visual Interpretation

Fig. XII provides visualization maps for D-Putamen and E-Cerebellum compared to their anatomical counterparts. We can notice comparable behavior to Cerebellum and Putamen CNNs, discussed in Section 3.3.4.

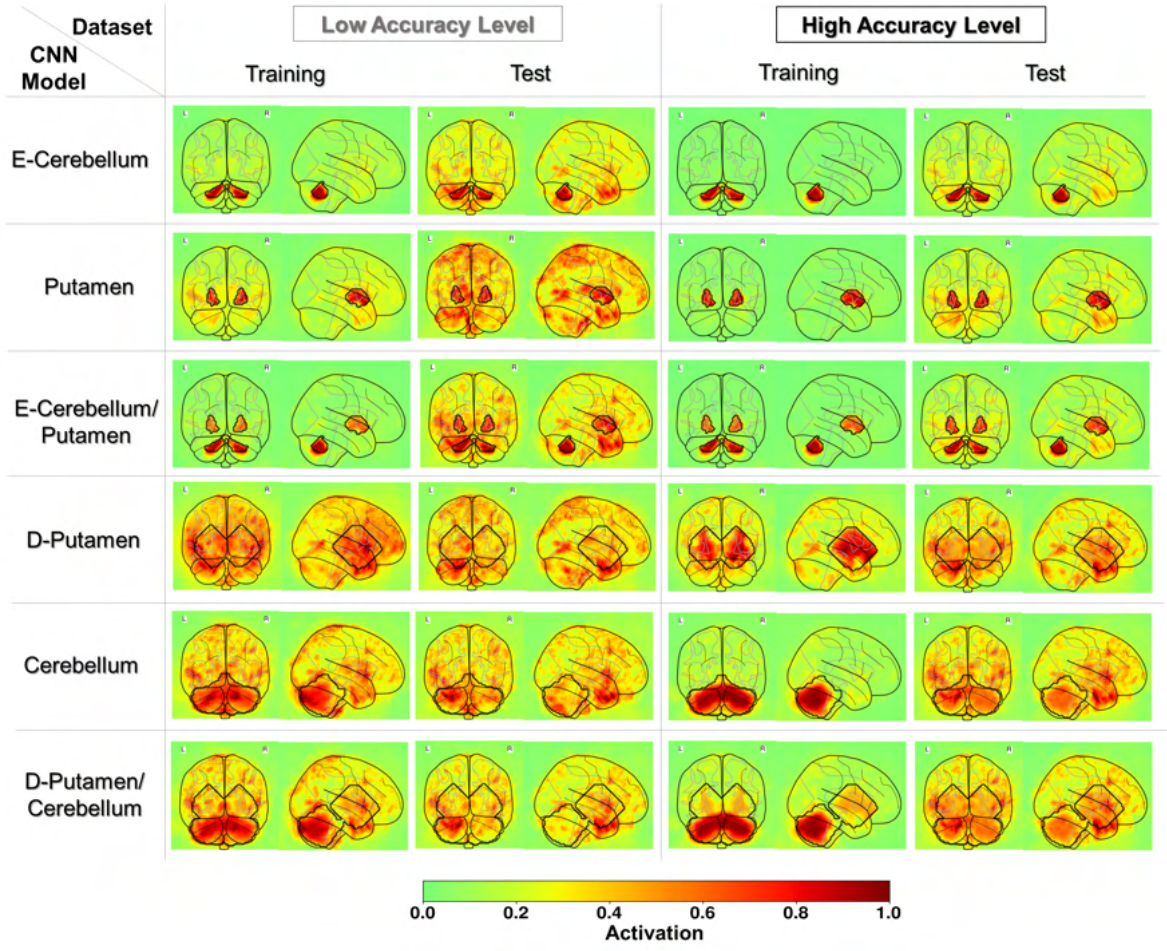


Figure XII: *APMaps for CNN Interpretability - Visual Interpretation.* Mean visualization maps showing the absolute difference between the average of correctly classified patients per class. We considered Low (L = 0.65) and High (H = 1.00) as accuracy levels per region. Black contours delineate the regions targeted in training. APMaps: Altered Parametric Maps; CNN: Convolutional Neural Network; D: Dilated; E: Eroded; IQR: Interquartile Range

Our primary focus was investigating CNN behavior accounting for the various accuracy levels. Aware that the intensity increase plays a role in pattern retrieval, we compared the high accuracy level (comprising regions with different intensity increases) and the case with regions presenting both maximum intensity increases.

To this end, we calculated visualizations maps for the maximum intensity increase available

(either 165% for E-Cerebellum or 99% for the other regions). For biregion APMaps, we kept the same intensity increase for both regions considering the maximum percentage (e.g. E-Cerebellum/Putamen with 165% since the E-Cerebellum has a maximum intensity increase of 165%).

We can observe from Fig. XIII that both training and test sets present the highest activations in the targeted regions with considerably reduced noise in the case of maximum intensity increase compared to the high accuracy level. In addition, considering the regions at the same maximum intensity increase, they presented comparable high activations to the case of different intensity increases (see the results on the training set for Cerebellum/Putamen with high accuracy level vs. max intensity increase).
These findings point out that there is still room for improvement to better understand CNN behavior via the use and interpretation of visualization techniques.

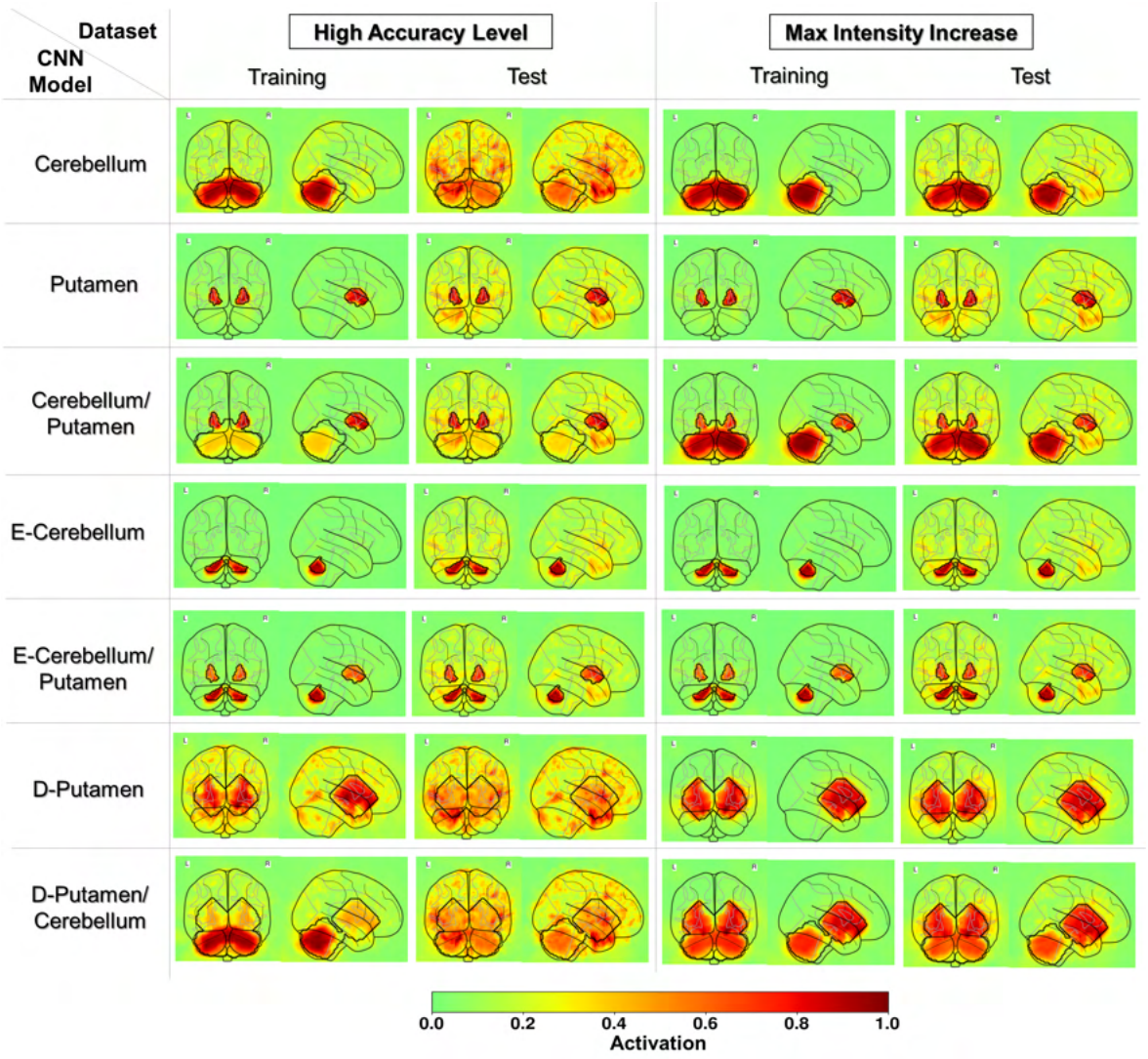


Figure XIII: *APMaps for CNN Interpretability - Visual Interpretation.* Mean visualization maps showing the absolute difference between the average of correctly classified patients per class. We considered the case with the high accuracy level and the one with the maximum intensity increase for all the regions. Black contours delineate the regions targeted in training. APMaps: Altered Parametric Maps; CNN: Convolutional Neural Network; D: Dilated; E: Eroded; IQR: Interquartile Range

B Investigation of Training Set Content

Fig. XIV provides the standard deviation for the performance of VGGNet model.
Fig. XV and XVI provide the results obtained by using the GoogLeNet and ResNet architectures described in A.2.1.
Performances were comparable to those achieved by the CNN architecture inspired by VGGNet (see Section 4.4.2.2.2). However, the Mild cluster performed slightly worse on the Severe (mean accuracy around 0.90 compared to 1.00 for the VGGNet), whereas we found similar trends regarding sensitivity and specificity.

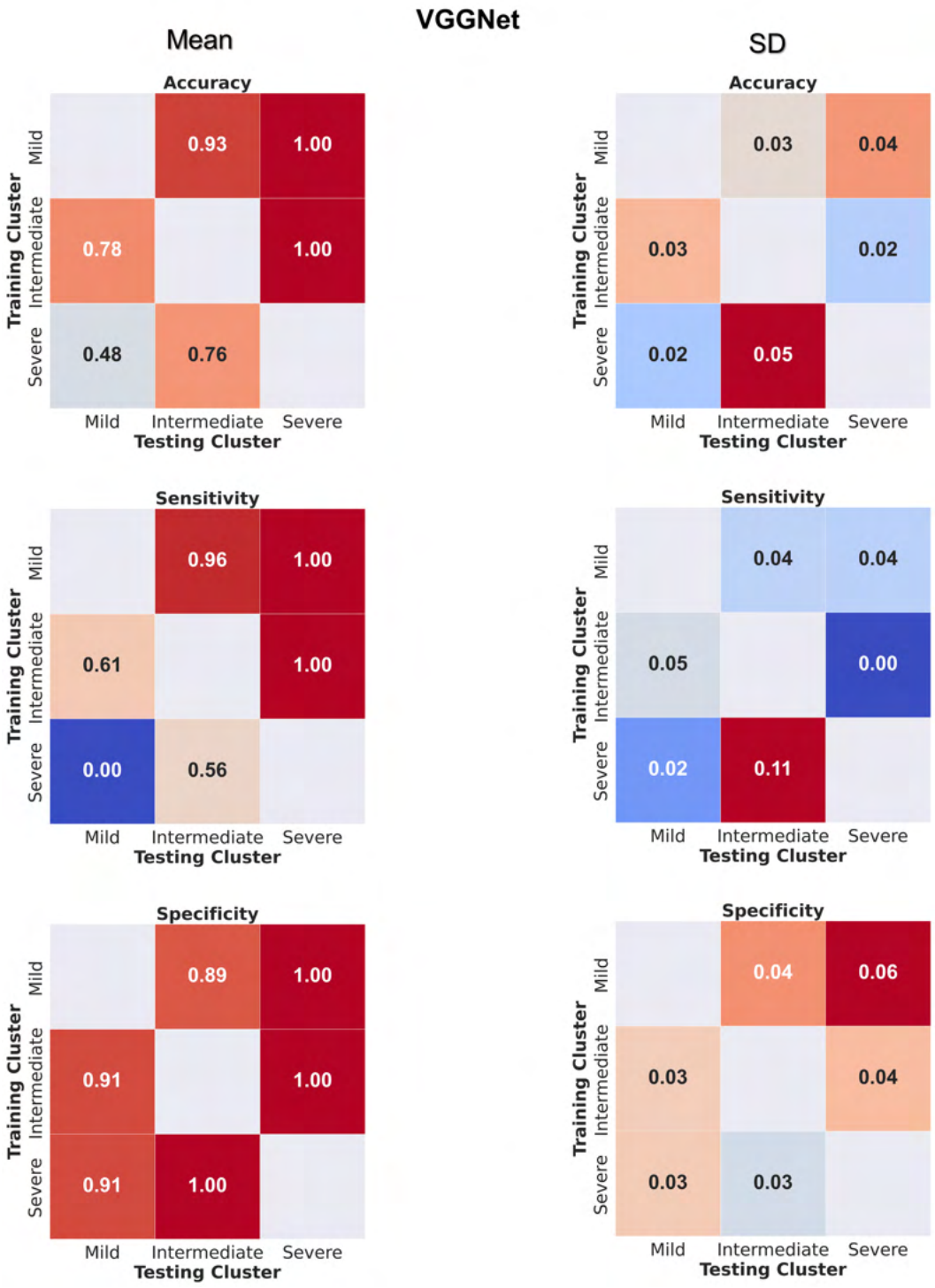


Figure XIV: Performances from the CNN inspired by the VGGNet architecture, according to the cluster of MSA patients used for training and testing. Blue represents lower values, whereas red indicates higher values. We report mean and SD over the 30 random samplings from the set of healthy individuals used for training. CNN: Convolutional Neural Networks. MSA: Multiple System Atrophy; SD: Standard Deviation

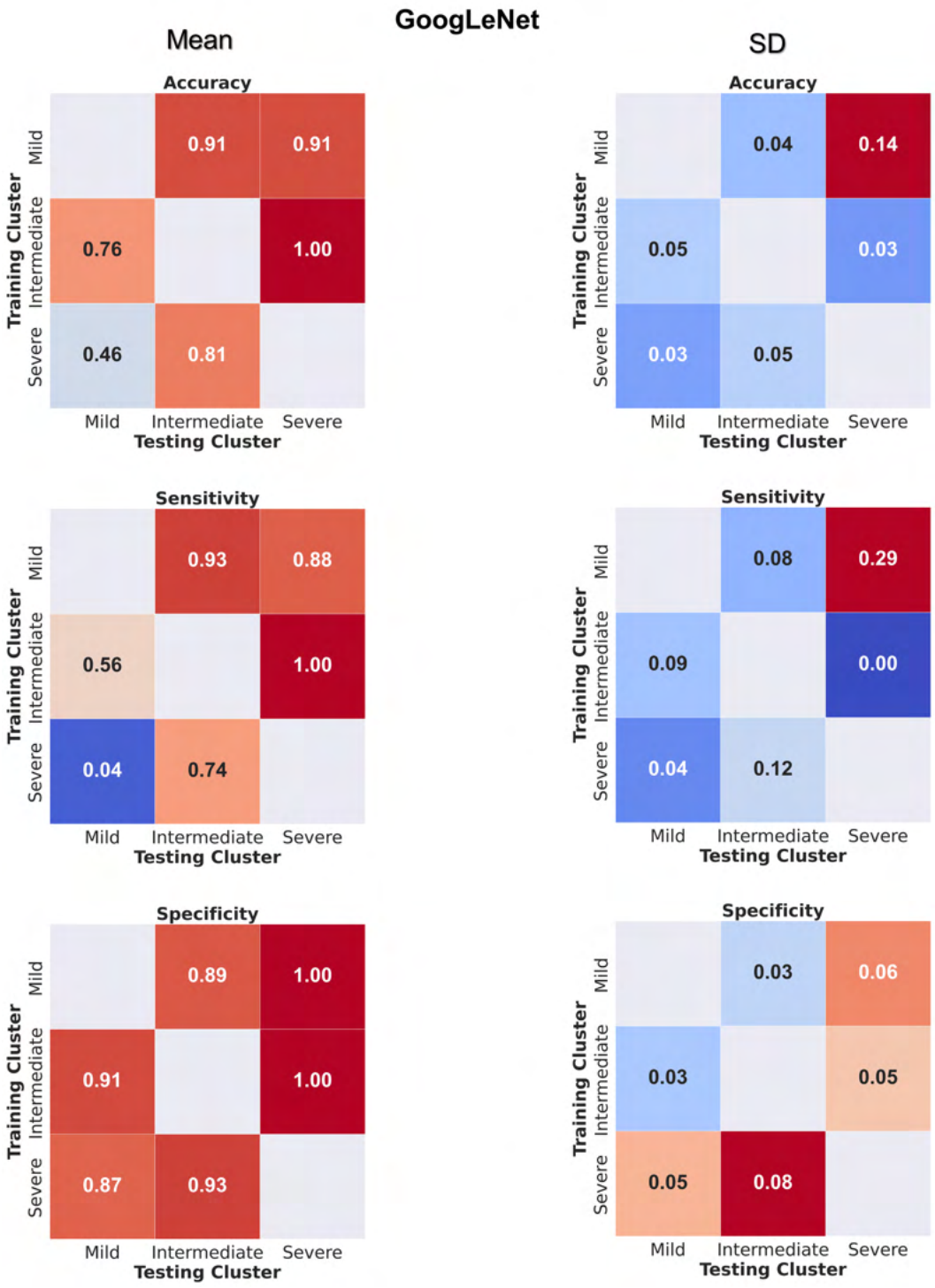


Figure XV: Performances from the CNN inspired by the GoogLeNet architecture, according to the cluster of MSA patients used for training and testing. Blue represents low performance, whereas red indicates high performance. We report mean and SD over the 30 random samplings from the set of healthy individuals used for training. CNN: Convolutional Neural Networks. MSA: Multiple System Atrophy; SD: Standard Deviation

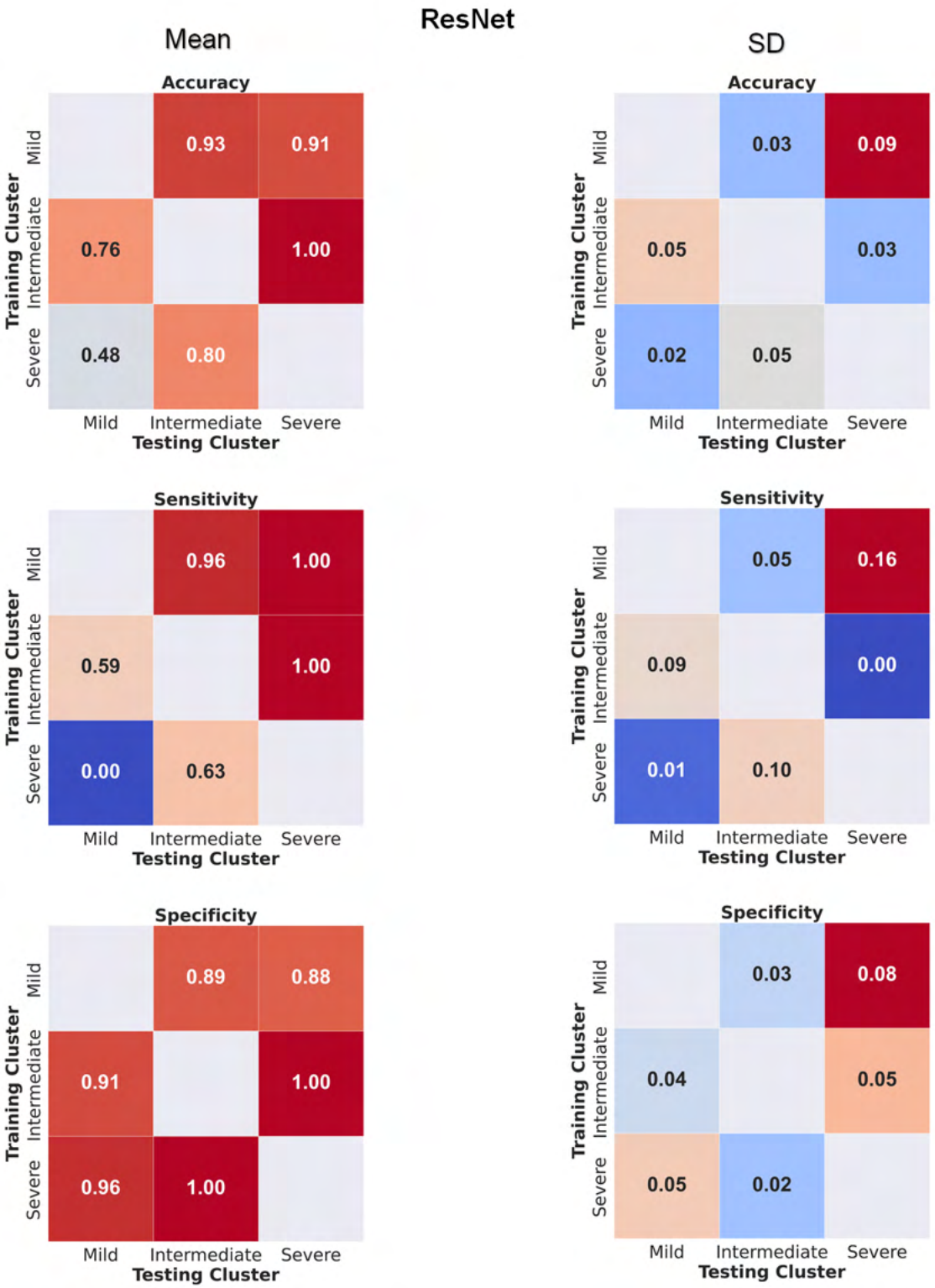


Figure XVI: Performances from the CNN inspired by the ResNet architecture, according to the cluster of MSA patients used for training and testing. Blue represents lower values, whereas red indicates higher values. We report mean and SD over the 30 random samplings from the set of healthy individuals used for training. CNN: Convolutional Neural Networks. MSA: Multiple System Atrophy; SD: Standard Deviation