PONTIFICAL CATHOLIC UNIVERSITY OF RIO GRANDE DO SUL
SCHOOL OF TECHNOLOGY
GRADUATE PROGRAM IN COMPUTER SCIENCE

FEATURE VISUALIZATION FOR DYSLEXIA IDENTIFICATION IN FUNCTIONAL MRI DATA

LAURA ANGÉLICA TOMAZ DA SILVA

Dissertation presented as partial requirement for obtaining the degree of Master in Computer Science at Pontifical Catholic University of Rio Grande do Sul.

Advisor: Prof. Dr. Duncan D. Ruiz
Co-Advisor: Prof. Dr. Felipe R. Meneguzzi

Porto Alegre
2019

REPLACE THIS PAGE
WITH THE LIBRARY
CATALOG INFORMATION

REPLACE THIS PAGE
WITH THE
PRESENTATION TERM

VISUALIZAÇÃO DE FEATURES PARA IDENTIFICAÇÃO DE DISLEXIA EM DADOS DE RESSONÂNCIA MAGNÉTICA FUNCIONAL

RESUMO

Dislexia é um distúrbio do desenvolvimento que afeta várias habilidades específicas de aprendizado, entre elas a leitura. É uma dificuldade de aprendizagem complexa caracterizada por um prejuízo significativo no desenvolvimento de habilidades de leitura não relacionadas a problemas de acuidade visual, escolaridade ou saúde mental em geral. Pesquisadores têm procurado por biomarcadores de dislexia usando ressonância magnética ou ressonância magnética funcional. Tais técnicas já fornecem aos especialistas em neuroimagem ferramentas para diagnosticar a dislexia (e muitos outros distúrbios de aprendizagem), mas conhecimento sobre os processos neurais no cérebro ainda são necessários. Com o objetivo de fornecer conhecimento significativo sobre a condição que está sendo classificada, adotamos algoritmos de aprendizagem profunda para classificação de dislexia. Modelos modernos de classificação de alta precisão (ou seja, aprendizado profundo) não são muito transparentes e, portanto, são menos úteis do ponto de vista da pesquisa em neurociência. Para resolver esse problema, utilizamos várias técnicas de visualização de rede para mostrar que o uso dessas técnicas em algumas camadas de rede neural convolucional pode fornecer *insights* significativos e apoiados por especialistas sobre a condição que está sendo classificada. Essas técnicas de visualização fornecem uma ferramenta para neurocientistas e outros pesquisadores se aprofundarem nas condições neurológicas sendo estudadas.

Palavras Chave: Aprendizado Profundo, Visualização de Features, Neuroimagem, Dislexia.

FEATURE VISUALIZATION FOR DYSLEXIA IDENTIFICATION IN FUNCTIONAL MRI DATA

ABSTRACT

Dyslexia is a developmental disorder affecting several specific learning skills, most commonly reading. It is a complex learning difficulty characterized by a significant impairment in the development of reading skills unrelated to problems with visual acuity, schooling or overall mental health. Researchers have been searching for biomarkers of dyslexia using MRI or functional MRI. Such techniques already provide neuroimaging specialists with tools to diagnose dyslexia (and many other learning disorders), but more insights into the neural processes in the brain are needed. Aiming to provide meaningful insights into the condition being classified, we adopt deep learning algorithms to subject classification. Modern high-accuracy classification models (i.e. deep learning) are not very transparent, and so, they are less useful from a neuroscience research perspective. To solve this problem, we leverage a number of network visualization techniques to show that using visualization techniques in some of convolutional neural network layers can provide meaningful and expert-backed insights into the condition being classified. These visualization techniques provide a tool for neuroscientists and other researchers to pry deeper into the neurological conditions they are studying.

Keywords: Deep Learning, Feature Visualization, Neuroimaging, Dyslexia.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ACERTA – *Avaliação de Crianças em Risco de Transtorno de Aprendizagem* - Evaluation of Children at Risk for Learning Disorder

AI – Artificial Intelligence

BOLD – Blood Oxygen Level Dependent

BNF – Backus-Naur Form

3D CNN – 3D Convolutional Neural Networks

CNN – Convolutional Neural Networks

DL – Deep Learning

FMRI – Functional Resonance Imaging

GP – Genetic Programming

GGP – Grammar-based Genetic Programming

GRAD-CAM – Gradient-weighted Class Activation Mapping

ML – Machine Learning

MRI – Magnetic Resonance Imaging

PET – Positron-emission tomography

SVM – Support Vector Machines

CONTENTS

1. INTRODUCTION

Dyslexia is a learning disability [HF+11] characterized by difficulties with specific skills, such as writing, spelling and most commonly reading. Symptoms and impairments caused by this learning disability on one's daily life can vary according to its stage of development. Dyslexia is a persistent condition that can lead to learning delays interfering with academic achievements into adulthood.

Dyslexia can be misdiagnosed due to the lack of clear diagnostic criteria [Ros09]. Reliable diagnoses can be behaviourally determined after some years of education, when the discrepancy between normal cognitive and reading abilities become evident. However, dyslexia can still be interpreted as other disorders (e.g., Attention Deficit Hyperactivity Disorder (ADHD) because of their symptoms similarity).

Trying to overcome this limitation, researchers [GSR+85, HMB+11, FSH+14] used other methods to find biomarkers among dyslexics in comparison to non-dyslexics. Neuroimaging techniques, such as Functional Magnetic Resonance Imaging (fMRI), were applied to obtain precise images of brain structure and its functioning within patients' brains affected by this disorder, finding evidence of anomalies among the group. fMRI is a neuroimaging technique that indirectly detects neural activity using magnetic properties peculiar to the brain [HSM04]. In our work, we analyze which brain regions can contribute to distinguish between dyslexics and non-dyslexics by using neuroimages from a specific type of experimental design: the task-related. In this experimental design, the scanned subject needs to perform a word reading task, having the metabolic changes in the brain captured by the MRI.

We applied deep learning techniques to create an fMRI-data driven model capable of differentiating dyslexic from non-dyslexic subjects. The use of deep learning algorithms is so the algorithm can discover patterns without having to manually define feature models as it is necessary in traditional machine learning approaches [TVGS16]. To create the model, we use the ACERTA *(Avaliação de Crianças em Risco de Transtorno de Aprendizagem, Evaluationof Children at Risk for Learning Disorder)* project dataset.

Our goal for the use of deep learning models is to be able to collaborate with neuroimaging specialists, using feature visualization techniques to help visualize how different brain structures can influence the classification of an individual as dyslexic. Feature visualization techniques are a recent research area in deep learning [ZF13]. Deep learning models generally perform well on different tasks, but there is no clear understanding of why this happens. For this reason, feature visualization techniques can be applied to deep learning models to provide insights of how the network model look at a given input. These techniques are able to turn deep learning models interpretable to humans, thus, in our work contributing to insights of which brain regions are most relevant for the differentiation between dyslexics and non-dyslexics subjects.

1.1 Contributions

The diagnosis of neurodevelopmental disorders is largely based on patient's behavior [WBB+15]. The problem with this type of diagnosis is that its accuracy depends upon a number of variables that are not easy to control, such as examiner skill and patient collaboration. Therefore, it is important to apply measurable and reliable diagnosis methods.

Task-based functional neuroimaging is a specific type of neuroimaging technique that requires great collaboration of the patient. This method is used to detect neural activity in response to certain experimental conditions, i.e. tasks or stimuli relevant to the research hypothesis. For instance, to activate language networks of the brain, a paradigm might contain reading tasks during the scan.

The main goal of this work is to provide insights of the differences that underpine a specific neurodevelopmental disorder (i.e. dyslexia) to neuroimaging specialists. Our main objective is to apply feature visualization techniques in deep learning models that are able to extract high-level abstractions of fMRI data as discriminant in dyslexia diagnosis. Therefore, aiming to provide a better comprehension of how a deep learning model classify a subject in dyslexic or non-dyslexic to neuroscientists without them needing to understand the complexity of such models. In this context, we propose the following research questions:

- Is it possible to explore visualization techniques of learned features from convolutional neural networks to reveal insights about the conditions being studied from fMRI data?
- How effective are deep learning approaches to classify small datasets of fMRI data?

Thus, the main contribution of our work is to design a deep learning model using neuroimaging techniques that can automatically filter the data to diagnose dyslexia based on whole brain data. We generate a deep learning model that is able to classify data with high accuracy. By analyzing gradients from the model, we found high activations in brain regions that are related to dyslexia and may help in dyslexia identification. Therefore, the contributions of this work are:

- Create a deep learning model using task-based fMRI data to discriminate subjects with and without dyslexia.
- Use feature visualization techniques to better understand which brain regions contributed to subject classification.
- Show the applicability of deep learning methods in neuroscience research projects.
- Evaluate the results of the feature visualization techniques with neuroscientists at the Brain Institute of Rio Grande do Sul (*Instituto do Cérebro do Rio Grande do Sul* - INSCER).

1.2 Document Structure

The remainder of this document is structured into 6 more Chapters. First, in Chapter 2, we present the application domain in which we want to apply classification. Next, in Chapter 3 we present the theoretical background through which we ground our proposed solution. In Chapter 4, we go over our approach to generate a visualization of features from our deep learning models to provide a better understanding of how dyslexia was classified by the model to neuroimaging specialists. Then, in Chapter 5 we detail experiments done to find a solution for the application domain and the results obtained with our approach. After that, in Chapter 6, we provide an overview of prior studies that are related to our proposed solution. Finally, in Chapter 7, we draw conclusions and indicate potential future work.

2. NEUROIMAGING

For many years, scientists had to rely on post-mortem autopsies to be able to study the most important and complex organ of the human body – the brain. With the advance of technology and research, scientists were able to see the inside of a living human brain; however, invasive techniques were still used. In order to measure brain activity, electrodes were placed within the brain to investigate behavioural, perceptual and cognitive processes [KSW15]. Other significant experiments on, for instance, motor and perceptual function were performed in awaken patients during brain surgery for treatment of epilepsy or tumor removal [Pen47].

Neuroimaging or brain imaging technology encompasses a variety of non-invasive techniques that produce images of different brain structures; therefore, not requiring surgery, incision of the skin, or exposing the brain to harmful radiation. One of these non-invasive imaging techniques is Magnetic Resonance Imaging (MRI). MRI uses a magnetic field and radio waves to create detailed, high quality, and high-resolution three-dimensional anatomical images. A typical research scanner has a field strength of 3 teslas (T), which is about 50,000 times greater than the Earth's field. During the image acquisition process, a strong and constant magnetic field is produced around the area to be imaged forcing the protons (hydrogen protons from water molecules) in tissues of the body to align with that field. Throughout the scan field magnetization may vary, but tissue magnetization not necessarily changes at the same rate for all tissues.

After the alignment of the protons to the scanner magnetic field, radio frequency waves (pulse sequence) are pulsed through the patient aiming at knocking protons out of alignment. When the radio frequency field is turned off, the MRI sensors are able to detect the energy released by the protons as they realign with the magnetic field generating the activation signal. The time taken for protons to realign with the magnetic field and the energy released changes depending on the environment and chemical nature of the molecules. The contrast seen between different tissues in the generated image is determined by the pulse sequence used. Due to different pulse sequences, the MRI scanner can distinguish between white matter and grey matter and also be used to diagnose aneurysms and tumors. MRI is a flexible and powerful clinical tool used to examine multiple biologically interesting properties of tissues, yet it only reveals the anatomy of the brain. Studying the structure of the brain helps understanding neurological disorders and patterns related to those conditions. On the other hand, MRI does not provide information on physiological changes in the brain caused by neural activity. To overcome this limitation, Functional Magnetic Resonance is used.

2.1 Functional Magnetic Resonance Imaging (fMRI)

Functional Magnetic Resonance Imaging (fMRI) is a noninvasive imaging technique for obtaining three-dimensional images demonstrating regional, time-varying changes in brain metabolism

[HSM04, MHN+04]. Metabolic changes can be the result of simple tasks such as controlling your hand to open a door or complex cognitive activities such as reading, as well as present patterns of neural activity when we are at rest revealing particular networks of correlated areas in the brain [HSM04]. fMRI has been applied in many studies. For instance, fMRI is being used to better understand how the healthy brain works and how the normal function is disrupted in disease. Also, it is largely adopted in cognitive neurosciences, clinical psychiatry/psychology, and presurgical planning [Glo11].

fMRI is a relatively new form of neuroimaging used to measure and map brain activity by changes in blood flow through time [MHN+04]. It detects changes in blood oxygenation and flow that occur in response to neural activity, i.e., when a brain region is more active it consumes more oxygen increasing the demand of blood flow to this, now, active region. More accurately, fMRI measures the ratio of oxygenated to deoxygenated hemoglobin in the blood, being able to distinguish between both states because of their magnetic properties [OLKT90, OL90], with respect to a control baseline at many individual location within the brain [MHN+04]. Blood oxygen level is said to be influenced by local neural activity, therefore, fMRI is an imaging technique that uses blood oxygen level-dependent (BOLD) response as indicator of neural activity [MHN+04].

The fMRI scanner assesses the value of the fMRI signal (BOLD response), generating a four-dimensional functional neuroimage: a sequence of three-dimensional images over a time interval, called time series. The three-dimensional points within the image are called voxels (volume elements), or 3D pixels. A typical three-dimensional brain image has 10,000 to 15,000 voxels containing cortical matter and each voxel contains hundreds of thousands of neurons.

The haemodynamic response (HR) triggered by a rapid event, lasting less than a second, boosts blood delivery to the active neuronal tissue showing an increase in fMRI BOLD signal for many seconds, enduring from four to five seconds to rise to peak and five to seven seconds to return to baseline [MHN+04]. The figure 2.1 shows part of fMRI data collected over a fifteen second interval in which the subject read a word and decided if it was a noun or a verb (in this example, it was a noun) to then wait for another word [MHN+04].

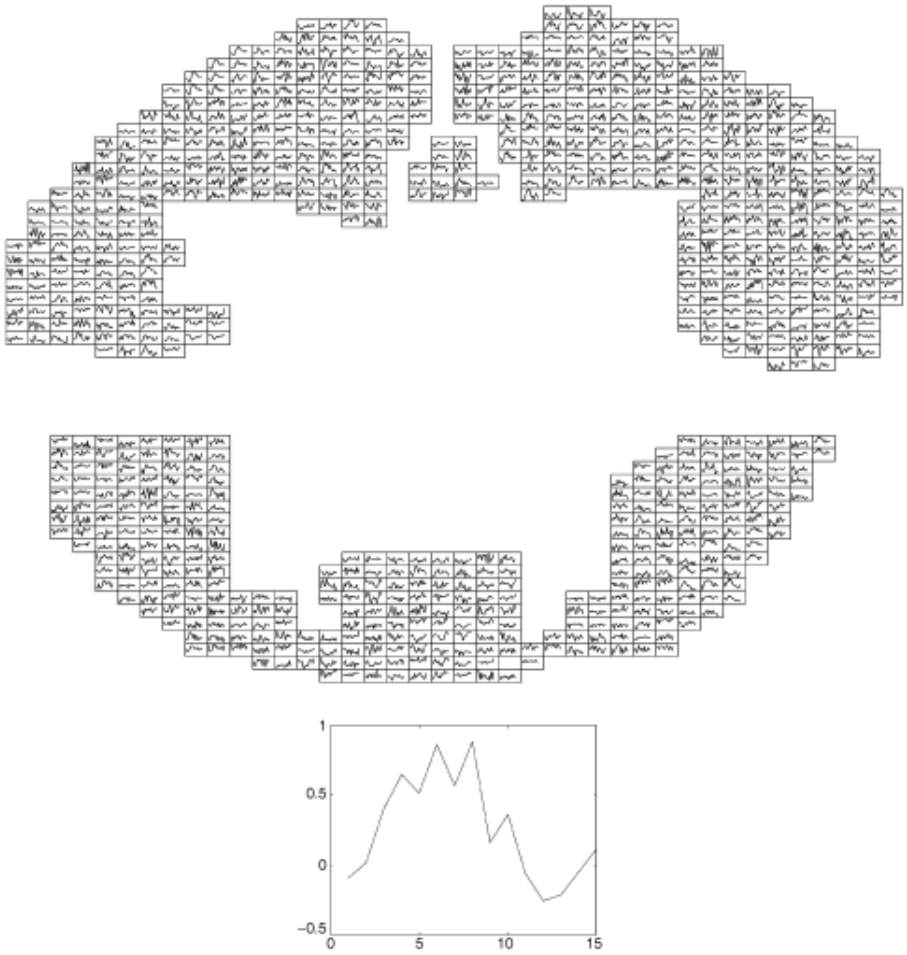

Figure 2.1 – Typical fMRI data. The top part of the figure presents fMRI data for a selected set of voxels in the cortex, from a two-dimensional image plane through the brain. A fifteen second interval of fMRI data is plotted at the location of each voxel. The bottom part of the figure shows one of these plots in greater detail. Adapted from [MHN+04]

2.2 Preprocessing fMRI data

The MRI scanner is susceptible to any generating source of magnetic field, creating what is called noise in the acquired images. Noise can result from thermal sources of heat-related motion of electrons in the scanner, in the patient and in the environment surrounding them, bulk motion of the patient's head, cardiac and respiratory-induced noise, as well as variations in baseline neural metabolism. Thus, separating BOLD signal from noise is a very important step for fMRI experiments, which requires the data to be passed through a preprocessing phase [HSM04]. These steps are crucial in making statistical analysis valid and supporting model assumptions.

The most common steps in preprocessing are: 1) slice timing correction, a technique used to eliminate differences between the time of acquisition of each slice in the volume since the whole brain is not scanned once but with a series of successives slices; 2) high-pass and/or low-pass temporal filtering to remove unwanted temporal frequencies of a time series, such as physiological

noise from breathing, cardiovascular functions or scanner noise, and signal generated from other tissues such as the skull and the fat that envelops the brain that are of no interest; 3) head motion correction or motion coregistration, this approach is used to adjust all the images to a high-resolution reference image from the subject's brain to maintain activation data and voxels always in the same position; 4) spatial normalization, human brains are variable in sizes and shapes, thus this step enables comparison of brain activations across subjects by coregistering each fMRI to a specific brain template so that a given voxel represents the "same" location from brain to brain; 5) spatial filtering or smoothing, its main goal is to remove noise and at the same time retain the signal of interest, improving the signal to noise ratio (SNR).

These are some of the routinely used techniques to improve data quality in fMRI studies [HSM04]. Without appropriate preprocessing of the data prior to analysis, the model assumption would not hold and the resulting statistical analysis would be invalid. Consequently, preprocessing is used to increase the plausibility of the fMRI results and confidence in the interpretations.

2.3 Dyslexia

In 1896, W. Pringle Morgan published an article containing noted cases of individuals with apparently normal intelligence who could not learn to read in the British Medical Journal, calling it, at the time, word blindness. In a brief description of one case, he said, "Percy F., aged 14, has always been a bright and intelligent boy. Quick at games, and in no way inferior to others of his age. His great difficulty has been and is now his inability to learn to read" [Mor96]. This introduction has intrigued scientists to research why some people face difficulties in learning to read.

Dyslexia is a learning disability characterized by difficulties with accurately and/or fluently recognizing words (reading) and by poor spelling and decoding abilities [LSS03]. It represents one of the most common problems affecting children and adults (80 % among those having learning disabilities) [HF$^+$11].

Symptoms may vary from one individual to another as well as the impact dyslexia has in their lives. Dyslexic children may experience writing and reading letters and words backwards in early stages of learning to read and write. Another sign of dyslexia is connected to trouble decoding words, a language skill called phonological awareness. This ability is responsible for an individual's identification of sound structures of words, such as syllables and the smallest units of sounds, phonemes; orthographic coding, what can be described as the composition of visual memory representations of letters, letter patterns, and sequences of letters in order to map phonemes within words; short-term auditory memory, which helps us to hold and assimilate spoken languages; and a rapid automatic naming, consisting of being able to quickly name aloud objects, pictures, colors and symbols. Therefore, the lack of these patterns of basic speech units and rules of pronunciation make difficult to use the alphabetic code to decode written language.

Dyslexia is neurobiological in origin, which means that the problem is located physically in the brain. Findings in brain anatomy and activity show brain differences between people with and without dyslexia. Postmortem studies have indicated abnormalities in the brains of dyslexic individuals [GK79, GSR+85], revealing an absence of the usual asymmetry in the planum temporale, which is one of the most important functional area for language. Structural differences among those individuals were also seen in the corpus callosum, responsible for controlling the communication between the two hemispheres of the brain [RH98, RBDH00]. Neuroimaging techniques have also been used in studies to find differences among dyslexic and non-dyslexic individuals [CGLdlTC05, FSH+14].

Most of these studies were conducted in adults, and the findings were used to infer what might be found in children with dyslexia. This approach made possible to evaluate children in the first signs of learning difficulties. Assessments are usually available in schools and institutions of higher education, involving measurement of reading, spelling, writing, rapid naming, rhyme, motor aspects and arithmetic skills [RPF03]. One standard test developed to identify developmental delay [Tor98] is The Woodcock Reading Mastery Test-Revised [Tor98], containing norms for performance on letter-naming.

Still, standard tests are not 100 % accurate in the identification of dyslexia [HMB+11]. For this reason, other methods must be researched to correctly detect dyslexia in individuals and provide specific treatments for their disability based on the severity of each case. One example is the ACERTA project, that attempts to find solutions for a more accurate prognosis.

2.4 ACERTA

ACERTA[1] (*Avaliação de Crianças em Risco de Transtorno de Aprendizagem*, Evaluation of Children at Risk for Learning Disorder) is a conjunct project of three research centers in Brazil. It has three main goals:

- better understand how the children's brain change during the literacy development and how dyslexia influences this process;
- raise awareness and provide information on learning disabilities in school communities;
- engage university cooperation, bringing Medical, Psychology, Literature, Mathematics, Education, Engineering and Informatics schools together in the research stages.

The project started in 2013 [TLB16], evaluating children in literacy phase through the *Provinha Brasil* (Little Test Brazil) in three Brazilian cities: Porto Alegre, Florianópolis and Natal, where the research centers are located. This test can be performed either at the beginning or end of school term of the first or second year (Primary school). It evaluates children in Portuguese language, reading

[1] For more information visit: www3.pucrs.br/portal/page/portal/inscer/Capa/ACERTA

and mathematics. The skills evaluated in reading [PBL18] are: recognition of letters and syllables, identification of the relationship between phonemes and graphemes, capacity to locale information, understanding of the relationships between parts of the text, and inferring information. The other subject on the test is mathematics [PBM18], having the following competences measured: creation of concepts and structures in relation to the meaning and representations of numbers, problem solving with addition, subtraction (and application of ideas leading to multiplication and division), recognition of geometric shapes, conceptualization of magnitudes and interpretation of graphical and textual data. The results from the test are used with neuropsychological and psychoeducational assessments to complement neuroimaging data in the ACERTA project.

2.5 Identifying Dyslexia

Salles et al. [SPZT13], proposed a cognitive task to identify dyslexia. The task consisted of identifying types of words that appeared in a screen while subjects had their brain scanned. The words were in Portuguese and divided into two categories: Non-Word and Word. Non-Words are junctions of random syllables appearing to be real words and respecting Portuguese spelling rules. Words are existing words subdivided into two classes: Regular and Irregular.

Regular words contain letters with their usual Portuguese pronunciation, for instance the letter <X> in the word "peixe". On the other hand, Irregular words consist of letters which its pronunciation sounds differs from the pronunciation of other sets of letters, such as the letter <X> in the word "táxi" that sounds like <CS>.

Children with dyslexia are supposed to have different brain's activation while performing word classification. This occurs due to every representation of the word within the subject's brain activate a different region responsible for a different concept of the representation. For example, lexical orthographic representation in the visual area. The adoption of machine learning is proposed to identify dyslexia through discrepant brain's activation during reading process. During the preparation phase, classification of words that are obtained in the fMRI are crossed with time series, generating a classification that represents the expected value. Consequently, this classification is used in learning process to identify different activation for a presented word.

3. MACHINE LEARNING

The ability to learn and gain knowledge is the most prominent feature of human intelligence. Machine Learning (ML) is a field of Artificial Intelligence (AI) designed to build algorithms able to learn through experiences. ML investigates how to simulate human learning to achieve computational intelligence. To be intelligent, a system needs to be capable of learning and adapting to a changing environment.

The concept of learning has been widely discussed. Mitchell [Mit97] formally defines it as: "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. (Mitchel 1997, p. 2)". To better understand this definition, T refers to the task performed by the machine learning algorithm. Generally, T is difficult to articulate in a formal way by humans. Some examples of tasks are speech and face recognition, image classification, object detection and many more. A performance measure P is a quantitative method used to evaluate the ability of the algorithm in the learned task, it allows the algorithm to modify and improve its behaviour according to performance results of the chosen metric in future tasks. At last, an experience E is the data with observations from tasks already executed being the source of experience for the learning algorithm given the task domain. Some applications of ML algorithms are sales prediction, spam filtering, autonomous car driver, detection of fraudulent use of credit cards, and others.

The advances of computer technology made possible to store and process large amounts of data. It creates the opportunity to collect data from diverse fields to be analyzed and turned into information to identify patterns in the originated data. Even though it is not possible to understand the complete process that explains the data, we can construct approximations. These approximations are known as hypothesis $h(x)$. Finding the best $h(x)$ in the hypothesis space can be seen as the learning step of an algorithm [Mit97, FLGC11].

Machine Learning can be broadly divided into three categories of learning process [RN10]: supervised learning, unsupervised learning, and reinforcement learning. These categories are based on available experience during the learning process, which have its particularities. In the interest of this work, we present in more detail supervised and unsupervised learning.

Supervised learning algorithms are presented with a dataset containing features, and each example is associated with the desired output (label or target). The term supervised learning comes from the idea that the targets act as an instructor or teacher to the algorithm [GBC16]. Each training instance for supervised learning can be described in terms of input-output pairs (x_i, y_i), where x_i is a feature vector associated to a class y_i. This type of learning includes classification (given an example identify to which category it belongs, e.g., if an email is spam or not) and regression (generates a real value for an example, such as predicting house prices for determined house size).

Unsupervised learning algorithms are designed to recognize useful properties of the dataset structure. Unlike supervised learning, there are no labels or targets (outputs y) to the set of inputs x. Unsupervised learning does not have an instructor or teacher, and the algorithm needs to make

sense out of the data without guidance. This method observes several examples of a feature vector x, and attempts to model the underlying structure or distribution in the unlabeled data. A well-known example of unsupervised learning is *clustering*. Clustering algorithms aim to discover inherent groupings in the data, dividing it into clusters based on the similarity of their features. In relation to deep learning, unsupervised learning algorithms attempt to learn the probability distribution that originated the dataset [GBC16].

3.1 Traditional Machine Learning Approach in Neuroimaging

In this section, a subset of supervised algorithms that have been used with neuroimaging data are briefly described.

3.1.1 Support Vector Machines

Support Vector Machines (SVMs) [RN10] are discriminative linear classifiers based on the concept of decision planes that defines decision boundaries learned between different classes, linear or nonlinear. SVMs are a supervised learning technique that works by estimating an optimal hyperplane that best separates the given classes. When these classes are not linearly separable, SVM uses external functions (kernels) that map the original data into a new feature space where the data become linearly separable [CV95].

A SVM-based classifier for a set of binary-labeled training data separates the classes with what is known as the maximal margin hyperplane, which is maximally distant from two classes (for example, Dyslexic and Normal classes). The objective is to build a function that will correctly classify new examples. For instance, Linear SVM parameters define a decision hyperplane in the multidimensional feature space [Bur98, IGL$^+$11], that is:

$$g(x) = w^T x + b = 0 \tag{3.1}$$

where $\boldsymbol{x}$ represents the feature vector, $\boldsymbol{w}$ is known as the weight vector and b is the threshold. The decision hyperplane position is determined by vector $\boldsymbol{w}$ and b: the vector is orthogonal to the decision plane and b determines its distance to the origin. For linear SVM, the vector $\boldsymbol{w}$ can be explicitly computed. The design of the classifier consists of finding the unknown parameters, that is, w components of w (w_n, $n = 1...M$, where M = number of features) and b, which allows building a hyperplane that separates the two classes optimally.

Despite its popularity, SVMs have been criticized for not performing well on raw data and requiring the expert use of design techniques to extract the less redundant and more informative features (a step known as "feature selection") [BGC15, PHS$^+$14]. These features, rather than the original data, are then used for classification. SVM still remains a very popular technique within

the neuroimaging community, but an alternative family of ML methods known as deep learning (DL) [B+09] is gaining considerable attention in the wider scientific community [BGC15].

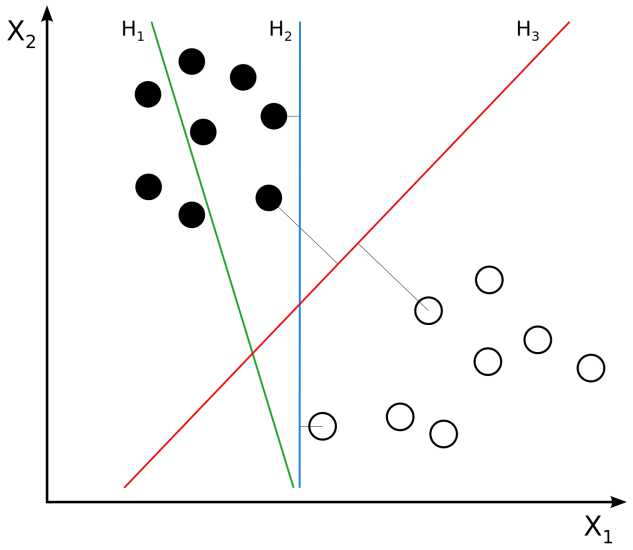

Figure 3.1 – Representation of how support vector machines would choose the best line to separate two classes of points. The hyperplane H1 does not separate correctly the classes. The hyperplane H2 does, but only with a small margin. And the hyperplane H3 separates the examples with maximum margin.

SVMs are basically applied to binary classification problems, which can be considered a limitation of this algorithm. To allow SVMs to be used in multi-class classification, extension of SVMs have been made. Two approaches to solve this problem can be described. One is by constructing and combining several binary classifiers to find the correct class by a voting system while the other is by directly considering all data in one optimization formulation.

3.2 Artificial Neural Networks

Research in artificial neural networks has been increasingly popular after a decade of lower activity [BGC15]. Artificial neural networks (or neural networks) are computing systems inspired by biological neural networks aiming to solve problems in the same way as a human brain would [Hay94]. The systems are able to gradually improve their performance (learning) on tasks by observing examples from the data. For instance, a common approach to image recognition is manually labeling examples, such as cats or dogs, and use these images to identify other cats and dogs throughout the dataset. The knowledge to classify cats or dogs, in the example, is not acquired a priori, meaning that it does not know cats and dogs have tails or fur among other characteristics. On the contrary, the network evolves its own set of characteristics information from the learning environment.

Neural networks are composed of computing cells (processing units or nodes) called neurons. Making a parallel to the biological inspiration, each neuron receives input signals from its dendrites and produces output signals along its axon. The axon branches out and connect via

synapses to dendrites of other neurons. In the computational model, the connection between neurons are synaptic weights (network weights), transmitting signal from one neuron to another and storing acquired knowledge [Hay94]. That is, the signals that travel along the axon (e.g. x_0) interact by multiplication (e.g. $x_0 \times w_0$) with the dendrites of the other neuron based on the synaptic strength at that synapse (e.g. w_0). Synaptic strengths (weights w) are adjusted in the learning process and their weight can increase or decrease the influence (and the direction: excitatory (positive weight) or inhibitory (negative weight)) of a signal at a connection. Basically, dendrites carry the signal to the cell body to be summed. After the sum, the value of the bias(b) is added to the result. At the end, if the final sum is above a predefined threshold, the neuron will forward the signal to the next connection. This is done by executing an activation function f. The simplest activation function is the threshold function, which takes the value 1 if the signal exceeds certain threshold and 0 otherwise. In neural networks, neurons are frequently organized in layers, each layer can perform different types of transformations on their inputs.

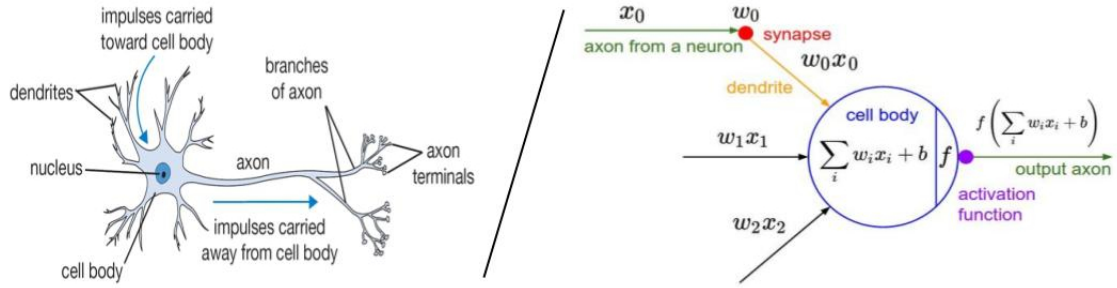

Figure 3.2 – Representation of Neurons [2]. The left part of the image shows a biological representation of a neuron. The right part of the image represents the mathematical model of a neuron in a neural network.

In neural networks, the process of adjusting the network weights to minimize errors in classification is called backpropagation [LBD+89]. Backpropagation algorithm is the current dominant approach employed in multilayer networks [BGC15]. The goal of backpropagation is to optimize the weights so the neural network can learn to correctly map inputs to outputs. In other words, backpropagation performs the update of each of the network weights to approximate the network output to the target output. This is done by calculating the difference between the network output and the target output, i.e, how far from the expected value the network output was. After that, the difference is propagate backwards in the network to adjust the weights [Alp14]. One of the limitations of this algorithm is the vanishing gradient [GBC16]. It refers to gradients that become vanishingly small at the first layers, preventing the weight from changing its value. To overcome this limitation, researchers proposed new methods, presented in the sequel.

[2] Adapted from http://cs231n.github.io/

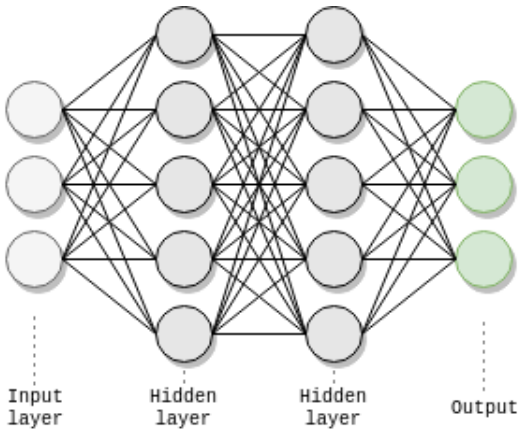

Figure 3.3 – Architecture of an artificial Neural Network.

3.3 Deep Learning

Traditional machine learning approaches face the *curse of dimensionality*. Cursed phenomena occur in high-dimensional spaces (data points described using many features) because the number of possible configurations of $\mathbf{x}$ becomes much larger than the number of training examples available in X [Mur12]. Many machine learning algorithms examine neighboring data points to determine the most probable behavior of an unseen data point, but the concept of proximity becomes blurry in sparse high dimensions since most configurations will have no training example associated with it [Alp14, B$^+$09, BCV13]. Such cursed phenomena undermines the performance of traditional machine learning algorithms, making it harder to efficiently generalize.

This failure is what, in part, motivated the development of Deep Learning (DL). Deep learning, a specific research area in machine learning, has taken the spotlight in recent years [BCV13]. The term *deep learning* refers to the deep architecture the brain is organized, in which each area in the cortex is responsible for interpreting different stimuli [BGC15]. A study conducted by Huebel and Wiesel (1962) to understand the primate visual system, observed that the brain first extracts edges, then patches, then surfaces, followed by objects and so on [HW62]. Therefore, each level is learning features or representations at different levels of abstraction.

Inspired by the architectural depth of the brain, neural network researchers decided to replicate the same architectural concept in a computer. Deep learning methods aim at learning feature hierarchies, building higher concepts on top of lower simple ones [B$^+$09, Mur12]. The ability to automatically learn features at several levels of abstraction allows the system to learn complex functions, without depending entirely on human-crafted features.

The first successful application of this method was reported in 2006 [B$^+$09]. Researchers had positive results in a deep multilayer neural network with two or three layers (one or two hidden layers); however, the deeper the network the poorer the results. Hinton et al. (2006), introduced Deep Belief Networks using stacking learning units with an external learning algorithm to train one layer at a time [HS06]. After that, other algorithms exploiting the same idea were proposed.

Autoencoders are neural models with one particular task: reconstruct its input at the output. In order to do so, it tries to find hidden structures within the data not requiring it to be labelled, for this it is considered an unsupervised learning algorithm [B+09].

With the increase of computational power and available data, deep learning is being applied to many domains. In addition, deep networks were able to find solutions to unsolved problems in AI and discover intricate structures in high-dimensional data [B+09]. Some deep learning applications are: sentiment analysis, language translation, prediction of potential drug molecules activity, reconstruction of brain circuits, prediction of the effects of mutations in non-coding DNA on gene expression and disease [B+09].

The popularity of this research area led to several other deep learning methods and techniques. One of them is Convolutional Neural Networks explained on the next section.

3.3.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs or ConvNets) are specialized in processing data in the form of multiple arrays, such as images (2D), audio and video or volumetric data (3D) [BGC15]. The first use of CNNs was to recognize simple shapes, in this case handwritten digits [LBD+89]. However, CNNs only attracted more attention recently when this architecture won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [KSH12]. After that, CNNs have been employed in many areas and are state of the art in several applications, especially in computer vision.

The term "Convolutional Neural Network" indicates that a mathematical operation called *convolution*, which is a special type of linear operation denoted by the $*$ symbol, is used by the network. In short, convolutional neural networks use convolutions in place of general matrix multiplication in at least one of the layers [GBC16]. In the case of a two-dimensional image I and a two-dimensional kernel K, a single two-dimensional convolution operation S for the point (i, j) is given by applying the kernel filter on top on an (m, n) overlap segment of the image, which is expressed by the formula:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \tag{3.2}$$

this represents a computation of the inner product between a vectorized kernel and a vectorized segment of the image. The filter is then slided on top of the whole image. The output of the convolution operation is sometimes referred to as *feature map*, and it is usually followed by an activation function for nonlinearity [GBC16].

There are four key ideas behind CNNs, according to LeCun et al. (2015): local connections, shared weights, pooling and the use of many layers.

In traditional neural networks, matrix multiplication is used to describe the interaction between each input and output unit. For example, an image as input will have each pixel value

connected to every neuron in the following layer. That means each neuron in the next layer is getting input from every part of the image. CNNs, on the other hand, exploit spatially-local correlation by enforcing a local connectivity pattern between neurons of adjacent layers. In other words, each neuron on adjacent layers only receives input from a small local group of close together pixels in the input image.

The idea of shared weights in CNNs means that rather than learning a separate set of parameters for every location, the network learns only one set [GBC16]. By comparison, in neural networks each element of the weight matrix is used exactly once to compute the output of a layer. That is to say, the weight is multiplied by one element of the input and then never revisited. In CNNs, each part of the kernel (matrix weight or filter) is used at every position of the input [GBC16].

In this sense, the convolution operation consists of a filter K (or kernel) over an input image I. The size of K is smaller than I. At each step K overlaps with a region of I computing the inner product between them. The step size is called stride and it refers to the size of the jump K will perform between regions of I. This layer is called convolutional layer and the output produced by it is called feature map. Multiple feature maps compose each computational layer of the network [Hay94].

Pooling (or subsampling) layers perform local averaging and subsampling, hence reducing the dimensionality and selecting the most relevant features on images [Hay94]. CNNs consist of subsequent convolution and pooling layers, not necessarily in that order (i.e., the network can have several convolutional layers before a pooling layer). Both layers composes the feature extraction step.

To exemplify this process we will think of images (array data) as input to the network. Images have local groups of often highly correlated values that form distinctive local motifs. Also, local statistics and other signals are invariant to location in images. This means that a motif in one part of the image could appear anywhere; therefore, units at different locations share the same weights and detect the same pattern in different array parts. The role of the convolutional layer is to detect conjunctions of features from the previous layer, while the pooling layer merges features that are semantically similar into one. A pooling unit generally computes the maximum of a local patch of units in one feature map (or in some feature maps), reducing the dimension of the representation and becoming invariant to small shifts and distortions. Even if elements vary in position and appearance in a previous layer, pooling allows representations to vary very little. Under the same observation of the primate visual system, in images local combinations of edges produce motifs, motifs assemble into parts, and parts form objects [BGC15].

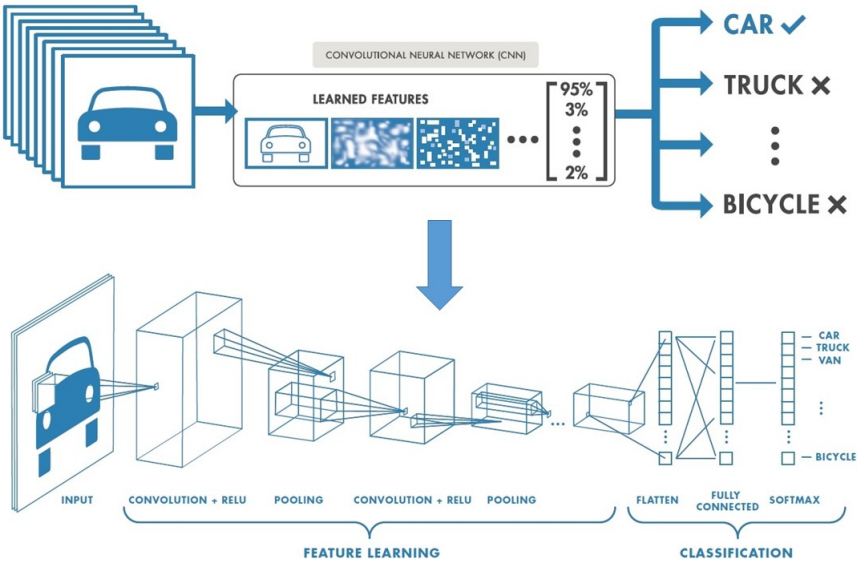

Figure 3.4 – Representation of a convolutional neural network for image classification[3].

3.3.2 3D Convolutional Neural Network

3D Convolutional Neural Networks (3D CNNs) are networks capable of dealing with volumetric images, or changes through time. The same principles used in 2D CNNs can be extended to 3D by using 3D convolution kernels and 3D pooling to obtain systems that can be applied to volumetric data, such as fMRI images.

In comparison to 2D CNNs, convolutions are used to extract features from spatial dimensions only. 3D CNNs, on the other hand, are capable of computing features from both spatial and temporal dimensions, convolving a 3D kernel to the cube formed by stacking frames together [JXYY13]. As a result, the 3D CNNs create multiples channels of information and performs convolutions in each channel. Figure 3.5 illustrates the difference between 2D and 3D architectures, where (a) shows how the 2D convolutions work in one spatial dimension while (b) is how the convolutions work in a 3D architecture.

3D CNNs introduce 3D convolution kernels increasing the number of parameters in the architecture, the training time, as well as the need for more data. However, medical image datasets are often small, which makes training 3D CNNs with volumetric data a challenge.

3.3.3 Feature Visualization

As deep neural network models progress, our understanding of how these large neural models operate has yet to improve. Neural networks are known as "black boxes" due to the difficulty to understand how any trained neural network functions because of the large number of interacting,

[3]Adapted from https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html

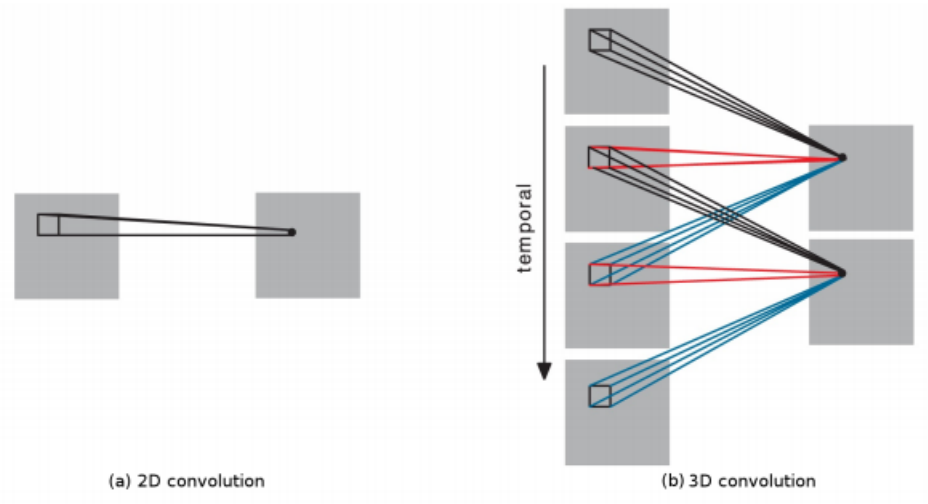

Figure 3.5 – Comparison of 2D (a) and 3D (b) convolutions. The size of convolution kernel in the temporal dimension on a 3D CNN (b) is 3, i.e., the same 3D kernel is applied to overlapping 3D cubes in the input video to extract motion features. Adapted from [LKF10].

non-linear parts. We feed in the data and then we get the output. Whatever happens in between this flow is very difficult to debug. Although we get accurate predictions, it may not be true that they are intelligent enough to perceive the same way as we do. Deeper neural networks are even harder to study because of their size and amount of parameters. One motivation for understanding what is learned by the network is to be able to improve models.

As the depth of a CNN increases, higher-level visual concepts are captured. Convolutional layers naturally retain spatial information which is then lost in fully-connected layers, therefore we can expect the last convolutional layers to have the best compromise between high-level semantics and detailed spatial information. Neurons in these layers look for semantic class-specific information in the image (say object parts). Knowing the importance of each neuron activation (feature maps) for a particular class of interest can help us better understand where the whole deep model is looking at. For example, if a neural network would predict "person" for a given image, we would ideally expect the neural network to recognize feature maps that look for hands, faces, legs, etc. as being more important than other feature maps.

One of the first techniques proposed for visualizing features is the deconvolutional network [ZF13], which gives insight into the function of intermediate feature layers and the operation of the classifier by highlighting the portions of a particular image that are responsible for the firing of each neural unit. This technique for visualizing the features learned by the hidden units of deep neural networks suggested an architectural change of smaller convolutional filters that led to state of the art performance on the ImageNet benchmark in 2013 [ZF13].

A different approach is to use the gradient to find images that cause higher activations [SVZ13] and lower activations [SZS+13] for output units. This technique is composed of a collection of: extreme pixel values, structured high frequency patterns, and copies of common motifs without global structure that cause high (or low) activations [SVZ13, SZS+13].

A more recent gradient approach technique called Grad-CAM [SCD+17] attempts to describe attribution scores using fully connected layers. The idea is, instead of trying to propagate back the gradients, to be able to infer downsampled relevance map of the input pixels from the activation maps of the final convolutional layer. The downsampled heatmap is upsampled to obtain a coarse relevance heatmap.

To explain in simple terms, we take the final convolutional feature map and then we weigh every channel in that feature with the gradient of the class with respect to the channel. This can be described as nothing but how intensely the input image activates different channels by how important each channel is with regard to the class. One of the advantages of this approach is that it does not require any re-training or change in the existing architecture.

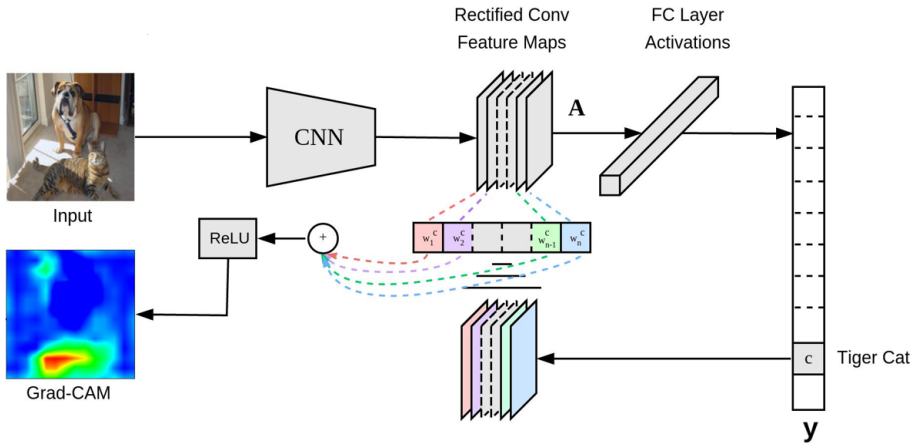

Figure 3.6 – Grad-CAM overview: Given an image and a class of interest (e.g., 'tiger cat' or any other type of differentiable output) as input, we forward propagate the image through the CNN part of the model and then through task-specific computations to obtain a raw score for the category. The gradients are set to zero for all classes except the desired class (tiger cat), which is set to 1. This signal is then backpropagated to the rectified convolutional feature maps of interest, which we combine to compute the coarse Grad-CAM localization (blue heatmap) which represents where the model has to look to make the particular decision. Adapted from [SCD+17].

3.4 Genetic Programming

Genetic Programming (GP) is an important sub-area of Evolutionary Computation (EC) described in the early 1980s, but clearly defined and established by the research of Koza [Koz92]. GP is a technique that aims at automating the construction of computer programs to perform a previous specified task from a high-level statement of a problem. Inspired by biological evolution and its mechanisms, GP follows Darwin's theory of evolution "survival of the fittest" and "genetic propagation of characteristics" principles. To evolve programs, GP uses nature-inspired genetic operations, such as crossover, mutation, reproductions, gene duplication and gene deletion. GP

may also employ developmental processes in which an embryo is transformed into a fully developed structure.

To exemplify this process, GP randomly constructs an initial population of programs using functions and terminals appropriated to the problem domain. Each program is then measured to evaluate how well it solves the given problem (fitness of the program). Next, programs are selected for reproduction by Darwin's survival of the fittest principle. Selection methods for reproduction are numerous, but for this work, we briefly explain tournament selection (TOS) [HH08]. This method is a variant of rank-based selection methods, in which individuals are randomly selected and then ranked according to their relative fitness value, selecting the fittest among them for reproduction. Genetic operations of crossover and mutation are then used to generate new offspring programs from programs selected from the current population. The crossover operator creates new individuals using two parental programs, and the mutation operator generates new offspring by altering one individual program, constituting the new population. Each of the individuals of the new population is measured for fitness, and the process is repeated for many generations. The last generation, usually, is designated as the result produced by GP and formed by the best programs.

Over the years, various GP methods emerged departing from the original premise that candidate solutions are computer programs represented by tree structures. Our interest here is in Grammar-based genetic programming (GGP) [OR01, OR03].

Grammars play an important role in structure representations in computer science, being broadly employed to syntactically limit symbolic expressions [Hop08]. Their application range from the definition of valid expressions, enforcement of type restrictions to the description of constraints of a computer language. GGP uses formal grammar, usually in Backus-Naur Form (BNF) (notation for expressing the grammar of a language in the form of production rules) to restrict the search space, incorporating the domain knowledge of the problem.

GGP explores the biological process of gene expression, introducing the concept of genotype to phenotype mapping. In general, each generated individual has a variable-length linear genotype consisting of integer codons (i.e., integer or binary lists), to which genetic operators, such as mutation and crossover, are applied. The genotype is then mapped to phenotype, a program in the specified language by the context-free grammar, to evaluate the individual's fitness. GGP does not use trees for individual representation, but as a temporary structure in the course of mapping. That is to say that instead of operating solely on solution trees, as in standard GP, GGP allows search operators to act on genotypes, partially derived phenotypes, or fully-formed phenotypic derivation trees.

Despite the popularity of CNNs, its design involves a large number of choices, where most of them are made by an expert of the domain. Such choices impact heavily on the training and performance of CNNs, e.g., decisions need to be made concerning the number of layers, type of layers, and parametrization of multiple receptive fields that are part of it as the number of filters, stride, or filter sizes.

Due to the aforementioned factors, one alternative to improve decisions made on CNNs architectures and its performance is to exploit Genetic Programming to generate CNNs automatically.

4. FEATURE VISUALIZATION OF DEEP LEARNING MODELS FOR DYSLEXIA CLASSIFICATION IN FMRI DATA

In this chapter, we develop a feature visualization approach to dyslexia classification using deep learning models. The first Section 4.1 describes two approaches to use fMRI data in deep learning models. In Section 4.2, we describe the steps for using fMRI data as input to our deep learning models, which we proceed to describe in Section 4.3 to classify subjects among dyslexics and non-dyslexics. In Section 4.4 we leverage two approaches to visualize the features that contributed to the classification of subjects between the two classes. Finally, in Section 4.5 we conclude this chapter with a discussion about visualizing features of deep learning models in dyslexia classification.

4.1 Approaches to the use of fMRI data in Deep Learning

Every fMRI application has different features that are important. Paradigms or experimental designs are organized to induce mental states of a subject through specific stimulus modalities [Bux09]. For example, in an fMRI study of dyslexia, the paradigm for a task-based fMRI is designed to activate language networks while the subject press a button to decide whether the word seen is a word or non-world as explained in Chapter 2. For this reason, activation of features related to motor control will not be valuable to the analysis. In this case, from the hundreds of thousands of voxels contained in an fMRI scan of a brain, only a small portion of all of it constitute a relevant region for a certain study. To avoid that these unnecessary observations were part of the training process in classical machine learning techniques, feature selection methods were used. With the advent of deep learning algorithms and their ability to automatically learn features at several levels of abstraction without depending entirely on human-crafted features, these feature selection methods became less used [ST16]. On the other hand, deep learning methods applied to fMRI data in its entirety can have a high computational cost. Two approaches to combine fMRI data and Deep Learning algorithms, maintaining the autonomy of the classifier and at the same time, aiming at reducing computational resources needed, are briefly described on the next subsections.

4.1.1 Whole Brain

In this method, the classifier searches for patterns in the data that are spatially distributed across the brain instead of using feature selection methods that assume a predetermined hypothesis about where those patterns are. To do so, the method uses a binary mask filling the brain volume to retrieve data from all brain regions.

Figure 4.1 – Binary mask used for extracting all the voxels that are inside the brain.

4.1.2 Parcellations

Feature selection using brain parcellation methods divide the brain into individual regions that can be used to build a network to study its structure and functionality. These parcellations are often derived from clustering algorithms applied to brain images. In other words, voxels constituting a brain region are grouped into chunks to then calculate the average activation within the group resulting in the final value of the activation of each of the parcellations.

One example of parcellations is the cc200 [CJH+12]. This parcellation is based on spatially constrained clustering algorithm using Resting-state fMRI data partitioned into 200 regions. These regions (groups) are created according to the voxels' similarity. Figure 4.2 represents the cc200 division.

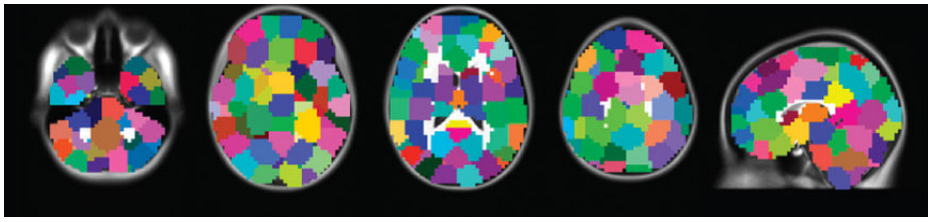

Figure 4.2 – Parcellation *cc200*.

4.2 Preprocessing fMRI data

fMRI measures and map brain activity by changes in blood flow through time [MHN+04]. The data obtained from the value of the fMRI signal (BOLD response) is a four-dimensional functional neuroimage: a sequence of three-dimensional images over a time interval, as illustrated in Figure 2.1. A typical three-dimensional brain image is noisy, requiring preprocessing the data to make statistical analysis valid and support model assumptions, as shown in Chapter 2.

We now present the steps taken to use the fMRI data from the ACERTA project as input to our deep learning model. This process stage is composed of three phases, as follows.

- Apply a binary mask to the fMRI data.
- Normalize the fMRI data.
- Use data augmentation techniques to the fMRI data.

In the first phase, the pre-processing stage is responsible for removing all outside noise from brain imaging data so we only have the brain volume from all brain regions. This stage does not require further feature selection methods to be applied to fMRI data, allowing deep learning methods to learn relevant features for a task.

Second, after applying a whole brain mask to our fMRI data, this normalization phase removes variation in the data to simplify the detection of subtle differences we are interested in.

Lastly, the third phase allows us to enhance our data so we can improve our deep learning model performance on subject classification task. This stage is called data augmentation, a technique applied to small datasets in conjunction to deep learning models [PW17], which provide us with more data to increase our deep learning model ability to generalize it. Such techniques are already employed in several image problems in deep learning models, but are still at its initial stage in fMRI data [MG18]. Therefore, it is important to highlight its valuable contribution to our work.

4.3 Deep Learning Model

Using the information obtained from the pre-processing stage, we applied two methods to analyze the fMRI data for dyslexia classification. Our motivation to use convolutional neural networks in our problem classification is to exploit visualization techniques from intermediate layers to understand which brain regions led to distinguish between dyslexics and non-dyslexics, providing insights to neuroscientists. The first method uses a 2D convolutional neural network to classify fMRI data as having or not dyslexia from a subset of the ACERTA project. The second method replaces the 2D convolutional layer with a 3D convolutional layer to evaluate who 3D convolutions affect accuracy in a relatively limited dataset.

4.3.1 2D Convolutional Neural Network

Convolutional neural networks can exploit spatially local correlation by enforcing a local connectivity pattern between neurons of adjacent layers, i.e., each neuron is connected to only a small region of the input volume. In other words, the inputs of hidden units in a layer are from a subset of units in the previous layer. This characteristic makes CNNs suitable to process bioimagery for classification tasks, since they are able to successfully capture spatial and temporal dependencies in an image through the application of filters. In our 2D convolutional neural network (2D CNN), we adopt two different approaches to generate 2D CNN architectures. We designed a typical 2D

CNN based on examples from the literature [LBB+98, KSH12]. For the second approach, we apply Genetic Programming (GP) to generate 2D CNNs individuals according to a grammar developed in order to methodically improve the accuracy of a CNN architecture without our direct design.

In order to use 2D CNNs we need to transform our data into 2D images, we do so by using conversion of the z dimension into the number of slices from this dimension. We use the med2image[1] python library for the conversion. This approach is not ideal given that we have 3D images, but since deep neural networks for 3D images have a higher computational cost and require more training data we apply this technique.

For our first approach of 2D CNNs several attempts to find a suitable architecture were made. We started with 5 convolutional layers followed by pooling layers. All convolutional layers had ReLU activations in them. After the last pooling layer we added one fully-connected layer and a dropout layer before the final fully-connected layer with the Softmax activation for the two classes. Our model did not perform well on the test data leading us to change the architecture. We tackled this problem by, at first, decreasing the number of convolutional layers. Then, changing the hyperparameters in our model, such as learning rate, number of filters in each convolutional and pooling layers. At last, we added more fully-connected and dropout layers, improving accuracy. After several iterations of different architecture topologies, hyperparameters and optimizer choices, we achieved an accuracy above 80% with our design.

Our second approach to generate 2D CNNs uses Grammar-based Genetic Programming (GGP) while optimizing hyperparameter choices in the same way as our first approach. In our grammar, we tried to be as inclusive of architecture choices as possible. Our GP-grammar start a CNN architecture with a convolutional layer, followed either by pooling or Batch Normalization layers. The activation functions for a convolutional layer are Linear and Sigmoid activation functions besides the ReLU activation we used in the first approach. Pooling layers use either Max or Average functions. The hyperparameters set used in our grammar are described in Table 5.1. Generating CNNs individuals using GGP provided us with three best architectures for classying dyslexia among subjects. We were able to achieve an accuracy above 94% with a 2D CNN topology for our problem.

We employed GP and GGP in our second approach to generate CNN architectures, however this was not the focus of our work. The focus of our work is to be able to find a suitable convolutional neural network architecture that can classify fMRI data of dyslexia with high accuracy, allowing us to provide feature visualization of gradients that contributed in the classification.

Grammar for 2D Convolutional Neural Network

In our GP approach, we use a grammar for the generation of individuals. Its definition is shown in Grammar 4.1. It represents a context-free grammar, in the Backus-Naur Form (BNF) format, where the tag $< FINAL\_EXP >$ indicates the final expression for the CNN. FC, pool, and conv refer to fully connected layer, pooling layer, and convolution layer respectively. The values

[1] Available at: https://github.com/FNNDSC/med2image

for the variables N, K, and M determine the number of layers that should be used. We restrict the value of M and N to 3 because of the computational cost to evaluate several architectures. However, these values can be expanded and adapted to solve complex problems through the generation of deeper architectures.

$$
\begin{aligned}
\langle FINAL\_EXP \rangle &::= \langle EXP\_2 \rangle \langle FC \rangle \\
\langle EXP\_2 \rangle &::= (\langle EXP\_1 \rangle * \langle M \rangle) \\
\langle EXP\_1 \rangle &::= (\langle CONV \rangle \langle POOL \rangle) \\
\langle FC \rangle &::= \text{fc} * \langle K \rangle \\
\langle POOL \rangle &::= \text{pool} \\
&\quad | \ \epsilon \\
\langle CONV \rangle &::= (\text{conv} * \langle N \rangle) \\
\langle N \rangle &::= 1 \\
&\quad | \ 2 \\
&\quad | \ 3 \\
\langle K \rangle &::= 0 \\
&\quad | \ 1 \\
&\quad | \ 2 \\
\langle M \rangle &::= 1 \\
&\quad | \ 2 \\
&\quad | \ 3
\end{aligned}
$$

Grammar 4.1 – Example of BNF grammar applied in the study.

Figure 4.3 shows the flowchart for the construction of a CNN architecture based on the grammar. The grammar allows the generation of multiple convolutional layers followed or not by a pooling layer. A Fully connected layer is also optional. An example of architecture that the grammar can produce is this genotype: (((conv*1)pool)*2)fc*2. Figure 4.4 shows the architecture created to represent this configuration. For didactic purposes, we use as example an 32×32 input to make visualization of the architecture easier to understand.

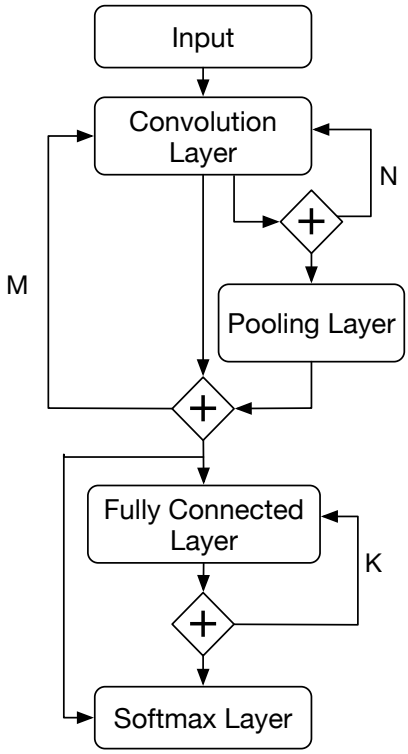

Figure 4.3 – Flowchart of possibilities for CNN topologies using the proposed grammar.

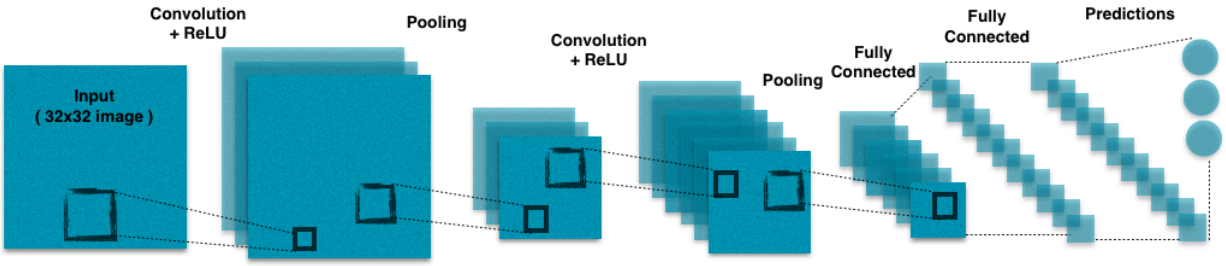

Figure 4.4 – Architecture of a CNN for the following genotype: (((conv*1)pool)*2)fc*2.

4.3.2 3D Convolutional Neural Network

3D convolutions apply a three dimensional filter to the dataset and the filter moves in all three-directions (x, y, z) to calculate the low level feature representations. Their output shape is a three dimensional volume space such as cube or cuboid. They are helpful in event detection that have temporal dimensional, such as videos, and volumetric medical images, as fMRI data.

We replaced all convolutional layers from our designed 2D CNN architecture to 3D convolutional layers. Since our data is a 3D shape, we expanded the data in the last dimension to add the channel dimension for gray images (1 channel). We did not use GGP to generate the 3D CNN topology. This model was able to achieve an accuracy of 70%, but did not outperform our best 2D CNN architectures.

4.4 Feature Visualization

Several methods for understanding and visualizing convolutional neural networks have been developed in the literature, in part as a response to criticism that the learned features in a neural network are not interpretable to humans.

Neural networks are, generally speaking, differentiable with respect to their inputs. For instance, if we want to find out what kind of input would cause a certain behavior – whether that is an internal neuron firing or the final output behavior – we can use derivatives to iteratively tweak the input towards that goal.

To achieve this goal, we adopted two approaches. The first one is rather simple, a straightforward visualization technique to show the activations of the network during the forward pass. The activations, for ReLu function, usually start out looking relatively blobby and dense, but as the training progresses the activations become more sparse and localized. One pitfall from this visualization technique is that some activation maps may be all zero for many different inputs, which can indicate dead filters. Figure 4.5 illustrates layer activations of the first convolutional layer from AlexNet [KSH12] when looking at a picture of a cat.

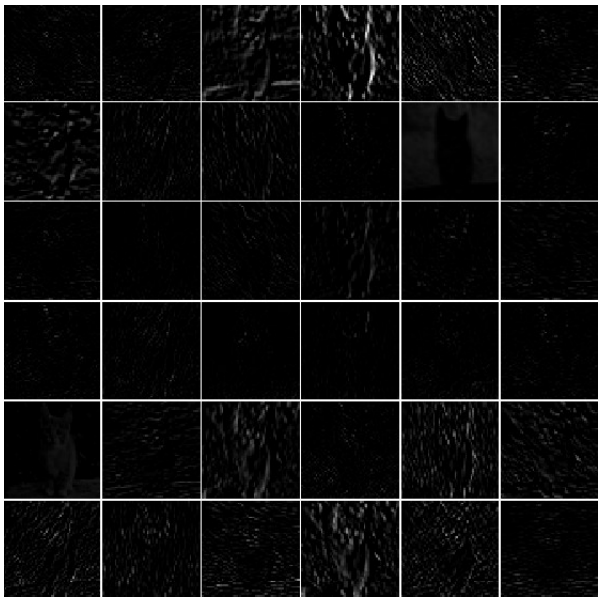

Figure 4.5 – Typical-looking activations on the first convolutional layer of a trained AlexNet [KSH12] looking at a picture of a cat [3]. Every box shows an activation map corresponding to some filter. The activations are sparse (most values are zero, in this visualization shown in black) and mostly local.

The second approach is to produce heatmaps of class activations over input images. While predicting the class labels for images, sometimes your model will predict wrong label for your class, i.e. the probability of the right label will not be maximum. In cases such as these, it is helpful to visualize which parts of the image your CNN is looking at and deducing the class labels.

[3] Adapted from http://cs231n.github.io/

A class activation heatmap is a 2D grid of scores associated with a particular output class, computed for every location for an input image, indicating how important each location is with respect to that output class. Figure 4.6 shows activations for two different classes: dog and cat.

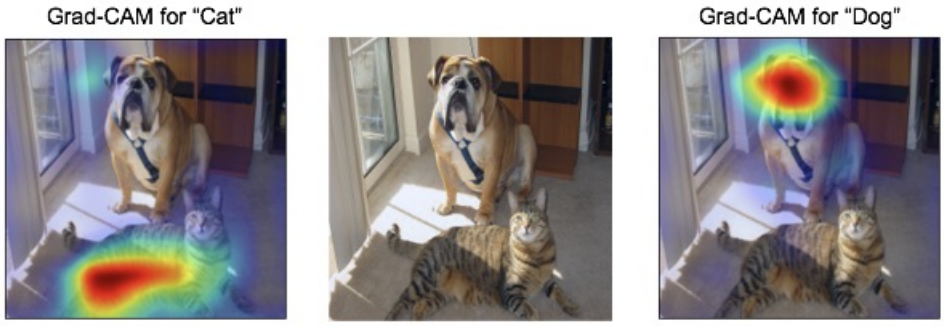

Figure 4.6 – GradCAM activations for class cat on the left side and dog on the right side when receiving the middle image as input. Adapted from [SCD+17].

The second approach provided us with a better understanding of our convolutional neural network classification for dyslexia. We were able to confirm with neuroimaging specialists how useful our technique was in terms of interpretability. The findings from our visualization technique matched the ones from a paper recently published by Buchweitz et al. [BCCT+18] on the ACERTA project about what brain regions differentiate children with dyslexia from typical readers.

4.5 Chapter Remarks

We have shown that our approach is able to provide meaningful insights to neuroimaging specialists from convolutional neural networks used in dyslexia classification of fMRI data. Even though, we showed two different approaches for feature selection our goal was to be able to achieve high accuracy without parcellations so we only applied the whole brain approach to our data. For this example, our approach accurately classify dyslexia among subjects employing deep learning models in fMRI data. Our deep learning approach and feature visualization technique is capable of dealing with high-dimensional data to produce great accuracies and better understanding of how a convolutional neural network distinguish between the two classes. The novelty of these techniques for fMRI data is important to the neuroimaging community because it not only decrease the need of human-crafted features for machine learning algorithms, but also shows that deep learning methods can be applied to this type of data and feature visualization can provide helpful insights to specialists. We describe our experiments and results in details in the next Chapter.

5. EXPERIMENTS AND RESULTS

In this chapter, we describe the experiments we conducted on fMRI data from ACERTA project. By using deep learning techniques, we aim to accurately classify dyslexic subjects from non-dyslexic, and analyze which features are more relevant in the classification. In this way, more relevant features can be useful to identify brain regions related to dyslexia. First, we present the protocol from the ACERTA project to collect the data. Followed by data augmentation techniques used to enhance the dataset to train a deep learning model. Second, we describe our classification task. This task consisted of a designed architecture, along with the neural network architecture generated by the grammar-based genetic programming approach and the model training. At last, we evaluate the feature visualization method used to help identifying brain regions from the learned classification model and discuss the relevance of the features identified by this method.

5.1 Data

5.1.1 Participants

The ACERTA study included 32 children divided into two groups, typical readers (TYP; n = 16) and dyslexic readers (DYS; n = 16). The participants were all monolingual speakers of Portuguese and right-handed. The two groups were matched for age and sex [age 7−13; mean = 9; SD = 1.39; TYP 07 female (mean age = 8.44; SD = 0.51); DYS 05 female (mean age = 9.63; SD = 0.88]. All children were participants of a reading disorders and development study, but from two different arms of the study: a longitudinal study (public schools) and a cross-sectional study (reading clinic). The 16 typical readers were part of a cohort of students from public schools enrolled in the longitudinal arm of the project; the goal of the study was to investigate the neurological markers of reading development over a 3-year period. The school children were evaluated at the end of the 2014 school year, and were scanned during the 2015 school year [BCCT$^+$18].

5.1.2 Resting-state and Word-reading task

A resting-state and a task-related paradigm were administered. 1) Resting state fMRI provides neural measurements of the functional relationship between areas of the brain when an explicit task is not being performed [SFM$^+$09]. During the scan, participants were asked to keep their eyes open and stare at a fixation crosshair for 7 min. 2) Task based fMRI examines brain regions whose activity changes from baseline in the BOLD signal in response to the performance of a task or stimulus [PD12]. The study conducted a mixed event-related experiment using a word and pseudoword reading test validated for Brazilian children [SPZT13]. The task consisted of 20 regular

words, 20 irregular words, and 20 pseudowords. The 60 stimuli were divided into two 30-item runs to give the participants a break halfway into the task. Words and pseudowords were presented on the screen one at a time for 7 seconds each. A question was presented to participants along with the each word ("Does the word exist?"), to which participants had to select "Yes" or "No" by pressing response buttons. After 10 trials (10 words) either a baseline condition or rest period was inserted in the study. The baseline condition consisted of presentation of a crosshair in the middle of the screen for 30 seconds. During the presentation of the crosshair, participants were instructed to relax and clear their minds [BCCT+18].

5.1.3 Data acquisition

Data was collected on a GE HDxT 3.0 T MRI scanner with an 8-channel head coil [BCCT+18]. The following MRI sequences were acquired: a T1 structural scan (TR/TE = 6.16/2.18 ms, isotropic 1mm^3 voxels); two task-related 5-min 26-sec functional FMRI EPI sequences; and a 7-min resting state sequence. The task and the resting-state EPI sequences used the following parameters: TR = 2000 ms, TE = 30 ms, 29 interleaved slices, slice thickness = 3.5 mm; slice gap = 0.1 mm; matrix size = 64 $\times$ 64, FOV = 220 $\times$ 220 mm, voxel size = 3.44 $\times$ 3.44 $\times$ 3.60 mm [BCCT+18].

5.1.4 Data preprocessing

The preprocessing steps for the task-based (word-reading task) and resting-state fMRI are described as follows. 1) Word-reading task: preprocessing included slice-time and motion correction, smoothing with a 6-mm FWHM Gaussian kernel, and a nonlinear spatial normalization to 3.0 x 3.0 x 3.0 mm voxel template (HaskinsPedsNL template). TR's with motion outliers (>0.9 mm) were censored from the data. The criteria for exclusion due to head motion were: excessive motion in 20% of the TRs. The average head motion for each group for the participants included in the study, in the word-reading paradigm, was: DYS M = 0.16 (SD = 0.08), TYP M = 0.18 (SD = 0.15). One participant from each group was excluded due to excessive head motion. 2) Resting-state: preprocessing for the resting-state data was equivalent to the word reading task. Additional preprocessing for the resting-state data included a nuisance regression of the six estimated motion parameters (x, y, z, roll, pitch, yaw) and time-series of the average signal of the white matter and cerebral spinal fluid, as well as signal detrending using a third order polynomial fit and a bandpass temporal filter (0.01 and 0.1 Hz) [WKG+09]. TR's with motion outliers (>0.9 mm) were censored from the data. Participants with more than 20% of the TRs exceeding the limit were excluded. The average head motion for the participants included in resting state study was: DYS M = 0.10 (SD = 0.05), TYP M = 0.15 (SD = 0.10). The average number of TRs censored for the Dyslexic Group was 9.0 ($\pm$9.2) and 12.8 ($\pm$14.0) for the typically developing group for the resting task [BCCT+18].

5.2 Data Augmentation

Models trained with small datasets tend not to generalize well data from the validation and test set, causing the models to suffer from the problem of overfitting [MK17]. To reduce overfitting, several methods have been proposed. We briefly describe three of them. The first and simplest one is to add a regularization term on the norm of the weights. The second method is dropout. Dropout works by probabilistically removing a neuron from designated layers during training or by dropping certain connection [GG16, KTW16]. The third is Batch normalization [MK17] technique, it normalizes layers allowing us to train the normalization weights.

Aiming to increase the performance of the training of a model, data augmentation techniques were proposed. Specialized image and video classification tasks often have insufficient data. This becomes specially true in medical industry, where the cost of imaging exams are high and access to data is heavily protected due to privacy concerns.

One of the best ways to improve the performance of a deep learning model is to increase the number of data points in the training set [MK17]. A solution would be to gather data from public available sites. However, this is often not possible when working with fMRI data for the lack of public available datasets to diverse diagnosis and the impact that the methods used during acquisition and preprocessing can cause on the data. To improve our model's ability to generalize and correctly label images we applied few data augmentation techniques to our data.

Firstly, we used normalization techniques to remove some variation in the data to simplify the detection of subtle differences we are interested in (e.g. the presence of a pathology). When applying normalization to voxel intensities we have two options: zero-mean, unit variance normalization standard for qualitative images (e.g. weighted brain MR images, where the contrast is highly dependent on acquisition parameters, typically set by an expert); and range normalization (e.g. normalize all voxels to [-1, 1] or [0, 1]).

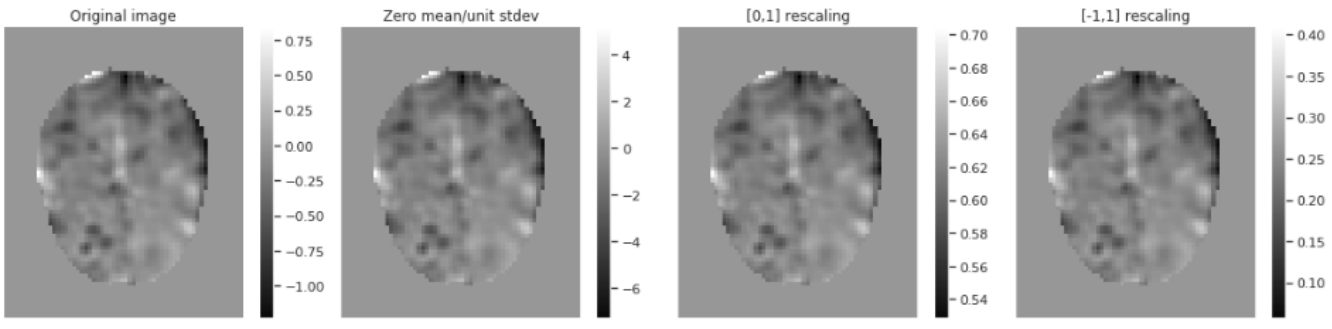

Figure 5.1 – Different intensity normalization methods on a T1-weighted brain MR image. Slice depicted is the central slice from a single subject of the ACERTA project dataset.

Secondly, we use intensity and spatial augmentations in the data. Two intensity augmentations were chosen: adding noise to training images to generalize to noisy images and adding a

random offset or contrast to handle differences between images. As for spatial augmentations, we opted for flipping the image tensor in directions on where to expect symmetry (e.g. a left/right flip on brain scans); and random deformations (e.g. for mimicking differences in organ shape).

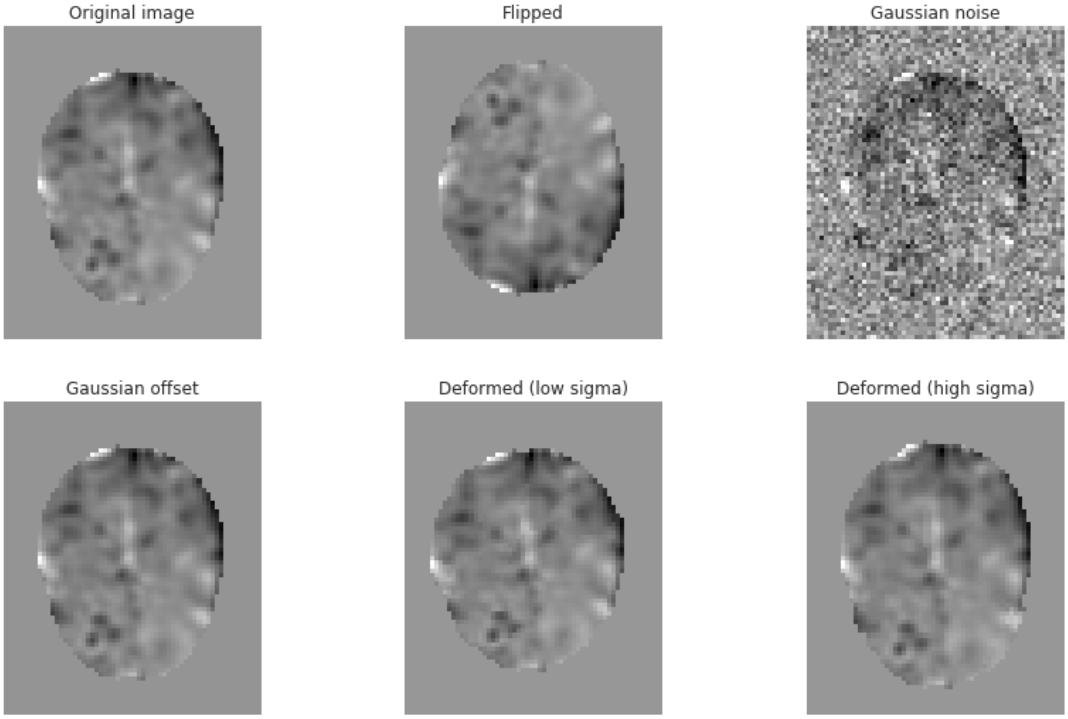

Figure 5.2 – Example of intensity and spatial augmentation techniques. Slice depicted is the central slice from a single subject of the ACERTA project dataset.

5.3 Classification Task

We use two approaches of convolutional neural network for subject classification: 2D and 3D convolutional neural networks. The CNNs receive as input the whole brain volume after the binary mask is applied to it. As for our 2D CNNs we convert the whole brain volume into 2D images.

For the first type of CNNs (2D) we employ two different approaches. The first one is to manually create an architecture able to classify the subjects between dyslexics and non-dyslexics, choosing its parameters in comparison to the accuracy achieved on a prior set of parameters. The second one is to use genetic programming, more specifically grammar-based genetic programming (GGP). In this process, a population of CNN architectures is generated, where each CNN architecture is considered an individual, and it is evaluated to produce a fitness value.

Our first approach to generate a 2D CNN architecture consisted of several attempts of fine-tuning hyperparameters. Number of convolutional layers, activation functions of convolutional layers, batch normalization after convolutional layers, max pooling, dropout layers, fully connected layers, learning rate and optimizer were continuously altered in order to achieve high accuracy on

subject classification. We were able to end up with an architecture containing approximately $175k$ parameters, which can be considered a small amount in comparison to a deeper architecture, for example, VGG-16 [SZ14] which contains over 138 million of parameters.

The network was able to achieve 81.03% accuracy on subject classification. To establish a comparison with the accuracy achieved by the CNNs architectures generated by the GGP, we split the dataset into training and test set. Leaving 80% of the 288 images after data augmentation for the training set, and the test set with the remaining 20%. The model was trained for 100 epochs and a batch size of 16. Figure 5.3 summarizes the model of the CNN network. We used Keras open source library [C+15] in the development and execution of our model.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 58, 71, 16)        8800
_________________________________________________________________
batch_normalization_1 (Batch (None, 58, 71, 16)        64
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 29, 35, 16)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 27, 33, 32)        4640
_________________________________________________________________
batch_normalization_2 (Batch (None, 27, 33, 32)        128
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 13, 16, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 11, 14, 64)        18496
_________________________________________________________________
batch_normalization_3 (Batch (None, 11, 14, 64)        256
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 5, 7, 64)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 2240)              0
_________________________________________________________________
dense_1 (Dense)              (None, 64)                143424
_________________________________________________________________
dropout_1 (Dropout)          (None, 64)                0
_________________________________________________________________
preds (Dense)                (None, 2)                 130
=================================================================
Total params: 175,938
Trainable params: 175,714
Non-trainable params: 224
```

Figure 5.3 – Model summary of the 2D convolutional neural network.

For our second approach, we apply a grammar from GGP. The code was implemented using Python programming language. To run GGP, we choose to use PonyGE2 framework [FMF+17]. The Keras open source library [C+15] was used in the development and execution of the deep neural networks.

We generate a population of CNN architectures using GGP. Every CNN was considered an individual in a population and, as so, it was evaluated to produce a fitness value. The network topology of all CNN generated based on the grammar could have different hyperparameters. Table 5.1 summarizes the hyperparameters.

This study was performed using a population of 15 individuals and 15 generations. The individuals have their fitness value calculated according to the accuracy they obtained when testing

Table 5.1 – CNN hyperparameters

Hyperparameters	Values
Kernel size	Ranging from 1 to 5
# of filters	Starts with 32; duplicate after every 2 convolutions
Stride	Ranging from 1 to 3
Max or Avg pooling shape	Ranging from 1 to 5
Learning rate	Starting at 1×10^{-5}
# of epochs	100
# of Neurons FC layer	128, 256, 512, 1024, 2048

Table 5.2 – Experimental parameters.

Parameter	Value
Number of generations	15
Population size	15
Crossover rate	75%
Mutation	Int Flip Per Codon
Mutation rate	1/Genome length
Tournament size	2
Elite size	1

the CNN on our ACERTA project dataset. For this research, greater accuracy leads to greater fitness. Parameters applied in the execution of the genetic algorithm are summarized in Table 5.2.

The selection process used is the tournament selection technique, in which two random individuals are chosen from the population, and only the one with the best fitness is selected. After the selection, the crossover step is performed by randomly picking two individuals and swapping their genetic material by applying the variable one-point technique (also known as Single Point Crossover) [SK14]. A probability of 75% is used to determine if the crossover must occur in the pair of individuals. After the crossover, every pair of parents produces a pair of children. Mutation is operated on every individual in the child population after crossover is applied. For this process, Int Flip Per Codon [FMF$^+$17] mutation is operated on the genomes and randomly mutates every individual codon with a given probability. Every population is replaced by a newly generated child population.

To perform the analysis of the network, the dataset was split into training and test set. The proportion for the training set was 80% of the 288 images after data augmentation, and test set had the 20% left. All CNNs were trained for 100 epochs with a batch size of 16.

Table 5.3 – Three best results from GGP execution.

CNN representation	Accuracy (%)
((pool*1)conv*2)fc*2	94.83
(conv*2)fc*1	93.10
(conv*5)fc*1	89.66

To evaluate the quality of the generated CNNs, we used the accuracy achieved during classification of a test sample set. This metric measures the percentage of the images correctly classified as their real class. During its execution, GGP algorithm evolved its population until it reached a level from where applying mutation in the individuals was only providing slightly better accuracies.

For this algorithm, the best accuracy value obtained was 94.83%. The three best-achieved accuracies were 94.83%, 93.10%, and 89.66%. Table 5.3 summarizes the individual genotypes that represent the best CNNs as well as their accuracy. The best CNN architecture had on its first layer an average pooling followed by a two convolutional layers, two fully-connected layers having a Softmax function in the last fully-connected layer. The depth of all CNN topologies was considerably shallower in comparison with state-of-the-art architectures for other problems in computer vision [SZ14, SLJ+15]. The last point to highlight is the fact that our grammar allows the creation of CNNs in which the convolution layer is not necessarily followed by pooling layers. All of these types of architecture were among the best results. Therefore, a deeper architecture not always results in better accuracy for our problem.

For our last proposed approach, we converted the network created based on domain expert knowledge into a 3D convolutional neural network. We, now, provide a 4-dimensional array as input to the network. We expanded our data adding one channel for gray images. In comparison to the 2D architectures, the 3D CNN had the worst accuracy on subject classification. The model achieved an accuracy of 70.69% distinguishing between dyslexics and non-dyslexics. The 3D CNN was also more prone to overfitting in the first epochs of training. Parameters used for dataset split, batch size and number of epochs were preserved in the 3D CNN.

5.3.1 Baseline Algorithm

For validating the performance of our deep learning algorithms, we compare it with a well-known machine learning technique: Support Vector Machines (SVMs) [CV95]. This technique has been used in neuroimaging studies [TVGS16, FFMB14] and it is able to perform good generalization in case of high-dimensional data and small set of training patterns. This is relevant for fMRI applications because datasets typically have many features (voxels), but only a relatively small set of subjects. While this SVM property is useful to reduce the "curse of dimensionality" problem by reducing the risk of overfitting the training data, it is still important to reduce the number of

voxels as much as possible. In machine learning, this feature reduction step is referred to as feature selection. One way of feature selection consists in restricting the number of voxels to the ones in anatomically or functionally defined regions-of-interest (ROIs).

As the purpose of our work is to be able to use deep learning algorithms to exploit their ability to automatically learn features at several levels of abstraction without depending entirely on human-crafted features, these feature selection methods will not be employed. Therefore, we opted for using whole brain data.

Our first experiment was to classify the participants in dyslexic and non-dyslexic. For that, we used the fMRI data from 32 participants (16 dyslexic and 16 typical control) after data augmentation which resulted in a total of 288 images. The input of our machine learning algorithm was 60 $\times$ 73 $\times$ 61 voxel, which corresponds to the whole brain volume and a binary mask filling the brain volume was used to retrieve data from all brain regions. This is necessary because scikit-learn algorithms only accept 2-dimensional samples $\times$ features matrices. The reduction process from 4D-images or 3D-images to feature vectors comes with the loss of spatial structure; however, it allows to discard uninformative voxels, such as the ones outside of the brain. Such voxels only carry noise and scanner artifacts, affecting the quality of the estimation.

During the training of our SVM models, we evaluated different methods of cross-validation. We report the results from splitting the data into train and test (80% train, 20% test) and leave one out cross-validation (LOOCV) for Linear SVM implemented using scikit-learn [PVG$^+$11] library in python.

The same way we did with our deep learning approaches, we applied an exhaustive search over specified parameters values for our SVM estimator [PVG$^+$11]. GridSearchCV considers all parameter combinations to search candidates in order to optimize a given estimator. Typical example of parameters that can be fed to the search include kernel functions, regularization parameter (C) among others. The best accuracy achieved with SVM was using a linear kernel.

5.3.2 Results

To evaluate the quality of all of our models for subject classification, we used the metrics: accuracy, precision, recall (sensitivity) and F1-Score achieved during classification of a test sample set. The first metric measures the percentage of the images correctly classified as their real class. The second metric is the ratio of correctly predicted positive observations to the total predicted positive observations, in our case how many dyslexic were correctly classified. High precision relates to the low false positive rate. Third metric calculates how many of the predicted positive observations are true positive. Finally, the F1-Score is the weighted average of precision and recall. It takes both false positives and false negatives into account. We present all results of the classification of the participants as having dyslexia or not in Table 5.4.

We can conclude that our deep learning approaches yield better accuracies than the traditional machine learning technique widely employed in neuroimaging problems.

Table 5.4 – Summary of the results for Dyslexia classification.

Technique	Accuracy (%)	Precision	Recall	F1-Score
SVM (80% train, 20% test)	70	0.4	1	0.57
SVM (LOOCV)	46	0.47	0.47	0.47
First GGP 2D CNN	**94.83**	0.81	0.80	0.80
Second GGP 2D CNN	**93.10**	0.79	0.78	0.77
Third GGP 2D CNN	**89.66**	0.74	0.72	0.73
Handcrafted 2D CNN	81.03	0.74	0.74	0.74
Handcrafted 3D CNN	70.69	0.71	0.71	0.70

5.4 Feature Visualization Task

In order to be able to visualize the features that are more relevant to our classification, we analyze the layer activations and the neural network gradients of our convolutional neural network.

For understanding how our CNN model is able to classify an input image, we need to understand how our model sees the input image by looking at the output of its convolutional layers. This is useful to understand how successive convolutional layers transform their input. Representations learned by convolutional layers are highly agreeable to visualization, partly because they are representations of visual concepts. Thus, visualizing layers activations consists of displaying the feature maps that are output by various convolution and pooling layers in a network. This gives a view into how an input is decomposed into the different filters learned by the network. Each channel encodes relatively independent features, so the proper way to visualize these feature maps is by independently plotting the contents of every channel as a 2D image. The Figure **??** shows us how each convolutional layer learned the features for our dyslexia classification.

The second approach is to visualize the gradients of our network. The method to visualize the features learned by the neural network chosen was Grad-CAM [SCD+17]. This method uses

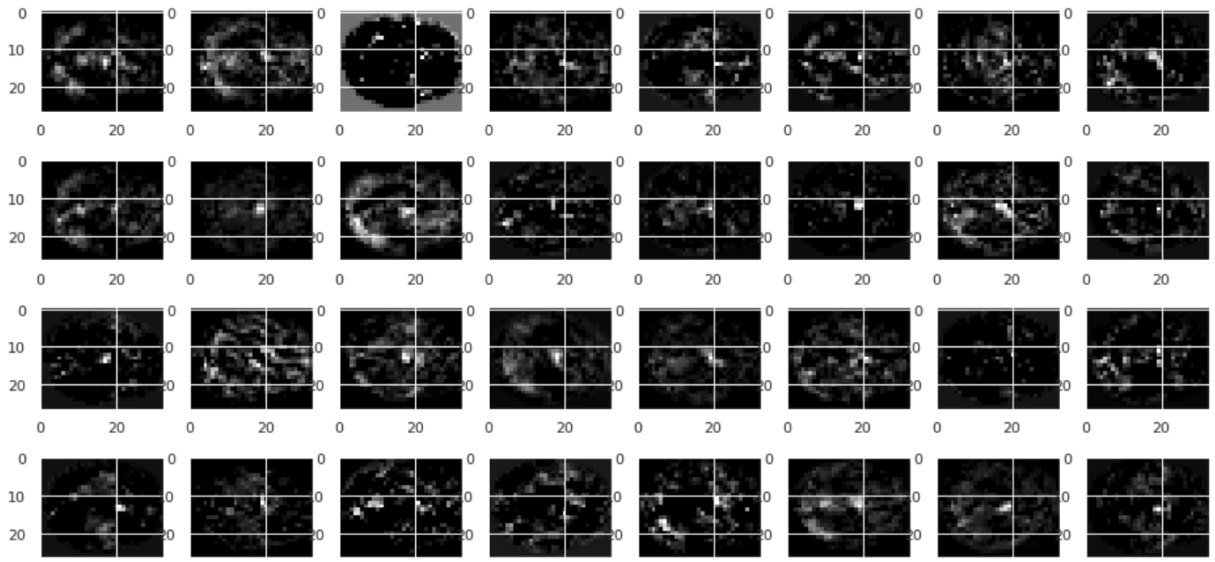

Figure 5.4 – Intermediate Convolutional layer activations for each of the filters for Dyslexia classification. Slice depicted is the central slice of the brain volume from ACERTA dataset.

the gradients of any target concept flowing into the final convolutional layer to produce a coarse localization map highlighting the important regions in the image for predicting a concept.

To obtain the class-discriminative localization map, Grad-CAM computes the gradient of y^{c} (score for class c) with respect to feature maps A^{k} of a convolutional layer, i.e. $\frac{\partial y^c}{\partial A^k_{ij}}$. These gradients flowing back are global-average-pooled to obtain the importance weights α^c_k:

$$\alpha^c_k = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A^k_{ij}}}_{\text{gradients via backpropagation}} \tag{5.1}$$

The weight α^c_k represents a *partial linearization* of the deep network downstream from A and it captures the importance of feature map k for a target class c.

$$S^c = \frac{1}{Z}\sum_i\sum_j\sum_k \frac{\partial y^c}{\partial A^k_{ij}} A^k_{ij} \tag{5.2}$$

In other words, we take the final convolutional feature map and weigh every channel in that feature with the gradient of the class with respect to the channel. This represents how intensely the input image activates different channels by how important each channel is regard to the class. The spatial score of a specific class S^{c} is the global average pooling over two dimensions i and j for the gradient of the respective class output y^{c} with respect to the feature map A^k_{ij}. After that, we multiply the resulting value with the feature map along its channel axis k. The result is average/poolover the channel dimmension k. At the end we have the spatial score map of size $i \times j$.

After the training of our convolutional neural network model, we load the model with the best accuracy to visualize the learned gradients. The purpose of it is to be able to provide insights from the imaging data to specialists in neuroimaging, demystifying the black box model for which deep learning algorithms are known.

To evaluate the method we chose a pair of subjects as input, one dyslexic and one non-dyslexic participant. The images generated when applying the method to our participants with respect to the gradients learned by the network model are shown in Figures 5.5 and 5.6 respectively. Figure 5.5 represents the model activation for classification of dyslexic participants. And Figure 5.6 represents model activation for classification of non-dyslexic participants. It is possible to observe that areas with higher activations range from yellow to red, which means these areas had a great impact on the model classification of subjects.

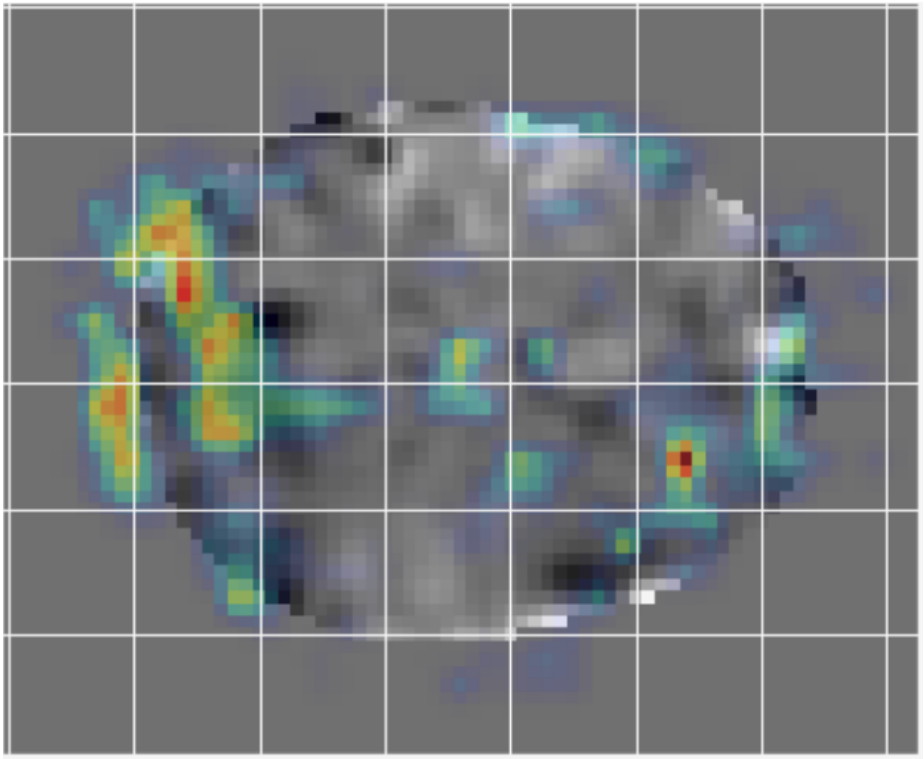

Figure 5.5 – Class activation mapping for dyslexic participants classification. Slice depicted is the central slice of the brain volume from ACERTA project dataset.

In a recent study published by Buchweitz et al. [BCCT+18], their goal is to investigate intrinsic and reading-related brain function associated with dyslexia and typical readers from the ACERTA project. According to the authors, the results show (a) underconnectivity between the occipitotemporal region (visual word form area) and the brain's default-mode network in dyslexic readers and (b) more activation of the anterior cingulate cortex for typical readers relative to dyslexic readers. Their findings provide evidence for brain connectivity and function differences in an underrepresented population in fMRI studies of dyslexia. And also suggests atypical intrinsic function, and differences in directed attention processes in dyslexia.

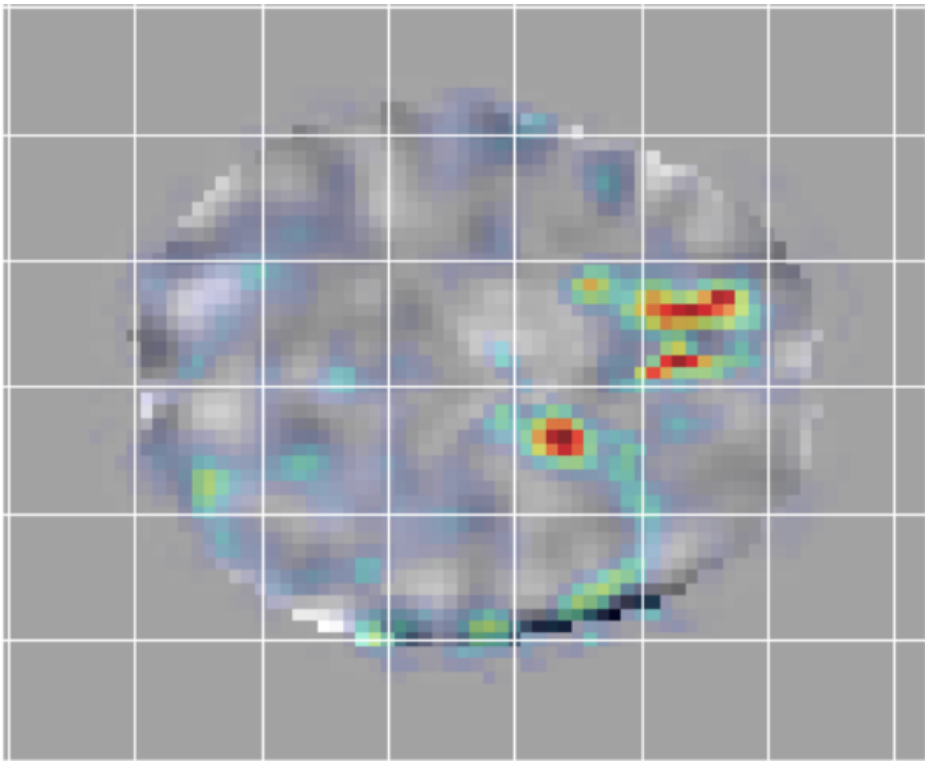

Figure 5.6 – Class activation mapping for non-dyslexic participants classification. Slice depicted is the central slice of the brain volume from ACERTA project dataset.

The findings published by the authors corroborates with the results of our feature visualization technique. We are able to identify a bigger activation of the occipitotemporal region present in the dyslexic subject and more activation of portion of the Anterior Cingulate Cortex (ACC) in typical readers (non-dyslexic subject). Therefore, our results demonstrate that feature visualization techniques are capable of bringing a better understanding of deep learning algorithms applied to neuroimaging data.

6. RELATED WORK

Medical Imaging analysis is a challenging task due to the large amount of information one image can contain and the size of the datasets. Deep Learning algorithms perform better in big datasets decreasing their chance for overfitting the data [GBC16]. However, medical imaging datasets from fMRI experiments are usually small and have fewer training samples in comparison to learnable parameters taken by the network. In order to help this task, some researches [PHS+14, KCSL16, LZS+14, ST16, RZC+17, TVGS16] proposed different techniques to automate (in some level) this process. The resume of this researches are shown in Table 6.1.

The quality and the amount of the data is important to reach good results with a neural network. All the datasets used by the researches were different because they had different goals. Most of them applied their approaches in fMRI datasets. Among them [KCSL16], and [ST16] used a specific kind of fMRI, the resting-state fMRI. During this type of scan the patient is asked to rest and try not to think about anything in particular. Three of the papers, [PHS+14], [TVGS16], and [LZS+14], used MRI data which provides a detailed image of brain tissues (its structure), in place of fMRI. Only one paper used more than one type of data. Li et al. [LZS+14] did not solely use MRI data, but also PET-Scan (Positron-emission tomography) data.

All related work implemented Deep Neural Networks (DNNs) for their tasks, except for one. [PHS+14] implemented a Deep Belief Network with 3 depths, having an architecture of 50-50-100 hidden units in the first, second and the top layer respectively. This approach used stacking learning units with an external learning algorithm to train one layer at a time. [LZS+14] and [RZC+17] implemented a 3D CNN. These networks learn the data by taking into account the previous feature. The first, [LZS+14] implemented an architecture that received one volumetric MR patch as input and provided a volumetric PET (Positron-emission tomography) patch as output [LZS+14]. The second, [RZC+17] adopted a framework called VoxNet, responsible for training convolutional neural networks from a large dataset of hundreds of thousands of available brain network volume maps, and then applying on testing samples for network classification and recognition. [KCSL16] did not report details about the network architecture, but they used a stacked autoencoder to initialize the weights, parcellations to select brain regions of interest, pre-trained the network and added hidden layers. [ST16] employed LeNet-5 as their architecure to classify patients with Alzheimer. At last, [TVGS16] applied SVM, a traditional machine learning approach, to classify young adults with and without dyslexia.

Most of the papers applied their algorithms for subject classification. Despite that, they used this to identify different diseases. [PHS+14] tried to identify patients with Huntington, this is a degenerative disease that causes the degeneration of neurons in certain areas of the brain. [KCSL16] worked with data of patients with Schizophrenia, their goal was identify the atypical fully connectivity patterns associated with the disease. The only one that did not implement a neural network for classification was [LZS+14]. In their work, they aimed not to identify patients with certain disease, but predict and integrate multiple-modality neuroimaging data. As a result, they designed

a 3D CNN architecture that received one volumetric MR patch as input and another volumetric PET (Positron-emission tomography) patch as output. [ST16] implemented a classifier using resting-state fMRI data for Alzheimer; this disease is a neurological, irreversible, progressive brain disorder and multifaceted disease that slowly destroys brain cells. [RZC+17] used whole-brain fMRI data to reconstruct interacting functional brain networks. [TVGS16] aimed to investigate whether individuals with and without dyslexia could be reliably classified based on anatomical differences using T1-weighted magnetic resonance images of grey matter from 22 students with dyslexia and 27 students without dyslexia. They employed SVM and cross-validation using whole-brain classification to examine neuro-anatomical networks involved in dyslexia determining which voxels were involved in the correct classification. Their work also intended to test the reliability of their trained classifier in an independent sample of 876 young adults.

The achieved results and final conclusions from the papers show that the DNN approaches are capable of achieving competitive results using MRI and fMRI data. [PHS+14] did not report their results, but believed deep learning had a great potential in neuroimaging applications. [KCSL16] deep network approach obtained a competitive error rate of 14.2% in comparison to 22.3% error rate of traditional machine learning method – SVM. They were also able to identify atypical functional connectivity patterns by training the weights used by the network. [LZS+14] demonstrated their network was able to predict and estimate the PET data given an MRI data as input and evaluated the proposed data completion method quantitatively by comparing the classification results based on true and predicted PET images. [ST16] network architecture achieved an accuracy of 96.85% when classifying their data. The method of [RZC+17] was able to handle noisy patterns in the dataset, showing high accuracy in their predictions and potential application of CNNs in fMRI data. [TVGS16] was the only work that chose to use a traditional machine learning approach to classify subjects among dyslexics and non-dyslexics. They report the prediction accuracy of dyslexia from anatomical scans of the small set (22 dyslexic and 27 non-dyslexic subjects) and the independent set (876 subjects). In the first set, they were able to achieve an overall accuracy of 80%. However, their classifier was not able to fully generalize to the independent set causing the overall accuracy to drop to 59% and the accuracy of dyslexic subjects to 43%. The authors concluded that various predictive values showed that their anatomical classifier was far away from use in clinical settings pointing out that the areas that contributed to the classification of students with and without dyslexia were helpful to better understand the brain anatomy in dyslexia.

Despite of that, to the best of our knowledge, none tackles the visualization of learned features from deep learning algorithms used in the classification of functional MRI or MRI data.

Table 6.1 – Related Work

Paper	**Dataset**	**Data**	**Approach**	**Task**
Plis et al. [PHS+14]	PREDICT-HD project[1]	MRI data	Deep Belief Network	Subject classification for Huntington Disease
Kim et al. [KCSL16]	Neuroimaging Informatics Tools and Resources Clearinghouse (NITRC)[2]	resting-state functional connectivity pattern (fMRI data)	DNN	Subject classification for Schizophrenia
Li et al. [LZS+14]	Alzheimer's Disease Neuroimaging Initiative (ADNI)[3]	MRI and PET-Scan data	3D CNN	Data Conversion
Sarraf et al. [ST16]	ADNI[3]	resting-state fMRI data	LeNet-5	Subject classification for Alzheimer
Ren et al. [RZC+17]	Human Connectome Project (HCP)	Functional brain networks from whole-brain fMRI data	3D CNN framework (VoxNet)	Intrinsic Connectivity Networks Classification
Tamboer et al. [TVGS16]	Non-disclosed dataset	Anatomical images from whole-brain (MRI)	SVM	Subject classification for Dyslexia

[1] Dataset: https://predict-hd.lab.uiowa.edu/

[2] fcon_1000.projects.nitrc.org/indi/retro/cobre.htm

[3] adni.loni.usc.edu

7. CONCLUSION

In this work, we contribute in finding patterns in task-based fMRI to diagnose the cognitive states associated with dyslexia. By creating diagnose models able to classify participants between dyslexics and non-dyslexics in a small fMRI dataset, we are able to avoid the use of feature selection methods. To accomplish this, we have applied deep learning techniques to automatically extract relevant features from fMRI data at several leves of abstraction without depending entirely on human-crafted features to point out regions of interest. In the experiments, we compared our technique with commonly used techniques in neuroscience and the results show that deep learning models, such as convolutional neural networks, achieve greater accuracies in high dimensional data. Thus, employing deep learning algorithms to our data we outperform SVM by achieving an accuracy of 94.83% with the classifier.

By using feature visualization techniques we are able to provide a better understanding of which regions of the brain contributed to the classification of a given class of participants. Feature visualization is able to create a more transparent comprehension of deep learning models by neuroimaging specialists as to adding valuable insights to their findings. Therefore, visualization techniques allow neuroscientists and researchers to visualize brain regions that had more influence in a classification task without the need to understand the complexity of a deep learning model.

As future work, we plan to improve the accuracy on our dataset employing different neural network architectures. We aim to use transfer learning techniques in public task-based fMRI datasets to aid the extraction of relevant features for a given discriminant task. We intend to employ different feature visualization techniques to help neuroimaging specialists better understand classification of deep learning models. We believe that the mixture of different imaging techniques may reveal new insights for neurodevelopmental diagnosis.

BIBLIOGRAPHY

[Alp14] Alpaydin, E. "Introduction to machine learning". MIT press, 2014.

[B+09] Bengio, Y.; et al.. "Learning deep architectures for ai", *Foundations and trends® in Machine Learning*, vol. 2–1, 2009, pp. 1–127.

[BCCT+18] Buchweitz, A.; Corrêa Costa, A.; Toazza, R.; Bassoa, A.; Metsavaht Cara, V.; Bianchini, N.; Aguzzoli, C.; Gregolim, B.; Fernando Dresch, L.; Dorigatti Soldatelli, M.; Dacosta, J.; Wetters Portuguez, M.; Franco, A. "Decoupling of the occipitotemporal cortex and the brain's default-mode network in dyslexia and a role for the cingulate cortex in good readers: A brain imaging study of brazilian children", *Developmental Neuropsychology*, 02 2018, pp. 1–12.

[BCV13] Bengio, Y.; Courville, A.; Vincent, P. "Representation learning: A review and new perspectives", *IEEE transactions on pattern analysis and machine intelligence*, vol. 35–8, 2013, pp. 1798–1828.

[BGC15] Bengio, Y.; Goodfellow, I. J.; Courville, A. "Deep learning", *Nature*, vol. 521, 2015, pp. 436–444.

[Bur98] Burges, C. J. "A tutorial on support vector machines for pattern recognition", *Data Mining and Knowledge Discovery*, vol. 2–2, Jun 1998, pp. 121–167.

[Bux09] Buxton, R. B. "Introduction to functional magnetic resonance imaging: principles and techniques". Cambridge university press, 2009.

[C+15] Chollet, F.; et al.. "Keras", 2015.

[CGLdlTC05] Casanova, P. F.; García-Linares, M. C.; de la Torre, M. J.; Carpio, M. d. l. V. "Influence of family and socio-demographic variables on students with low academic achievement", *Educational psychology*, vol. 25–4, 2005, pp. 423–435.

[CJH+12] Craddock, R. C.; James, G. A.; Holtzheimer, P. E.; Hu, X. P.; Mayberg, H. S. "A whole brain fmri atlas generated via spatially constrained spectral clustering", *Human brain mapping*, vol. 33–8, 2012, pp. 1914–1928.

[CV95] Cortes, C.; Vapnik, V. "Support-vector networks", *Machine learning*, vol. 20–3, 1995, pp. 273–297.

[FFMB14] Franco, A.; Froehlich, C.; Meneguzzi, F.; Buchweitz, A. "Classifying brain states for cognitive tasks: a functional mri study in children with reading impairments". In: XXIV Brazilian Congress on Biomedical Engineering? CBEB 2014, 2014, Brasil., 2014.

[FLGC11] Faceli, K.; Lorena, A. C.; Gama, J.; Carvalho, A. "Inteligência artificial: Uma abordagem de aprendizado de máquina", *Rio de Janeiro: LTC*, vol. 2, 2011, pp. 192.

[FMF+17] Fenton, M.; McDermott, J.; Fagan, D.; Forstenlechner, S.; Hemberg, E.; O'Neill, M. "Ponyge2: Grammatical evolution in python". In: Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2017, pp. 1194–1201.

[FSH+14] Finn, E. S.; Shen, X.; Holahan, J. M.; Scheinost, D.; Lacadie, C.; Papademetris, X.; Shaywitz, S. E.; Shaywitz, B. A.; Constable, R. T. "Disruption of functional networks in dyslexia: a whole-brain, data-driven analysis of connectivity", *Biological psychiatry*, vol. 76–5, 2014, pp. 397–404.

[GBC16] Goodfellow, I.; Bengio, Y.; Courville, A. "Deep learning". MIT press, 2016.

[GG16] Gal, Y.; Ghahramani, Z. "A theoretically grounded application of dropout in recurrent neural networks". In: Advances in neural information processing systems, 2016, pp. 1019–1027.

[GK79] Galaburda, A. M.; Kemper, T. L. "Cytoarchitectonic abnormalities in developmental dyslexia: a case study", *Annals of neurology*, vol. 6–2, 1979, pp. 94–100.

[Glo11] Glover, G. H. "Overview of functional magnetic resonance imaging", *Neurosurgery Clinics of North America*, vol. 22–2, 2011, pp. 133–139.

[GSR+85] Galaburda, A. M.; Sherman, G. F.; Rosen, G. D.; Aboitiz, F.; Geschwind, N. "Developmental dyslexia: four consecutive patients with cortical anomalies", *Annals of neurology*, vol. 18–2, 1985, pp. 222–233.

[Hay94] Haykin, S. "Neural networks: a comprehensive foundation". Prentice Hall PTR, 1994.

[HF+11] Handler, S. M.; Fierson, W. M.; et al.. "Learning disabilities, dyslexia, and vision", *Pediatrics*, vol. 127–3, 2011, pp. e818–e856.

[HH08] Hingee, K.; Hutter, M. "Equivalence of probabilistic tournament and polynomial ranking selection". In: Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on, 2008, pp. 564–571.

[HMB+11] Hoeft, F.; McCandliss, B. D.; Black, J. M.; Gantman, A.; Zakerani, N.; Hulme, C.; Lyytinen, H.; Whitfield-Gabrieli, S.; Glover, G. H.; Reiss, A. L.; et al.. "Neural systems predicting long-term outcome in dyslexia", *Proceedings of the National Academy of Sciences*, vol. 108–1, 2011, pp. 361–366.

[Hop08] Hopcroft, J. E. "Introduction to automata theory, languages, and computation". Pearson Education India, 2008.

[HS06] Hinton, G. E.; Salakhutdinov, R. R. "Reducing the dimensionality of data with neural networks", *science*, vol. 313–5786, 2006, pp. 504–507.

[HSM04] Huettel, S. A.; Song, A. W.; McCarthy, G. "Functional magnetic resonance imaging". Sinauer Associates Sunderland, 2004, vol. 1.

[HW62] Hubel, D. H.; Wiesel, T. N. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex", *The Journal of physiology*, vol. 160–1, 1962, pp. 106–154.

[IGL+11] Illán, I.; Górriz, J.; López, M.; Ramírez, J.; Salas-Gonzalez, D.; Segovia, F.; Chaves, R.; Puntonet, C. G. "Computer aided diagnosis of alzheimer's disease using component based svm", *Applied Soft Computing*, vol. 11–2, 2011, pp. 2376–2382.

[JXYY13] Ji, S.; Xu, W.; Yang, M.; Yu, K. "3d convolutional neural networks for human action recognition", *IEEE transactions on pattern analysis and machine intelligence*, vol. 35–1, 2013, pp. 221–231.

[KCSL16] Kim, J.; Calhoun, V. D.; Shim, E.; Lee, J.-H. "Deep neural network with weight sparsity control and pre-training extracts hierarchical features and enhances classification performance: Evidence from whole-brain resting-state functional connectivity patterns of schizophrenia", *NeuroImage*, vol. 124, 2016, pp. 127 – 146.

[Koz92] Koza, J. R. "Genetic programming: on the programming of computers by means of natural selection". MIT Press, 1992, vol. 1.

[KSH12] Krizhevsky, A.; Sutskever, I.; Hinton, G. E. "Imagenet classification with deep convolutional neural networks". In: Advances in neural information processing systems, 2012, pp. 1097–1105.

[KSW15] Krug, K.; Salzman, C. D.; Waddell, S. "Understanding the brain by controlling neural activity", 2015.

[KTW16] Kubo, Y.; Tucker, G.; Wiesler, S. "Compacting neural network classifiers via dropout training", *arXiv preprint arXiv:1611.06148*, 2016.

[LBB+98] LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P.; et al.. "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86–11, 1998, pp. 2278–2324.

[LBD+89] LeCun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W.; Jackel, L. D. "Backpropagation applied to handwritten zip code recognition", *Neural computation*, vol. 1–4, 1989, pp. 541–551.

[LKF10] LeCun, Y.; Kavukcuoglu, K.; Farabet, C. "Convolutional networks and applications in vision". In: Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on, 2010, pp. 253–256.

[LSS03] Lyon, G. R.; Shaywitz, S. E.; Shaywitz, B. A. "A definition of dyslexia", *Annals of dyslexia*, vol. 53–1, 2003, pp. 1–14.

[LZS+14] Li, R.; Zhang, W.; Suk, H.-I.; Wang, L.; Li, J.; Shen, D.; Ji, S. "Deep learning based imaging data completion for improved brain disease diagnosis". In: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014, Golland, P.; Hata, N.; Barillot, C.; Hornegger, J.; Howe, R. (Editors), 2014, pp. 305–312.

[MG18] Mikołajczyk, A.; Grochowski, M. "Data augmentation for improving deep learning in image classification problem". In: 2018 International Interdisciplinary PhD Workshop (IIPhDW), 2018, pp. 117–122.

[MHN+04] Mitchell, T. M.; Hutchinson, R.; Niculescu, R. S.; Pereira, F.; Wang, X.; Just, M.; Newman, S. "Learning to decode cognitive states from brain images", *Machine learning*, vol. 57–1, 2004, pp. 145–175.

[Mit97] Mitchell, T. M. "Machine learning. 1997", *Burr Ridge, IL: McGraw Hill*, vol. 45–37, 1997, pp. 870–877.

[MK17] Ma, Y.; Klabjan, D. "Convergence analysis of batch normalization for deep neural nets", *arXiv preprint arXiv:1705.08011*, 2017.

[Mor96] Morgan, W. P. "A case of congenital word blindness", *British medical journal*, vol. 2–1871, 1896, pp. 1378.

[Mur12] Murphy, K. P. "Machine learning: a probabilistic perspective". MIT press, 2012.

[OL90] Ogawa, S.; Lee, T.-M. "Magnetic resonance imaging of blood vessels at high fields: in vivo and in vitro measurements and image simulation", *Magnetic resonance in medicine*, vol. 16–1, 1990, pp. 9–18.

[OLKT90] Ogawa, S.; Lee, T.-M.; Kay, A. R.; Tank, D. W. "Brain magnetic resonance imaging with contrast dependent on blood oxygenation", *Proceedings of the National Academy of Sciences*, vol. 87–24, 1990, pp. 9868–9872.

[OR01] O'Neill, M.; Ryan, C. "Grammatical evolution", *IEEE Transactions on Evolutionary Computation*, vol. 5–4, 2001, pp. 349–358.

[OR03] O'Neil, M.; Ryan, C. "Grammatical evolution". In: *Grammatical Evolution*, Springer, 2003, pp. 33–47.

[PBL18] "Matrizes de referência - alfabetização e letramento inicial". Acessado: 2018, Capturado em: http://inep.gov.br/matrizes-de-referencia1, 2018.

[PBM18] "Matrizes de referência - alfabetização da matemática inicial". Acessado: 2018, Capturado em: http://inep.gov.br/matrizes-de-referencia1, 2018.

[PD12] Petersen, S. E.; Dubis, J. W. "The mixed block/event-related design", *Neuroimage*, vol. 62–2, 2012, pp. 1177–1184.

[Pen47] Penfield, W. "Ferrier lecture: some observations on the cerebral cortex of man", *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 1947, pp. 329–347.

[PHS+14] Plis, S. M.; Hjelm, D. R.; Salakhutdinov, R.; Allen, E. A.; Bockholt, H. J.; Long, J. D.; Johnson, H. J.; Paulsen, J. S.; Turner, J. A.; Calhoun, V. D. "Deep learning for neuroimaging: a validation study", *Frontiers in Neuroscience*, vol. 8, 2014, pp. 229.

[PVG+11] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. "Scikit-learn: Machine learning in Python", *Journal of Machine Learning Research*, vol. 12, 2011, pp. 2825–2830.

[PW17] Perez, L.; Wang, J. "The effectiveness of data augmentation in image classification using deep learning", *arXiv preprint arXiv:1712.04621*, 2017.

[RBDH00] Robichon, F.; Bouchard, P.; Démonet, J.-F.; Habib, M. "Developmental dyslexia: re-evaluation of the corpus callosum in male adults", *European Neurology*, vol. 43–4, 2000, pp. 233–237.

[RH98] Robichon, F.; Habib, M. "Abnormal callosal morphology in male adult dyslexics: Relationships to handedness and phonological abilities", *Brain and language*, vol. 62–1, 1998, pp. 127–146.

[RN10] Russell, S.; Norvig, P. "Artificial Intelligence: A Modern Approach". Prentice Hall, 2010, third ed..

[Ros09] Rose, J. "Identifying and teaching children and young people with dyslexia and literacy difficulties: an independent report", 2009.

[RPF03] Ramus, F.; Pidgeon, E.; Frith, U. "The relationship between motor control and phonology in dyslexic children", *Journal of Child Psychology and Psychiatry*, vol. 44–5, 2003, pp. 712–722.

[RZC+17] Ren, D.; Zhao, Y.; Chen, H.; Dong, Q.; Lv, J.; Liu, T. "3-d functional brain network classification using convolutional neural networks". In: 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017), 2017, pp. 1217–1221.

[SCD+17] Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. "Grad-cam: Visual explanations from deep networks via gradient-based localization". In: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 618–626.

[SFM+09] Smith, S. M.; Fox, P. T.; Miller, K. L.; Glahn, D. C.; Fox, P. M.; Mackay, C. E.; Filippini, N.; Watkins, K. E.; Toro, R.; Laird, A. R.; Beckmann, C. F. "Correspondence of the brain's functional architecture during activation and rest", *Proceedings of the National Academy of Sciences*, vol. 106–31, 2009, pp. 13040–13045, http://www.pnas.org/content/106/31/13040.full.pdf.

[SK14] Soni, N.; Kumar, T. "Study of various crossover operators in genetic algorithms", *International Journal of Computer Science and Information Technologies*, vol. 5–6, 2014, pp. 7235–7238.

[SLJ+15] Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. "Going deeper with convolutions". In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[SPZT13] Salles, J. F. d.; Piccolo, L. d. R.; Zamo, R. d. S.; Toazza, R. "Normas de desempenho em tarefa de leitura de palavras/pseudopalavras isoladas (lpi) para crianças de 1º ano a 7º ano", *Estudos e Pesquisas em Psicologia*, vol. 13–2, 2013, pp. 397–419.

[ST16] Sarraf, S.; Tofighi, G. "Classification of alzheimer's disease using fmri data and deep learning convolutional neural networks", *CoRR*, vol. abs/1603.08631, 2016, 1603.08631.

[SVZ13] Simonyan, K.; Vedaldi, A.; Zisserman, A. "Deep inside convolutional networks: Visualising image classification models and saliency maps", *arXiv preprint arXiv:1312.6034*, 2013.

[SZ14] Simonyan, K.; Zisserman, A. "Very deep convolutional networks for large-scale image recognition", *arXiv preprint arXiv:1409.1556*, 2014.

[SZS+13] Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. "Intriguing properties of neural networks", *arXiv preprint arXiv:1312.6199*, 2013.

[TLB16] Teixeira, M. T.; Limberger, B. K.; Buchweitz, A. "O desempenho de crianças em fase de alfabetização em avaliações de leitura e escrita", *Estudos Linguísticos (São Paulo. 1978)*, vol. 45–2, 2016, pp. 595–610.

[Tor98] Torgesen, J. "Catch them before they fall: Identification and assessment to prevent reading failure in young children (on-line)", *National Institute of Child Health and Human Development. Available: ldonline. org. ld_indepth/reading/torgesen_catchthem. html*, 1998, pp. 1–15.

[TVGS16] Tamboer, P.; Vorst, H.; Ghebreab, S.; Scholte, H. "Machine learning and dyslexia: Classification of individual structural neuro-imaging scans of students with and without dyslexia", *NeuroImage: Clinical*, vol. 11, 2016, pp. 508–514.

[WBB+15] Wolfers, T.; Buitelaar, J. K.; Beckmann, C. F.; Franke, B.; Marquand, A. F. "From estimating activation locality to predicting disorder: a review of pattern recognition for neuroimaging-based psychiatric diagnostics", *Neuroscience & Biobehavioral Reviews*, vol. 57, 2015, pp. 328–349.

[WKG+09] Weissenbacher, A.; Kasess, C.; Gerstl, F.; Lanzenberger, R.; Moser, E.; Windischberger, C. "Correlations and anticorrelations in resting-state functional connectivity mri: a quantitative comparison of preprocessing strategies", *Neuroimage*, vol. 47–4, 2009, pp. 1408–1416.

[ZF13] Zeiler, M. D.; Fergus, R. "Visualizing and understanding convolutional networks", *CoRR*, vol. abs/1311.2901, 2013, 1311.2901.