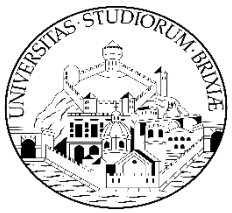



UNIVERSITÀ
DEGLI STUDI
DI BRESCIA

DOTTORATO DI RICERCA IN INTELLIGENZA ARTIFICIALE IN MEDICINA E INNOVAZIONE NELLA RICERCA CLINICA E METODOLOGICA

ING-INF/03

CICLO
XXXVI

# RESEARCHING THE POTENTIAL OF ARTIFICIAL INTELLIGENCE TO SUPPORT THE UNDERSTANDING OF NEUROLOGICAL DISEASES: THE CASES FOR FRONTOTEMPORAL LOBAR DEGENERATION DETECTION AND MICE ULTRASONIC COMMUNICATION ANALYSIS

DOTTORANDA: Tatiana Pilipenko

SUPERVISORE: Prof. Riccardo Leonardi

I would like to dedicate this thesis to my husband Gianluca, our cat Freya, and my friend J. who always helps me to find the right words.

# Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 20,000 words including bibliography, footnotes, tables and equations and has 23 figures.

Tatiana Pilipenko
02/01/2024

# Acknowledgements

And I would like to acknowledge my supervisor Prof. Riccardo Leonardi, my tutor Alessandro Gnutti, Prof. Ivan Serina, Prof.ssa Sara Anna Bonini, Marika Premoli and everyone who provided guidance and help during this academic journey.

# Abstract

This doctoral research aims to explore the intersections and synergies between machine learning and neurology. The study's core is comprised of three distinct but interconnected investigations that highlight the potential of artificial intelligence (AI) in enhancing our understanding of neurodegenerative diseases and animal communication, as well as improving diagnostic methods.

The first part of this research investigates the application of machine learning techniques in the diagnosis of Fronto-Temporal Dementia using Magnetic Resonance Imaging data. The study primarily uses a Multi-Voxel Pattern Analysis approach with Support Vector Machine and Random Forest algorithms for analysis. This section aims to address the challenges in the early detection of neurodegenerative diseases, providing medical professionals with a supportive diagnostic tool that could potentially improve treatment outcomes.

The second part of the research delves into the analysis of ultrasonic vocalizations (USVs) in mice, specifically focusing on the changes in ultrasonic communication patterns in mice treated with Cannabis sativa oil as compared to control mice. The investigation utilizes specialized recording equipment and dedicated software to analyze USVs, shedding light on the nuances of animal communication. This segment discusses the disparities in ultrasonic communication patterns between the two groups, correlating them with specific behaviors, presenting a comprehensive statistical exploration.

The third part of the study presents a data processing pipeline to analyze mouse audio data, employing advanced signal processing techniques and machine learning. This segment introduces a meticulous data labeling system that assigns each audio segment to one of eight behavioral categories. Spectrogram computations are used to visualize distinct acoustic characteristics of high-frequency mouse vocalizations. Deep learning experiments undertaken in this part aim to uncover insights into the neural mechanisms underlying mouse ultrasonic vocalizations, thereby enriching our understanding of the intricate relationship between ultrasonic vocalization and behavior.

# Riassunto

Questa ricerca di dottorato mira a esplorare le intersezioni e le sinergie tra l'apprendimento automatico e la neurologia. Il nucleo dello studio è costituito da tre indagini distinte ma interconnesse che mettono in evidenza il potenziale dell'intelligenza artificiale (IA) nel migliorare la nostra comprensione delle malattie neurodegenerative e della comunicazione animale, nonché nel migliorare i metodi di diagnosi.

La prima parte di questa ricerca studia l'applicazione di tecniche di apprendimento automatico nella diagnosi della demenza frontotemporale utilizzando dati di risonanza magnetica. Lo studio utilizza principalmente un approccio di analisi a pattern multivoxel con algoritmi di Support Vector Machine e Random Forest per l'analisi. Questa sezione mira a affrontare le sfide nella rilevazione precoce delle malattie neurodegenerative, fornendo ai professionisti medici uno strumento diagnostico di supporto che potrebbe migliorare potenzialmente gli esiti del trattamento.

La seconda parte della ricerca approfondisce l'analisi delle vocalizzazioni ultrasoniche nei topi, concentrandosi in particolare sulle modifiche nei modelli di comunicazione ultrasonica nei topi trattati con olio di Cannabis sativa rispetto ai topi di controllo. L'indagine utilizza attrezzature di registrazione specializzate e software dedicati per analizzare le vocalizzazioni ultrasoniche, facendo chiarezza sulle sfumature della comunicazione animale. Questo segmento discute delle disparità nei modelli di comunicazione ultrasonica tra i due gruppi, correlandoli a comportamenti specifici, presentando un'esaustiva esplorazione statistica.

La terza parte dello studio presenta un flusso di elaborazione dei dati per analizzare le registrazioni audio dei topi, impiegando avanzate tecniche di elaborazione del segnale e di apprendimento automatico. Questo segmento introduce un meticoloso sistema di etichettatura dei dati che assegna ciascun segmento audio a una delle otto categorie comportamentali presenti nel dataset. I calcoli degli spettrogrammi vengono utilizzati per visualizzare le caratteristiche acustiche delle vocalizzazioni ad alta frequenza dei topi. Gli esperimenti di deep learning intrapresi in questa parte mirano a scoprire informazioni sui meccanismi neurali alla base delle vocalizzazioni ultrasoniche dei topi, arricchendo così la nostra comprensione della complessa relazione tra vocalizzazione e comportamento.

# Table of contents

**List of figures** **xiii**

**List of tables** **xv**

**Nomenclature** **xvii**

**1 Introduction** **1**
1.1 Research framework and background . . . . . 1
1.2 Goals . . . . . 2

**2 Literature review** **3**
2.1 MRI feature-based detection of frontotemporal lobar degeneration . . . . . 3
2.2 Ultrasonic vocalization in mice: definition and general information . . . . . 4
2.3 Machine learning and deep learning: general overview and application in medicine . . . . . 6
2.4 Using ML and DL for studying USV patterns . . . . . 8
2.5 Cannabidiol and its use in modern pharmacology . . . . . 9

**3 Materials and methods** **13**
3.1 Machine learning techniques for MRI feature-based detection of frontotemporal lobar degeneration . . . . . 13
3.1.1 Dataset and features . . . . . 13
3.1.2 Proposed classification methods . . . . . 14
3.1.3 Pre-processing operations . . . . . 16
3.1.4 Description of the experiments . . . . . 17
3.2 Analysis of CS and VH mice ultrasonic communication and their behavioral structure . . . . . 18
3.2.1 Animals . . . . . 18
3.2.2 USV and behavioral recordings . . . . . 18

3.2.3 Statistical analysis . . . . . 20
3.3 Data preparation and automatic behavior classification . . . . . 22
3.3.1 Data preprocessing and dataset preparation for automatic behavior classification based on USV patterns . . . . . 22
3.3.2 Preliminary behavior classification algorithms trials . . . . . 23

**4 Results and discussion** **29**
4.1 Evaluating the accuracy of machine learning algorithms for detecting frontotemporal lobar degeneration . . . . . 29
4.1.1 Further evaluation . . . . . 32
4.1.2 Further experiments . . . . . 34
4.2 Ultrasonic vocalizations in CS and VH mice and their correlation with the behavior . . . . . 34
4.3 USVs dataset preparation and preliminary automatic behavior classification results . . . . . 39
4.3.1 Dataset preparation . . . . . 39
4.3.2 CNN: 8 convolutional layers . . . . . 43
4.3.3 CNN: 10 convolutional layers . . . . . 50

**5 Conclusion** **59**
5.1 Evaluating the accuracy of machine learning algorithms for detecting frontotemporal lobar degeneration: conclusion . . . . . 59
5.2 Ultrasonic vocalizations in mice and correlation with their behavior: conclusion 59
5.3 Dataset preparation and preliminary machine learning results: conclusion . 60

**References** **63**

# List of figures


2.1 Artificial Intelligence, Machine Learning and Deep Learning relationships [1] 6
2.2 2D Structure of Cannabidiol [2] . . . . . 10

3.1 An example of fMRI image for each normalization technique. On the left the class FTD-GNR+, on the right the class oHC. . . . . 15
3.2 Data extraction and spectrogram generation scheme . . . . . 19
3.3 USVs classifications by Scattoni et al. [3] . . . . . 20
3.4 First version of the model structure with 8 convolutional layers . . . . . 26
3.5 Visual representation of 8 convolutional layers neural network. . . . . 26
3.6 Second version of the model structure with 10 convolutional layers . . . . . 28
3.7 Visual representation of 10 convolutional layers neural network. . . . . 28

4.1 Performance of SVM and RF in terms of accuracy using SSS. . . . . 30
4.2 Comparison with the work by Premi et al. [4] using LOOCV. . . . . 33
4.3 Distributions of variable-size patterns of syllables for CS and VH. . . . . 36
4.4 Distributions of syllables. Comparison between CS and VH for each behavior. 37
4.5 One-hot encoding scheme for the behavioral labeling. . . . . 41
4.6 A box plot representation of the sound duration distribution among all mice across different behaviors. . . . . 41
4.7 Gray-scaled inverted spectrogram visualization of a mouse ultrasonic vocalization chunk. The high-contrast grayscale representation enhances the visibility of acoustic features in the vocalization. . . . . 42
4.8 Example of a spectrogram from .png images dataset created with Matplotlib library . . . . . 43
4.9 Model summary of a 8 convolutional layers network . . . . . 45
4.10 Epoch timeline of a 8 convolutional layers network . . . . . 49
4.11 8 layers CNN confusion matrix. . . . . 50
4.12 Model summary of a 10 convolutional layers network . . . . . 53

4.13 Epoch timeline of a 10 convolutional layers network . . . . . 56
4.14 10 layers CNN confusion matrix. . . . . 58

# List of tables

4.1 Recap of SVM (top) and RF (bottom) accuracy for each fMRI normalization technique for the distinct feature standardization methods. The mean and standard deviation are also reported. . . . . 32
4.2 Comparison with the work by Premi et al. [4] using LOOCV. In bold the highest accuracy returned by our method with VMHC. . . . . 34
4.3 Abbreviation corresponding to each syllable. . . . . 35
4.4 Cosine similarity between CS and VH distributions assessed at varying of the pattern lengths and behaviors. Higher values indicate a stronger similarity. 38
4.5 Jensen-Shannon divergence between CS and VH distributions assessed at varying of the pattern lengths and behaviors. Lower values indicate a stronger similarity. . . . . 39

# Nomenclature

**Acronyms / Abbreviations**

ACM Anemia Control Model

AI Artificial Intelligence

ALFF Amplitude of Low Frequency Fluctuations

CBD Cannabidiol

CNN Convolutional Neural Network

CoS Cosine Similarity

DC Degree Centrality

DL Deep Learning

fALFF Fractional Amplitude of Low Frequency Fluctuations

FFT Fast Fourier Transform

fMRI functional MRI

FTD-GNR+ FTD patients carrying GRN Thr272fs mutation

FTD Frontotemporal Dementia

IUPAC International Union of Pure and Applied Chemistry

JSD Jensen–Shannon divergence

LOOCV Leave-One-Out Cross-Validation

LR Logistic Regression

ML Machine Learning

MMS Min-Max Scaler

MRI Magnetic Resonance Imaging

oHC elderly healthy subjects

PCA Principal Component Analysis

ReHo Regional Homogeneity

RF Random Forest

RS Robust Scaler

SS Standard Scaler

SSS Stratified Shuffle Split

STFT Short-Time Fourier Transform

SVM Support Vector Machine

THC Tetrahydrocannabinol

USV Ultrasonic Vocalization

VMHC Voxel-Mirrored Homotopic Connectivity

# Chapter 1

# Introduction

## 1.1 Research framework and background

This dissertation explores the intersection of artificial intelligence (AI) and neurological sciences, focusing specifically on the application of machine learning techniques for medical diagnostics and behavioral analysis. The research was conducted under the auspices of the Department of Information Engineering and the Department of Clinical and Experimental Sciences at the University of Brescia.

The first year of the research project involved the development and implementation of machine learning techniques for the detection of frontotemporal lobar degeneration in brain MRI scans. This work was carried out in collaboration with the Centre for Ageing Brain and Neurodegenerative Disorders (University of Brescia), which provided critical clinical insights into the disease and its representation in MRIs.

In the second year, the research shifted to the analysis of mice ultrasonic communication, with a particular emphasis on exploring the correlation between their communication structures and social behaviors. This involved extensive statistical analysis and the development of robust methodologies for interpreting and categorizing ultrasonic vocalizations.

The third year represented a continuation of the second year's work, focusing on refining the statistical analysis and preparing the data for further machine learning analysis. The ultimate aim of this research phase was to develop AI models capable of accurately predicting social behaviors based on patterns in ultrasonic communication.

## 1.2 Goals

The overarching objectives of this PhD thesis in Artificial Intelligence in Medicine are as follows:

- To investigate the potential of machine learning techniques in enhancing the detection and diagnosis of frontotemporal lobar degeneration using MRI scans. This aim encompasses the development and testing of algorithms capable of identifying distinctive features related to the disease in brain images.
- To explore the structure of two mice groups (control group VH vs. Cannabis sativa oil-administrated CS) ultrasonic communication through rigorous statistical analysis, focusing particularly on the categorization of various vocalization patterns and their inherent variations. A key objective is to discern any substantial correlations between these vocalization characteristics and the mice's social behaviors. Unraveling these associations can potentially facilitate the prediction of social behaviors based on vocalization attributes. The ultimate goal of this analysis is to assess whether mice possess different communication strategies by taking into account different behaviors.
- The output of the previous phase of the analysis serves as an input to the last layer of this study. Specifically, we researched a potential relationship between USVs and behaviors. Our idea is based on machine learning techniques aiming to accurately predict social behaviors in mice and identify patterns and features in their ultrasonic communication. Ultimately, this endeavor seeks to provide a novel tool for behavioral analysis applicable in a wide range of research contexts within the biomedical field.

# Chapter 2

# Literature review

## 2.1 MRI feature-based detection of frontotemporal lobar degeneration

Frontotemporal Dementia (FTD) is a neurodegenerative disorder that is believed to hit 60,000 people only in the United States per year [5]. FTD has distinct clinical phenotypes associated with multiple neuropathologic entities, and it encompasses clinical disorders that include changes in behaviour, language, executive control and often motor symptoms [6]. FTD has no cure and, as of today, there exist only supportive treatments that are able to slow down its course. Therefore, it is crucial to detect the presence of this neuropathology at its early stage of development in order to improve the quality of life of patients and their families [7].

Magnetic resonance imaging (MRI) is a well-established method for FTD diagnosis that can give a full picture of the neurodegenerative condition in a patient. The important role of MRI is not only recognised in its use to diagnose, but also in its application to exclude other pathologies [8]. Furthermore, the progress made in the magnetic resonance methodology allowed the possibility of FTD detection in its early stages. In general, a medical expert observes MRI data and evaluates the patient's condition [9]. However, this operation is time-consuming and may be subject to personal interpretation. As a matter of fact, the study of methods designed to automatically elaborate the information contained in MR images is nowadays a very active research field.

During the first year of PhD, we investigated the performance of some artificial intelligence (AI) algorithms for the automatic classification of brain images associated with FTD carriers and healthy subjects, of which Support Vector Machine (SVM) [10] and Random Forest (RF) [11] techniques have returned the best results.

## 2.2 Ultrasonic vocalization in mice: definition and general information

Sounds are an important tool for social communication not only for humans but also for other mammals, mainly speaking - rodents. We will discuss mice, their ultrasonic vocalizations (USV), and the relationship between USV and various behaviors throughout this assignment.

USVs are inaudible to human hearing due to the extremely high sound frequencies they produce (between 30 and 90 kHz or higher) [12, 13] and the fact that, in adult animals, they are often produced during close contact between animals [14]. Mice use USV in different social contexts, for example, awake male and awake female, awake male and anesthetized female, two awake males, etc. [15, 16]. Looking at spectrograms of mice sounds, it is possible to distinguish "stretches of positive power" (USV) and "zero power" (silence) [17]. Those spectrograms are composed of **phrases** (also called **songs**); phrases, in turn, consist of **syllables**. There are several different classifications of syllables; in this research we took as the reference the classification described in the work of Scattoni et al. [3].

In recent years, the exploration of USVs in mice has garnered significant scientific interest. This research avenue allows for investigating correlations between distinct behavioral traits and various factors such as specific rodent populations, disease conditions, or the impact of pharmacological interventions. Von Merten et al. [18], comparing two groups of domestic mice captured in France and Germany (they used three types of social context: different population - different sex, same population - same sex, and same population - different sex), says that there is a strong correlation between the number of songs mice emitted in different social contexts. Also, mouse identity influences the number of songs. Different sex – same population is the social context during which mouse produces more USV than during another two social contexts. Captured in Germany mice seemed to produce more sounds, but this effect was found statistically insignificant. In the same sex – same population situation, there was an influence of sex: female mice in this context produced more songs.

Mouse identity also influences the syllable parameters (together with the influence of social context). There are some differences in ending sequences related to population and sex. However, the difference at the beginning of the sentence was not significant.

There is a strong correlation between a number of songs and social context, particularly among the German population. Female mice spoke less if their female partner was sexually receptive, pregnant, or aged. The feeding status had an influence on the number of produced USV (well-fed mice produced more USV). There was no influence of social familiarity, even if some of the previous studies did find some.

The number of songs between populations did not differ in any social context. However, the number of syllables differed in the same population – different sex context: here, captured in France mice had a higher syllable rate than these captured in Germany. Structural parameters of syllables were different for every social context: mice from Germany tend to use longer syllables and syllables with turns, the latter being a vocal pattern that includes both ascending and descending notes within a single syllable. Conversely, mice from France produced more syllables per second, using more jump-types, abrupt, disjointed shifts in the frequency of vocalizations within a given time frame.

During same sex situations, females tend to communicate more between each other than males. This observation can be explained by their social behavior (for a female mouse, it is more common to live among other females than for a male among other males).

Wild mice do not randomly sequence their syllables in the sentence but follow a complex temporal system (syntax).

Another example can be shown by Chabout et al. [15]: here the authors explored the speech deficit related to the Foxp2 transcription factor mutation in the KE family [19]. As a result, scientists observed that even if mice do not have a complex vocal behavior (in comparison with humans), mutation of Foxp2 can affect vocalization sequencing; for example, there may be a reduction in the length and complexity of syllable sequences. Social context also can impact vocal plasticity: females produce USV, but less than males. Moreover, the skill of producing sounds increases with age (older pups more effortlessly switch between syllables).

Last but not least, Sangiamo et al. [16] studied individual mouse's USV behavior and discovered that different vocalization patterns arise as male mice engage in certain social behaviors. Dominating mice were more likely to use a variety of verbal cues than mice who avoided social contacts. Furthermore, vocal expression patterns influenced the behavior of the socially engaged partner but not the behavior of the other animals in the cage.

The examination of social behavior, specifically in the realm of ultrasonic communication, is integral to the advancement of modern natural sciences and medicine. This investigative approach can facilitate progress in our understanding and foster development in specific fields such as physiology and pharmacology. In addition, it can make significant contributions to a range of disciplines, including but not limited to, neuroscience, behavioral studies, and bioacoustics.

## 2.3 Machine learning and deep learning: general overview and application in medicine

The term machine learning (ML) refers, in general, to the study and development of computer algorithms that can improve themselves through experience by learning relevant information on a given dataset (from hereafter, the training dataset) and testing their efficiency and effectiveness on an out-of-sample new dataset (from hereafter, the test dataset).

ML has an interesting subset of techniques, the so-called deep learning (DL) (Fig. 2.1). As Chollet [1] brilliantly puts it: "[Deep learning is] a new take on learning representations from data that puts an emphasis on learning successive layers of increasingly meaningful representations. The 'deep' in 'deep learning' isn't a reference to any kind of deeper understanding achieved by the approach; rather, it stands for this idea of successive layers of representations", meaning that the algorithms rely on a number of layers of training (i.e., the depth) in order to achieve a good-quality forecast with the aim to validate or dispute a diagnosis. Deep learning is gaining popularity among data scientists because, as of today, these algorithms are those with a higher resemblance to the actual human brain functioning, with stratified processes of training and the creation of so-called "neurons" in neural networks.

While this technology is highly promising [20], the main goal of the research work should not be the complete passive substitution of actual professionals in the field but, rather, to effectively help them have a rapid and efficient output of behavioral experiments, in order to avoid the so-called false positive cases or, even worse for a medical diagnosis, false negative ones [21].

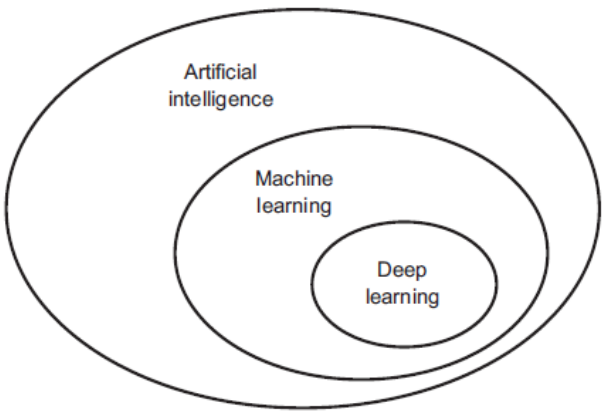



Fig. 2.1 Artificial Intelligence, Machine Learning and Deep Learning relationships [1]

Over the last years the field of artificial intelligence in medicine has experienced a phase of rapid growth. In the following paragraphs, we will review some articles regarding AI applications in different medical specializations.

In nephrology, an international group of researchers studied the impact of an artificial intelligence decision support system, named the Anemia Control Model (ACM) [22], on anemia outcomes in hemodialysis patients. The study found that, in the observation phase, median darbepoetin consumption decreased, whereas on-target hemoglobin values significantly increased. As a result, the ACM support helped improve anemia outcomes in hemodialysis patients. In other words, ACM provides a more accurate feedback control loop closer to physiological regulation, which in turn creates a buffer period for dose adjustment and reduces undesirable prescription patterns. Therefore, the ACM is more likely to provide a stable and accurate treatment for anemia patients not only in the field of nephrology but also with other diagnoses [23].

Another interesting example of AI in medicine could be found in anesthesiology. A group of scientists from California, USA, [24] developed an algorithm that could be useful in surgery. Their work discusses a machine learning algorithm that can predict hypotension in surgical patients. The algorithm was designed with two data sets: a retrospective cohort for training and a prospective local hospital cohort for external validation. The algorithm successfully predicted arterial hypotension with a sensitivity and specificity of respectively 88% and 87% 15 minutes before a hypotensive event. The results demonstrate that the machine learning algorithm can be trained to predict hypotension in surgical patients' records.

Using machine learning algorithms can potentially improve the prediction of readmissions in patients with heart failure. However, further research is needed to determine the most effective method for this purpose. A scientific group from Yale University [25] found that the Telemonitoring to Improve Heart Failure Outcomes trial provides valuable data that can be used to compare the effectiveness of different machine learning models for predicting 30-day and 180-day all-cause readmissions and readmissions due specifically to heart failure. RF and boosting combined hierarchically with SVM or logistic regression (LR) all outperformed traditional LR in terms of predictive power. This suggests that machine learning methods can be highly effective in predicting readmission rates after hospitalization for heart failure.

Probably, as of today, the most promising field for AI in medicine is radiology. The use of artificial intelligence in medical imaging, particularly for thoracic imaging, is a rapidly growing field. Deep learning methods, particularly convolutional neural networks (CNNs), have demonstrated substantial performance gains over classic machine learning techniques. There are many potential applications for AI in thoracic imaging, including evaluating lung nodules [26, 27], detecting tuberculosis [28] or pneumonia [29], and quantifying diffuse lung

diseases [30]. However, before these tools can be routinely used in clinical practice, they must be validated through prospective studies.

Ultimately, the most compelling area of interest for applying AI in medicine within the scope of this research is drug discovery. The incorporation of artificial intelligence in the process of drug discovery proves to be a formidable instrument for the synthesis planning of molecules [31–33]. Moreover, its applications for predicting physicochemical and absorption, distribution, metabolism, excretion, and toxicity have become increasingly common, demonstrating successful outcomes [34–36]. The utilization of artificial intelligence in *de novo* design has further resulted in the creation of novel, biologically active molecules possessing desired properties [37–39]. This technology is persistently advancing and expanding its reach, and its potential impact on the field of drug discovery holds immense promise.

## 2.4 Using ML and DL for studying USV patterns

USVs serve a vital function in various species, notably among rodents, allowing them to communicate over long distances and significantly influencing their mating and territorial behaviors. With regard to technological advancements, the ability to detect and record ultrasonic vocalizations is indeed within our grasp. What remains a challenge, however, are the requisite tools for automated analysis of these USVs, a component that is critical for furthering our understanding and interpretation of these sounds.

The use of ML and DL offers many advantages for studying mouse populations and their behavior patterns. First, ML and DL can help automate the identification of individual mice in a population [40]. This is important for understanding the behavior of a group, as individual mice may behave differently than the group as a whole. Additionally, ML and DL can be used to track the movements of individual mice over time, which can help identify patterns in their behavior [41]. Finally, ML and DL can help to identify correlations between mouse behavior and various environmental factors, such as food availability or temperature [42, 43]. This information can help researchers understand how the environment affects mouse behavior and develop strategies to optimize mouse populations and experiment conditions for specific purposes.

Currently, the state of research on USVs and ML/DL is still in its early stages. However, there has been some progress made in this area. For example, Coffey and colleagues from the University of Washington (USA) [44] trained a machine to detect, classify, and analyze different types of USVs.Their software, named DeepSqueak, can detect and analyze USVs automatically, rapidly, and reliably. It uses a regional convolutional neural network architecture (Faster-RCNN) to achieve these goals. DeepSqueak was designed to be accessible

to non-experts yet also flexible and adaptable. It has a graphical user interface and offers numerous input and analysis features.

Compared to other modern programs and manual analysis, DeepSqueak was able to reduce false positives, increase detection recall, dramatically reduce analysis time, optimize automatic syllable classification, and perform automated syntax analysis on arbitrarily large numbers of syllables. All of these improvements were made while maintaining manual selection review and supervised classification.

DeepSqueak, among other available software, provides a tool for integrating USV recording and analysis into existing rodent behavioral procedures. This tool has the potential to expose a wealth of innate responses, offering valuable behavioral insights when utilized in conjunction with conventional outcome measures. It is, however, important to note that, as with any tool, there may be limitations that need to be acknowledged.

However, there are several potential drawbacks to using machine learning and deep learning for studying animal populations. It is possible that machine learning algorithms can sometimes produce inaccurate results, as they are based on patterns that may be different from those used by humans. For example, the algorithm created by Valetta et al. [45] mistook a wildebeest for a tree, based on cues that are different from those used by humans. The same group of authors [45] remarked that despite not knowing exactly how a machine is differentiating between classes if validation procedures are done correctly, predictive models can still be used reliably. This is true as long as the data are representative of the population. Finally, Liu et al. [46] discussed how high-profile failures in machine learning can be attributed to the fact that even experts can fall victim to common pitfalls in data-driven analytics. These pitfalls include mistakenly inferring causation from correlation and being misled by some confounding factor or biased data sets.

Mice USVs have been extensively studied in the past few years as a measure of stress level. Studies of USVs in mice have shown that they can be used to signal anxiety, stress, and fear [47, 48]. In humans, these emotions are all related, so this information can be used to help us develop new treatments for anxiety and stress disorders.

Overall, the current state of research on USVs and ML/DL is promising. However, there is still much work that needs to be done in order to fully understand the potential of this research.

## 2.5 Cannabidiol and its use in modern pharmacology

Cannabidiol, or CBD, is one of the most talked about and studied cannabinoids found in cannabis. It has been shown to have a wide range of medical benefits, from helping to relieve

chronic pain [49] to reducing anxiety [50]. CBD is characterized by its non-psychoactive properties, thus, it does not provoke the psychoactive response commonly associated with tetrahydrocannabinol (THC). This distinct attribute has led to an increased interest in CBD, particularly for those seeking the potential therapeutic advantages of cannabis derivatives while simultaneously avoiding the cognitive alterations induced by THC. CBD is typically extracted from hemp [51], a type of cannabis containing minimal THC.

CBD is characterized as a terpenophenolic compound, comprised of twenty-one carbon atoms and possessing the chemical formula $C_{21}H_{30}O_2$. This molecular formula corresponds to a molecular mass of 314.464 g/mol (Fig. 2.2) [2]. The precise nomenclature of cannabidiol, as per International Union of Pure and Applied Chemistry (IUPAC) standards, is 2-[1R-3-methyl-6R-(1methylethenyl)-2-cyclohexen-1yl]-5-pentyl-1,3-benzenediol, a detail that was established in 1963 [52]. A cyclohexene ring, a phenolic ring, and a pentyl side chain make up the CBD molecule. In addition, the planes of the terpenic and aromatic rings are almost perpendicular to one another. The four known CBD side chain homologs are methyl, n-propyl, n-butyl, and n-pentyl. All known CBD compounds are trans in positions 1R and 6R [53]. The mechanism of action for CBD is not fully understood, but it is believed to involve multiple mechanisms of action, including interaction with adenosine receptors [54], activation of 5-HT1A receptors [55], and induction of apoptosis [56].

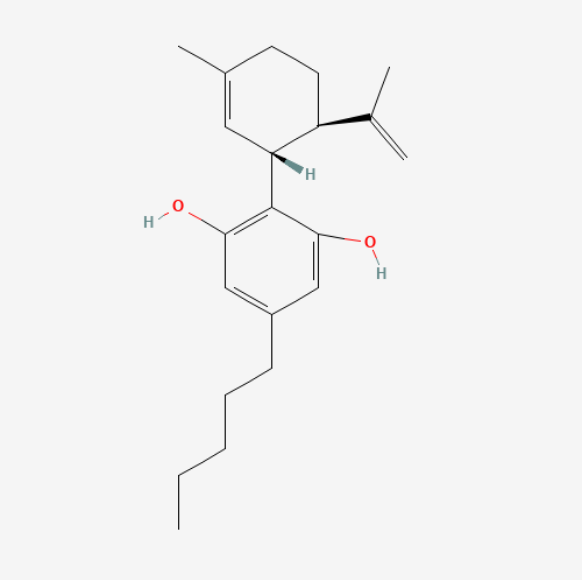



Fig. 2.2 2D Structure of Cannabidiol [2]

A growing body of scientific evidence suggests that CBD could help treat a variety of conditions, including:

- Depression [57]
- Insomnia [58]
- Post-traumatic stress disorder [59]
- Epilepsy [60]
- Parkinson's disease [61]
- Alzheimer's disease [62]
- Multiple sclerosis [63]

Side effects are rare and tend to be mild, such as dry mouth and fatigue [64]. Even though CBD is non-addictive, it could interact with other medications and cause side effects [65]. While more research is needed to confirm these findings, CBD is generally considered safe to use.

CBD oil is typically taken orally, either in drops [66] or capsules [67]. When taking CBD, starting with a low dose and increasing gradually as needed is essential. It is also important to choose a reputable brand that offers CBD oil in a variety of strengths.

The United States Food and Drug Administration has recently approved a CBD oral solution, Epidiolex, for managing two rare forms of epilepsy, "Dravet syndrome" and "Lennox-Gastaut syndrome," in pediatric patients. This new medication is expected to help improve the quality of life for many affected children [68].

In the European Union, CBD is not currently regulated under a harmonized law. Medicinal products containing CBD (such as Epidiolex and Sativex) are authorized in many EU countries and, in some cases, are reimbursed by the national health insurance system [69, 70]. CBD products placed on the market in a Member State must have a Community marketing authorization from the European Agency for the Evaluation of Medicinal Products [71].

Currently, in Italy, there are no laws that ban CBD, and at the moment, it is not yet registered as a medicinal product. Legally, this means that hemp shops can sell CBD products without restriction, but the substance cannot be used as a pharmacological treatment [72].

# Chapter 3

# Materials and methods

## 3.1 Machine learning techniques for MRI feature-based detection of frontotemporal lobar degeneration

### 3.1.1 Dataset and features

The dataset of interest was acquired at the Centre for Ageing Brain and Neurodegenerative Disorders, University of Brescia (Brescia, Italy), and consists of MRI data on 30 subjects, divided in 2 classes: 14 patients with FTD carrying GRN Thr272fs mutation (FTD-GNR+) [1] and 16 elderly gender-matched healthy subjects (oHC).

Different normalization techniques have been applied to the MRI dataset, such as Amplitude of Low Frequency Fluctuations (ALFF), Fractional Amplitude of Low Frequency Fluctuations (fALFF), Degree Centrality (DC), Regional Homogeneity (ReHo) and Voxel-Mirrored Homotopic Connectivity (VMHC), which will be introduced in the following. Fig. 3.1 shows an example of functional MRI (fMRI) image for each normalization technique and for each class.

The ALFF and fALFF measures are related to each other: ALFF is a measure that highlights, within specific regions, the physiological states of a resting brain and its spontaneous neural activity, whereas fALFF is a measure used in order to reduce the sensitivity of ALFF to physiological noises that the brain may experience such as hunger, fatigue, and pain [74]. VMHC provides an index of functional symmetry of a resting-state brain activity [75], while DC is a measure that extracts information from the single nodes composing the whole-brain

[1] Progranulin is a glycoprotein, encoded in humans by the GRN gene, located on the 17q21 chromosome. A mutation of this gene, named Thr272fs, is responsible for frontotemporal lobar degeneration cases [73]

network [76]. Finally, ReHo is the time consistency of blood oxygenation in local brain tissue [77].

### 3.1.2 Proposed classification methods

In this work, different machine learning techniques have been tested for the classification task. Since the experiments have shown that SVM and RF approaches clearly outperform other algorithms, this work focuses on them. However, the reader can find the complete set of conducted experiments in [78].

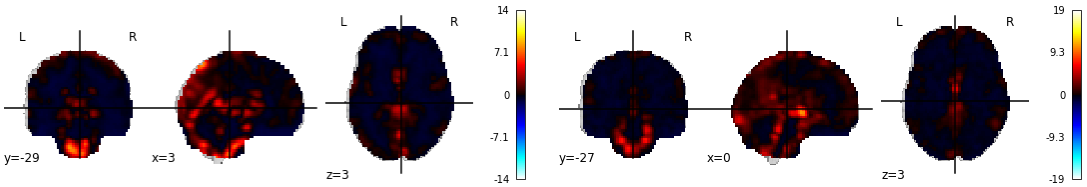
SVM is a supervised algorithm that can be used for classification and regression tasks. For classification, this algorithm defines a hyperplane in a higher dimensional space to separate the features. The main goal of the algorithm is to find a plane with the highest margin (i.e., the maximum distance) between points for the different classes. Then, any new sample can be mapped into the same higher dimensional space and consequently assigned to a category based on the position of the corresponding points with respect to the planes that separate the diffferent classes. SVM is mainly characterized by the *kernel* type [79] considered and some hyper-parameters [80] such as $C$ and $\gamma$. For the sake of completeness, we recall that we call $C$ the regularization parameter that controls the trade-off for achieving a low training error and a low testing error, and it determines the ability to generalize for the associated classifier with respect to unseen data. Higher values of $C$ create the best separation, but they may return a hyperplane that does not possess a large minimum margin between the two classes.

Instead, the parameter $\gamma$ influences the curvature we want in the decision boundary whenever we make use of a polynomial or sigmoid kernel.

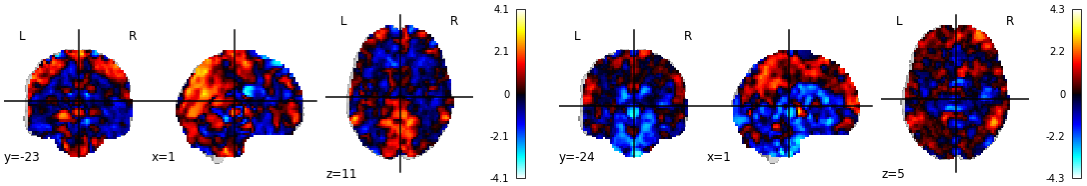
To clarify the way $C$ and $\gamma$ work, we can start from stating that we aim to solve the following optimization problem [81]:

$$min\frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\zeta_i \tag{3.1}$$

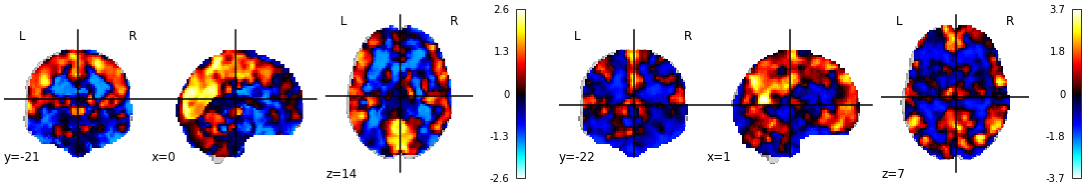
where $w$ stands for the normal vector to the hyperplane, $\zeta_i$ is a slack variable designed to indicate the distance between the $i^{th}$ voxel vector and the hyperplane, $m$ is the amount of subjects in the dataset and $C$ is the hyper-parameter previously mentioned. When we introduce a non-linear degree of separation between two classes, we generally make use of non-linear kernels when applying SVM, exploiting variations in the $\gamma$ values [81]. In particular, we may have:



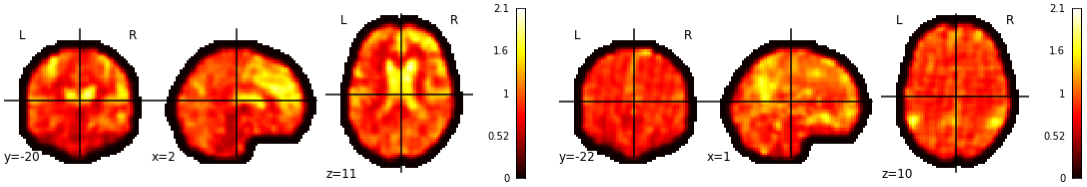

(a) Amplitude of Low Frequency Fluctuations (ALFF).



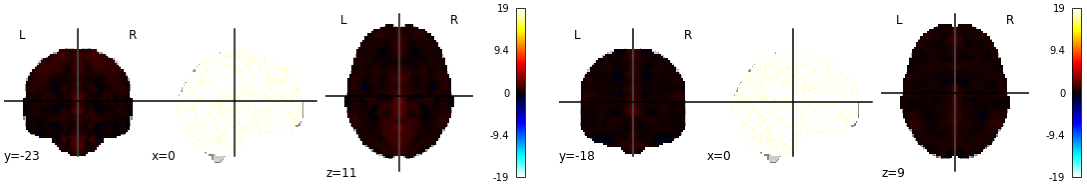

(b) Fractional Amplitude of Low Frequency Fluctuations (fALFF).




(c) Degree Centrality (DC).




(d) Regional Homogeneity (ReHo).




(e) Voxel-Mirrored Homotopic Connectivity (VMHC).

Fig. 3.1 An example of fMRI image for each normalization technique. On the left the class FTD-GNR+, on the right the class oHC.

- Sigmoid kernel, with the following mathematical form:

$$tanh(\gamma x_i^T x_j + r) \tag{3.2}$$

- Polynomial kernel, with the following mathematical form:

$$(\gamma x_i^T x_j + r)^d \tag{3.3}$$

with $x_i$ being the $i^{th}$ voxel vector, $x_j$ representing the voxel vector mapped onto the hyperplane space, $r$ being a constant and $d$ the number of subjects in the system. Similarly to $C$, high values of $\gamma$ may bring overfitting to the model.

RF [82] is another supervised machine learning algorithm that combines several decision trees in order to obtain a single optimal result. A decision tree can be represented as a flowchart where internal nodes test the attributes of the data-samples, branches are outcomes of this test and terminal nodes (leaves) are labels. The choice of several hyper-parameters can influence the final result: some of the most critical are the number of decision trees included in the forest algorithm, the maximum number of features that should be taken into account to have a decision split (this parameter is particularly useful when the amount of features is high, since it can reduce the dimensionality of a mathematical problem) and the highest number of levels that should be used in the algorithm training. Furthermore, the selection of the minimum number of samples required in order to have a decision split or a leaf can be crucial.

### 3.1.3 Pre-processing operations

Some pre-processing operations have been taken into account before applying the machine learning algorithms above mentioned. First, we remark that the 3D images composing the dataset represent human brains without the skull structure, surrounded by a large area of empty voxels. Since such an area is useless for classification purposes, then it can be safely removed from each image. As a matter of fact, the useless region for each image has been first identified, and then the intersection among these regions of all the images has been extracted. Finally, just the intersection has been discarded from each image, this way preserving the same size for each sample of the dataset.

To ensure equal significance across all considered attributes within the dataset, a standardization process was performed. In this context, the features refer specifically to the voxels, the individual units of the 3D image dataset. These voxels, as the fundamental elements of our 3D image, were the principal components considered and analyzed for classification

purposes. In our experiments, we have tested three different scalers for the standardization process:

- Standard Scaler (SS): let $X$ be the matrix containing all the data-samples in the columns and the features in the rows, while $\mu$ and $\sigma$ be the mean and standard deviation associated to each feature, respectively. Then, the standard scaler removes the mean and scales all the data to unit variance, namely:

$$X' = \frac{X - \mu}{\sigma} \tag{3.4}$$

- Robust Scaler (RS): let $m$ be the median observation found within the dataset for each feature, while $p_{75}$ and $p_{25}$ be the 75-th and 25-th percentiles, respectively. Then the robust scaler removes the median and uses a scaling method relying on the interquantile range, such as:

$$X' = \frac{X - m}{p_{75} - p_{25}} \tag{3.5}$$

- Min-Max Scaler (MMS): let $X_{\min}$ and $X_{\max}$ be the minimum and maximum values for each feature in the dataset, respectively. Then, the Min-Max scaler scales the dataset between 0 and 1 by:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{3.6}$$

Furthermore, the use of Principal Component Analysis (PCA) [83] has been assessed to evaluate the performance impact of a dimensionality reduction in the system. PCA can be used to highlight primary patterns from a large set of data with the isolation of the most relevant features (or “components") in the input information. However, since the use of PCA has provided more unstable results in our experiments, all the accuracy values included in chapter 4 have been collected without using principal component analysis.

### 3.1.4 Description of the experiments

In order to find the best values for the hyper-parameters of both the algorithms, a grid search hyper-parameter optimization is computed at the beginning of our experiments. For SVM, we have considered $C = \{0.1, 1, 10, 100, 1000\}$, $\gamma = \{1, 0.1, 0.01, 0.001, 0.0001\}$, while linear, polynomial and sigmoid functions have been tested as kernel.

For RF, we have analyzed the impact on the algorithmic training process by varying the number of decision trees between 200 and 2000 at a 10-step interval, the number of

maximum levels between 10 and 109 at an 11-step interval, the minimum number of samples required in order to have a decision split ($\{2,5,10\}$) and a leaf ($\{1,2,4\}$).

Once the optimal hyper-parameters have been found, a Stratified Shuffle Split (SSS) [84] strategy has been adopted for validation: it randomly splits data into train and test sets, by preserving the proportion of distribution of the information between train and test data. This strategy ensures an equal representation of patients and controls in the resulting dataset. As part of our validation process, we perform a set of 100 iterations known as "rounds of split". Each round involves a new random split of data, which allows us to account for any statistical anomalies and ensures the robustness of our model.

As a detail, we remark that our experiments have been implemented using the Python programming language, with the functionalities provided by the scikit-learn library. The code is publicly available at [78].

## 3.2 Analysis of CS and VH mice ultrasonic communication and their behavioral structure

### 3.2.1 Animals

In this research, 20 adult B6;129PF2 mice aged between 3-6 months were used. These mice were purchased from the Jackson Laboratories situated in Bar Harbor, ME, USA and hosted in the University of Brescia animal facility. The facility was maintained at a constant temperature of 23 ± 2°C and humidity level of 55 ± 15%. The mice were housed in standard cages and subjected to a 12-hour light/dark cycle, with unrestricted access to food and water.

For the purpose of the experiments, the mice were divided into two groups. The control group was named the vehicle (VH) group and consisted of 9 mice. These mice were treated daily with medium-chain triglyceride oil, via oral gavage over the span of 2 weeks.

The second group was called the Cannabis Sativa (CS) oil group and consisted of 11 mice. The mice in this group received CS oil, sourced from GARDAJOINT S.R.L.S (Sommacampagna, Italy), via oral gavage for a period of 2 weeks. The detailed treatment protocol has been mentioned in the work of Mastinu et al [85].

### 3.2.2 USV and behavioral recordings

The process of conducting behavioral tests was strictly in accordance with European Communities Council Directive guidelines and had received approvals from the University of Brescia's Animal Welfare Committee and the Italian Ministry of Health. The method in-

volved a 2-week treatment period for mice, followed by a male-female social interaction test. This test was initiated by isolating male mice for 48 hours, as specified in Silverman et al. [86], after which an unfamiliar female was introduced to the male's cage for a duration of 5 minutes.

The behaviors of these mice were recorded via a camera stationed 20cm above the cage. The Observer XT software (version 14.1, Noldus, The Netherlands) was used to analyze the social behavior of mice, such as nose-nose sniffing, body sniffing, anogenital sniffing, contact, and following. In addition, the software was also utilized to observe non-social behaviors such as cage exploring, immobile position, and self-grooming.

Concurrently, an ultrasound-sensitive microphone (CM16/CMPA, Avisoft Bioacoustics, Berlin) was employed to record the ultrasonic vocalizations emitted by the mice, a technique implemented in previous studies [87, 88]. Refer to Fig. 3.2 for a visual representation of our experimental procedure.

The resulting spectrograms of each mouse were meticulously examined by the operator, and the USVs were classified based on Dr. Scattoni's [3] classification (Fig. 3.3).

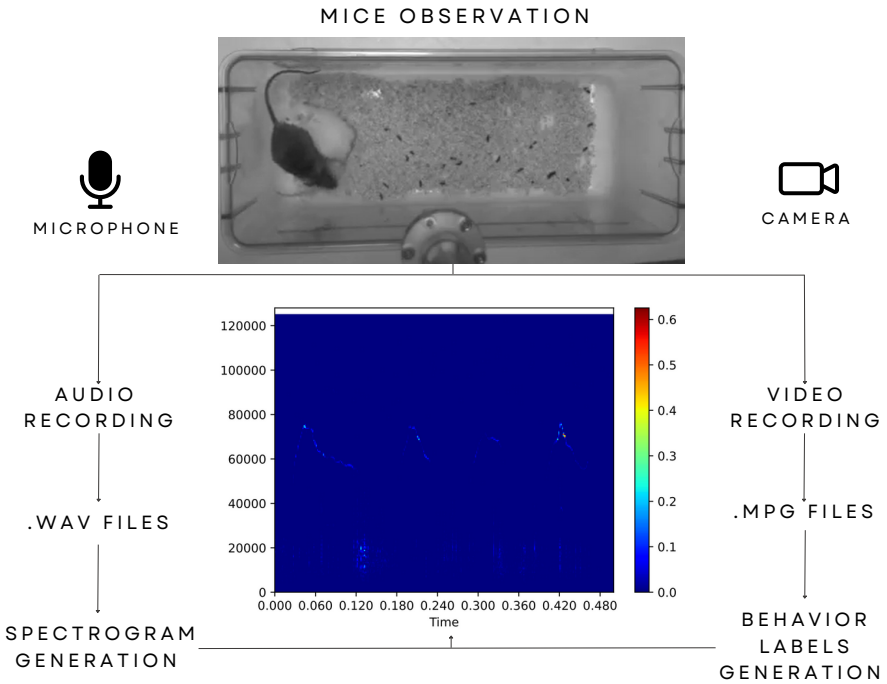



Fig. 3.2 Data extraction and spectrogram generation scheme

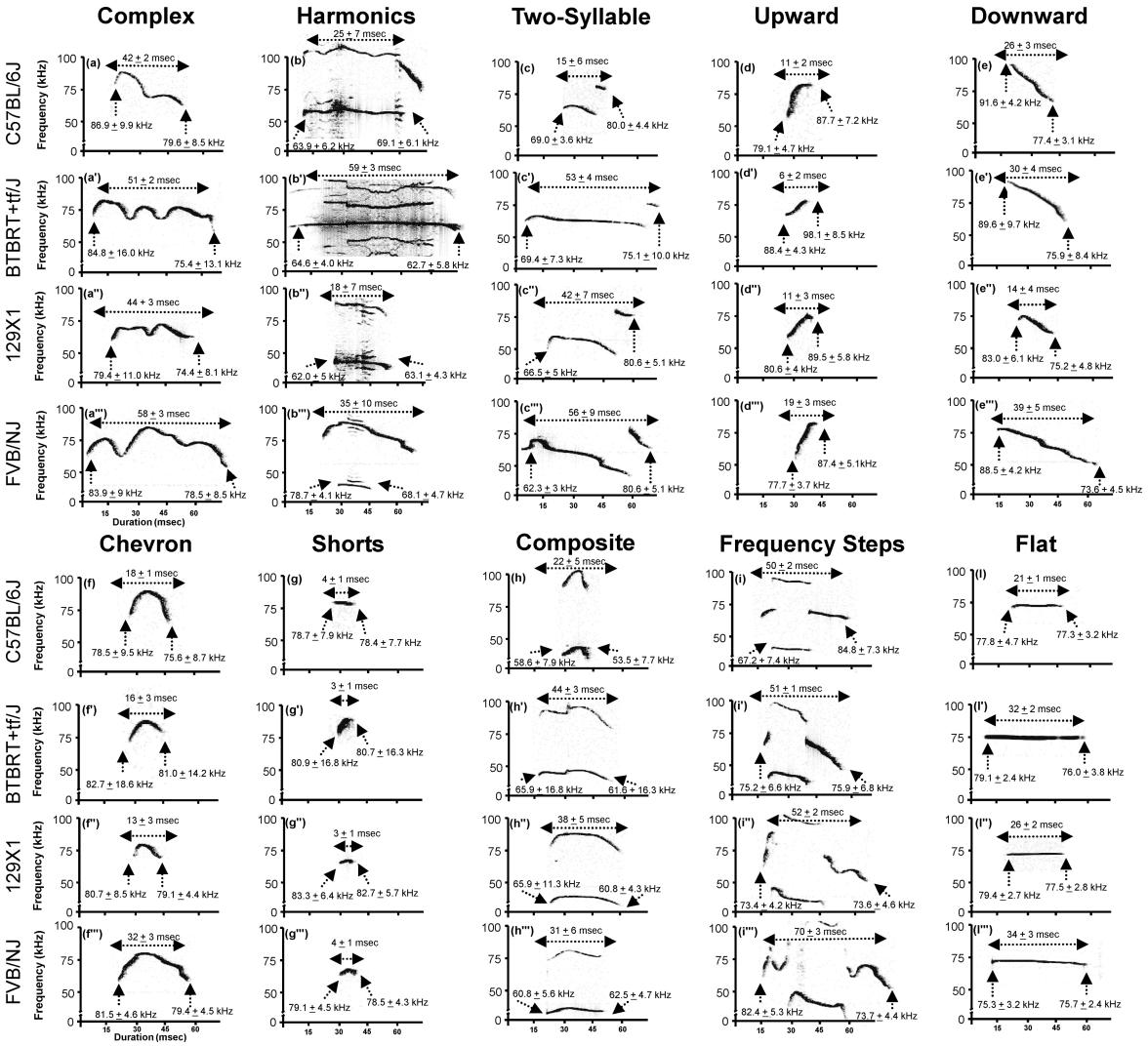



Fig. 3.3 USVs classifications by Scattoni et al. [3]

### 3.2.3 Statistical analysis

We utilized Python 3.8.13 [89] and Spyder integrated development environment [90] to facilitate the computational processes involved. Our approach was multifaceted, integrating several analyses to ensure a comprehensive understanding of the data.

A significant portion of our analysis revolved around the distributions of single syllables, variable-length patterns, and USVs-behavior analysis. For each group (CS and VH), we computed the total count of each syllable, which resulted in the single syllable distribution.

Next, we investigated patterns within the sequences of sounds. We specifically examined sequences of two, three, and four consecutive syllables. This variable-length pattern distribution analysis helped us identify any underlying patterns that might exist. We established a pause time threshold of 0.55 seconds to distinguish between closely connected sounds and the start of a new sequence. A cluster of syllables was classified as a pattern if the time gap between the constituent syllables fell below this threshold.

Moreover, we performed a USVs-behavior analysis where we segmented the spectrogram data based on specific mouse behavior and analyzed the total count of variable-length patterns of sounds for each behavior and exam group. To simplify the visualization of consecutive sounds, we selectively picked a particular portion of all USVs. We considered the most frequent 50% of available sounds for two-sound sequences, 5% for three-sound sequences, and 2% for four-sound sequences. This selection aimed to highlight the differences between the two groups of mice using a representative subset of the data.

After data collection and preprocessing, we turned to statistical tests to validate our findings. We applied the Chi-square goodness-of-fit test, setting our significance level [91] at $p = 0.05$. This test allowed us to assess the statistical significance of the observed differences in our data.

To further deepen our understanding, we measured the degree of similarity between the CS and VH distributions using two metrics: the Cosine Similarity (CoS) [92] and the Jensen–Shannon divergence (JSD) [93]. The CoS, used in an inner product space, measures the similarity between two non-zero vectors $X$ and $Y$ by computing the dot product of the vectors divided by the product of their lengths. The expression for CoS is:

$$\mathrm{CS}(X,Y) = \frac{\sum_{i=1}^{n} X_i Y_i}{\sqrt{\sum_{i=1}^{n} X_i^2 \sum_{i=1}^{n} Y_i^2}}, \tag{3.7}$$

where $X_i$ and $Y_i$ represent the $i$-th element of the two $n$-length vectors $X$ and $Y$.

Our analysis also included the use of the JSD, a metric allowing us to quantify the similarity between two probability distributions. The JSD, directly related to the Kullback-Leibler divergence, is symmetric and always yields a finite value. It is defined as:

$$\mathrm{JSD}(X||Y) = \frac{\mathrm{D_{KL}}(X||Z) + \mathrm{D_{KL}}(Y||Z)}{2}, \tag{3.8}$$

where $Z = \frac{X+Y}{2}$ and $\mathrm{D_{KL}}(X||Y)$ is the Kullback-Leibler divergence between $X$ and $Y$, defined as:

$$\mathrm{D_{KL}}(X||Y) = \sum_{i=1}^{n} X_i \log_2 \frac{X_i}{Y_i}. \tag{3.9}$$

Unlike the CoS, the JSD interprets 0 as the maximum similarity and 1 as the maximum distance between the distributions.

## 3.3 Data preparation and automatic behavior classification

### 3.3.1 Data preprocessing and dataset preparation for automatic behavior classification based on USV patterns

The preprocessing stage is a critical component of any machine learning workflow, particularly in the context of USV data. Given the complexity and high dimensionality of the USV data, appropriate processing steps are indispensable to streamline the data.

In the initial stages, our focus was on the reduction of spectrograms. However, given the memory efficiency considerations, we decided against generating the spectral representation plot. The creation of these plots had previously resulted in what we term 'memory errors', a situation where the computing system was overwhelmed and ran out of available memory. To circumvent this issue, we shifted our approach to generating .npy matrices from audio files instead.

After dimensionality reduction, one of the significant steps in the preprocessing pipeline is the application of one-hot encoding [94] to categorical label data. This technique, fundamental in machine learning, transmutes nominal categorical data into a form known as a binary vector. In this new form, each distinct category from the original data assumes a different binary digit ("0" or "1") in the resultant vector. In simpler terms, each category is presented as a unique "slot" in the binary vector, where the presence of a category is signified by a "1" (an "active" state) and its absence by a "0" (an "inactive" state). Each binary vector, therefore, offers a clear, unambiguous representation of its corresponding category.

After the one-hot encoding process, we initiated a purge of redundant labels, specifically targeting those whose vectors comprised solely of zeros. These labels, devoid of any "active" state, are typically considered as noise within the data set as they provide no discriminatory information for the learning algorithm. The indices of these zero-only label vectors were isolated and subsequently removed from the label and spectrogram data sets, thereby enhancing the efficiency and accuracy of the subsequent machine learning operations in the pipeline.

For quality assurance, a check was performed by plotting a spectrogram to verify if there had been a translation of the USVs. This involved creating a Short-Time Fourier Transform (STFT) of the signal values, taking the square of the absolute values to get a spectrogram, and displaying it using the librosa library [95]. The visualization was done in both jet color map and grayscale for a comprehensive understanding.

After creating matrix dataset, we decided to try to work also with .png images. For this purpose we used two different techniques: using PIL [96] and matplotlib [97] libraries.

During PIL dataset creation, for each spectrogram associated with an item, we normalized the spectrogram data, essentially compressing the data range between 0 and 1. This is a crucial step that aids in mitigating the influence of outliers and ensures uniformity in the data processing pipeline. The normalized spectrogram is then converted into a grayscale image, which is then saved in the appropriate "train" or "test" directory, depending on the index of the item.

For matplotlib technique, we iterated over the spectrogram data and displayed them with inverted colors for enhanced contrast using the 'imshow' function from matplotlib, with specific parameters set for optimized visualization. The color bar, legend, axes titles, ticks, and tick labels were eliminated to ensure a clean, unobstructed view of the spectrogram. As during previous technique, spectrograms were saved in "train" or "test" directory.

### 3.3.2 Preliminary behavior classification algorithms trials

For the first trials of automated prediction, a convolutional neural network was chosen. This algorithm takes an input image, assigns importance (learnable weights and biases) to various aspects of the image, and differentiates one from the other.

Different weights (automated, manual, and without any weight) and model structure settings were tested. Most promising results were obtained using automatic weighing and they are shown in this work.

The initial stage of the model architecture involves the creation of multiple convolutional layers. This is achieved using the Conv2D function from the Keras library [98], with varying numbers of filters (8, 16, 32, 64, 96) specified. The Conv2D function applies a 2D convolution operation to the input, helping the model learn local features.

Each convolutional layer uses a three-by-three filter (kernel size), the Rectified Linear Unit (ReLU) [99] activation function, and "same" padding, which implies that padding is done in such a way that the output feature map has the same width and height dimensions as the input feature map. This is achieved by padding the input feature map with zeros around the border before applying the convolution operation.

After each set of one or two convolutional layers, a MaxPooling2D layer [100] is added to the architecture. This layer uses a two by two pool size, which reduces the spatial dimensions (width and height) of the input by taking the maximum value over the window defined by pool size for each dimension along the features axis.

The BatchNormalization layer [100] is employed after every pooling operation. This layer normalizes the activations of the previous layer for each batch by transforming the

activations to have mean zero and standard deviation one. This helps to improve the speed, performance, and stability of the neural network.

Towards the end of the model, the GlobalAveragePooling2D function [100] is used to transform the feature map to a single number per feature map, by averaging the values. Thereafter, a fully connected (dense) layer is added with 64 neurons and ReLU activation function, followed by another dense layer with 32 neurons.

A Dropout layer [100] is incorporated after each dense layer to prevent overfitting. It does this by randomly setting a fraction rate of input units to 0 at each update during training, which helps to prevent overfitting.

The final layer of the model is another dense layer. The number of neurons in this layer corresponds to the number of output classes, either 4 or 8, based on the variable filter_behaviors. The activation function used in this layer is the softmax function [100], which outputs a vector representing the probability distributions of a list of potential outcomes.

In terms of model compilation, the Adam optimizer [101] is used with a learning rate of 1e-4 and decay of 1e-6. The model's loss function is specified as categorical cross-entropy, and the metrics for evaluating the model performance are accuracy, precision, and recall [102].

The input data is reshaped to include the channel dimension, and the model is trained using the fit function for 100 epochs with a batch size of 4. The model is then evaluated using the test dataset.

Finally, a confusion matrix is plotted to visualize the performance of the machine learning model. The matrix can either be plotted without normalization or with normalization by setting the normalized parameter to True. The confusion matrix provides a more detailed breakdown of a model's performance, showing the correct and incorrect classifications for each class.

Figs. 3.4, 3.5, 3.6 and 3.7 show Python structure and graphic visualization of presented CNNs.

```
    # Define the model
model = keras.Sequential()

# Add layers to the model
model.add(layers.Conv2D(8, (3, 3), activation='relu', padding='same',
                                    input_shape=(176, 3907, 1)))
model.add(layers.Conv2D(8, (3, 3), activation='relu', padding='same')
                                   )
model.add(layers.MaxPooling2D((2, 2)))
```

```
model.add(layers.BatchNormalization())

model.add(layers.Conv2D(16, (3, 3), activation='relu', padding='same'
                                    ))
model.add(layers.Conv2D(16, (3, 3), activation='relu', padding='same'
                                    ))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.BatchNormalization())

model.add(layers.Conv2D(32, (3, 3), activation='relu', padding='same'
                                    ))
model.add(layers.Conv2D(32, (3, 3), activation='relu', padding='same'
                                    ))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.BatchNormalization())

model.add(layers.Conv2D(64, (3, 3), activation='relu', padding='same'
                                    ))
model.add(layers.Conv2D(64, (3, 3), activation='relu', padding='same'
                                    ))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.BatchNormalization())

model.add(layers.GlobalAveragePooling2D())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.BatchNormalization())
model.add(layers.Dropout(0.5))

model.add(layers.Dense(32, activation='relu'))
model.add(layers.BatchNormalization())
model.add(layers.Dropout(0.5))

if filter_behaviors == 'Y':
    model.add(layers.Dense(4, activation='softmax'))
else:
    model.add(layers.Dense(8, activation='softmax'))

# Define the precision metric
precision = tf.keras.metrics.Precision()
recall = tf.keras.metrics.Recall()
# Compile the model
opt = tf.keras.optimizers.legacy.Adam(learning_rate=1e-4, decay=1e-6)
model.compile(optimizer= opt,
              loss='categorical_crossentropy',
```

```
            weighted_metrics=['accuracy']) #weighted metric
                                           accuracy, without
                                           precision e recall
```

Fig. 3.4 First version of the model structure with 8 convolutional layers

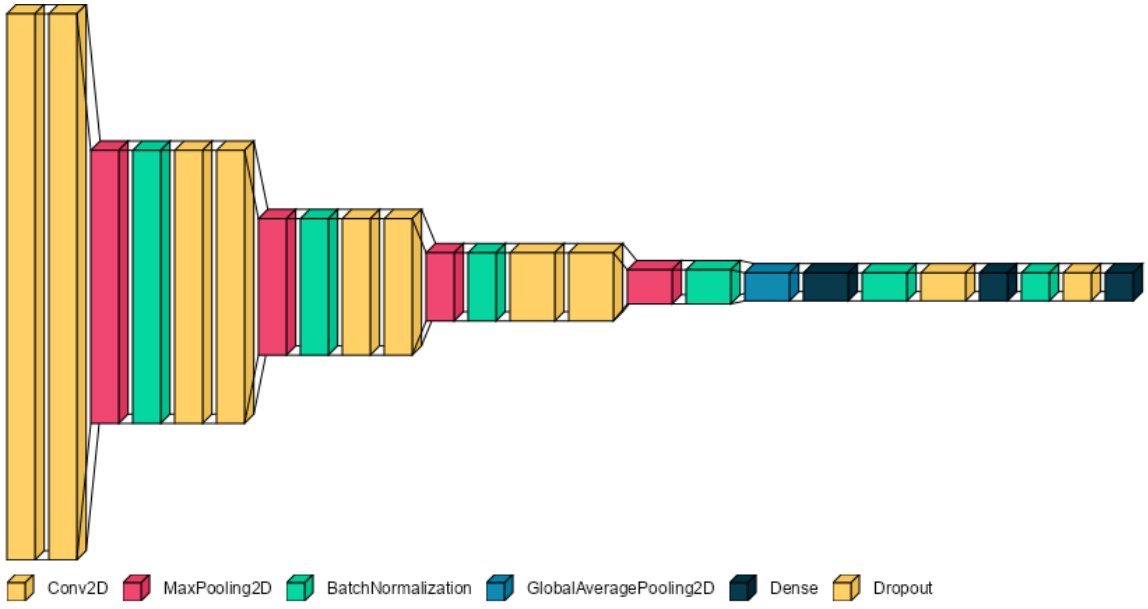



Fig. 3.5 Visual representation of 8 convolutional layers neural network.

```
# Define the model
model = keras.Sequential()

# Add layers to the model
model.add(layers.Conv2D(8, (3, 3), activation='relu', padding='same',
                                    input_shape=(176, 3907, 1)))
model.add(layers.Conv2D(8, (3, 3), activation='relu', padding='same')
                                   )
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.BatchNormalization())

model.add(layers.Conv2D(16, (3, 3), activation='relu', padding='same'
                                   ))
```

```
model.add(layers.Conv2D(16, (3, 3), activation='relu', padding='same'
                                    ))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.BatchNormalization())

model.add(layers.Conv2D(32, (3, 3), activation='relu', padding='same'
                                    ))
model.add(layers.Conv2D(32, (3, 3), activation='relu', padding='same'
                                    ))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.BatchNormalization())

model.add(layers.Conv2D(64, (3, 3), activation='relu', padding='same'
                                    ))
model.add(layers.Conv2D(64, (3, 3), activation='relu', padding='same'
                                    ))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.BatchNormalization())

model.add(layers.Conv2D(96, (3, 3), activation='relu', padding='same'
                                    ))
model.add(layers.Conv2D(96, (3, 3), activation='relu', padding='same'
                                    ))
model.add(layers.MaxPooling2D((1, 2)))
model.add(layers.BatchNormalization())

model.add(layers.GlobalAveragePooling2D())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.BatchNormalization())
model.add(layers.Dropout(0.5))

model.add(layers.Dense(32, activation='relu'))
model.add(layers.BatchNormalization())
model.add(layers.Dropout(0.5))

if filter_behaviors == 'Y':
    model.add(layers.Dense(4, activation='softmax'))
else:
    model.add(layers.Dense(8, activation='softmax'))

# Define the precision metric
precision = tf.keras.metrics.Precision()
recall = tf.keras.metrics.Recall()
# Compile the model
```

```
opt = tf.keras.optimizers.legacy.Adam(learning_rate=1e-4, decay=1e-6)
model.compile(optimizer= opt,
              loss='categorical_crossentropy',
              weighted_metrics=['accuracy']) #weighted metric
                                                accuracy, without
                                                precision e recall
```

Fig. 3.6 Second version of the model structure with 10 convolutional layers

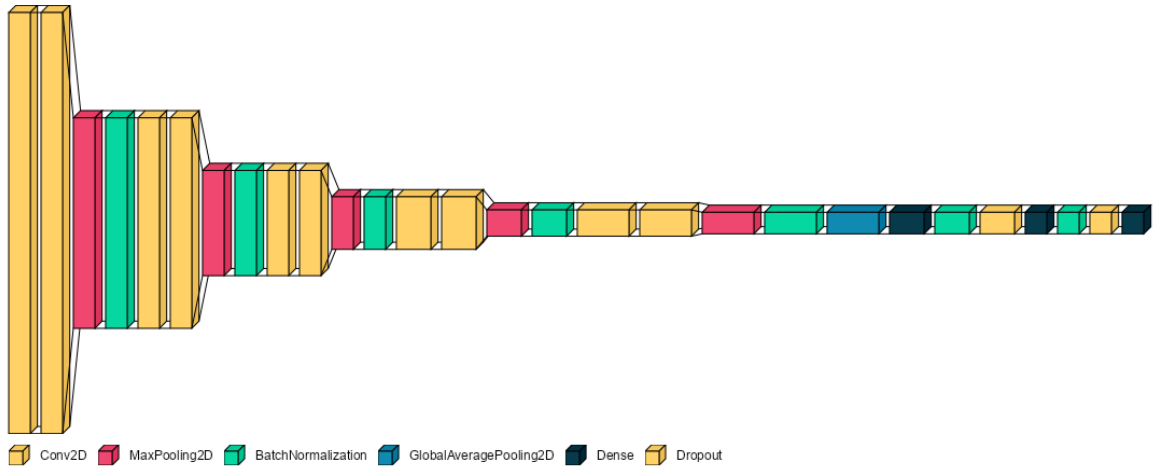



Fig. 3.7 Visual representation of 10 convolutional layers neural network.

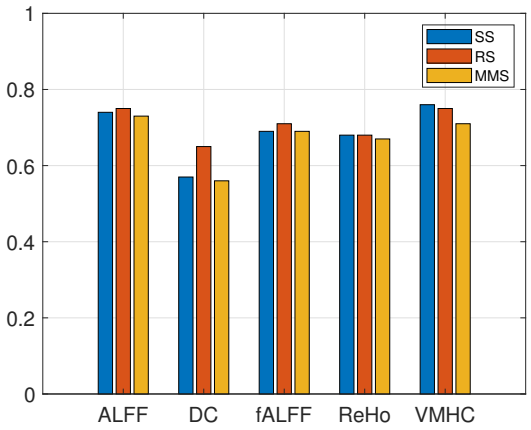
# Chapter 4

# Results and discussion

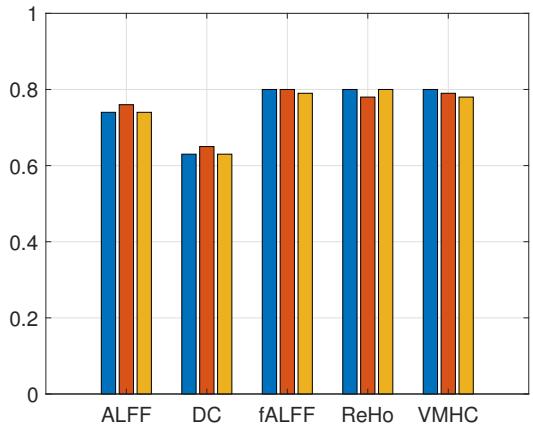
## 4.1 Evaluating the accuracy of ML algorithms for detecting FTD

The results obtained through the execution of the SVM algorithm have provided an interesting insight into the information contained within the voxel of fMRI images. Fig. 4.1a reports the accuracy performance for each MRI normalization technique, showing the results of the different feature standardization methods considered in our experiments, while Fig. 4.1c indicates the mean accuracy and standard deviation computed on the feature standardization methods. First, it is possible to observe consistent results among the different standardization techniques, with minor changes among methods. The average accuracy of the SVM methodology on the whole dataset on all the normalization techniques and scalers is equal to 69%, at a standard deviation of 6%.

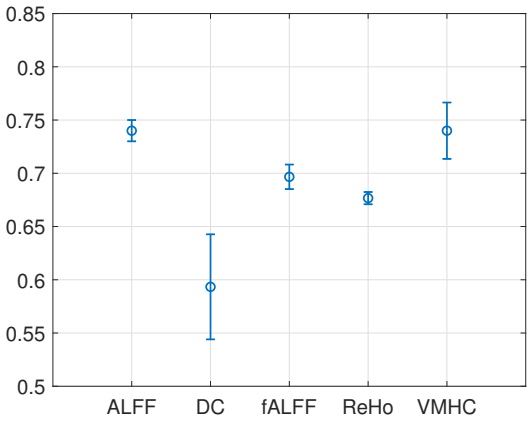
Looking more closely into the results obtained through the SVM technique, the ALFF and VMHC strategies have achieved the best performance at 74% of mean accuracy, with a maximum accuracy recorded on robust (75% on ALFF) and standard (76% on VMHC) scalers. To give more context to the results, the classification accuracy obtained on the other MRI normalization techniques has not exceeded 71%. Interestingly, the result obtained on fALFF (71% with the robust scaler, with an average accuracy on all scalers of 70% and 1% standard deviation) has been worse than that recorded on ALFF, indicating that the search for the optimal hyperplane to separate patients and controls in the dataset appears to be made more complex by the exclusion of the typical noise connected to mental states from the MR images. To clarify the term "typical noise" previously mentioned, it refers to the inherent variables or inconsistencies that exist within MR images due to fluctuating mental states of



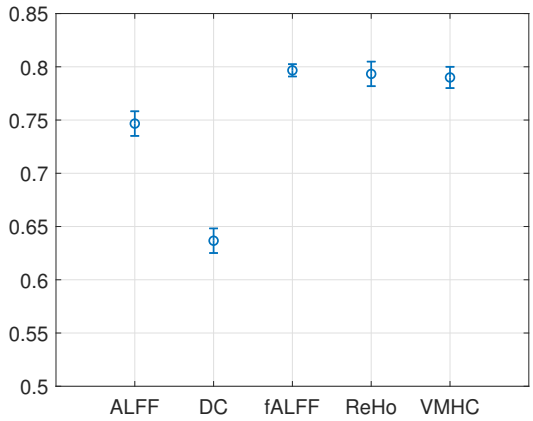

(a) SVM accuracy for each MRI normalization technique for the distinct feature standardization methods.




(b) RF accuracy for each MRI normalization technique for the distinct feature standardization methods.




(c) SVM evaluation for each MRI normalization technique (mean accuracy and standard deviation).




(d) RF evaluation for each MRI normalization technique (mean accuracy and standard deviation).

Fig. 4.1 Performance of SVM and RF in terms of accuracy using SSS.

subjects — such as feelings of hunger or anxiety — during the imaging process. It was noted that this trend was consistent across all three data scalers utilized in the study.

Furthermore, the DC technique has revealed the worst results, a factor that may indicate the low relevance of single nodes composing the whole-brain network to spot the presence of FTD in elderly patients. This conclusion has been confirmed by a coherent set of results recorded using different data scalers.

In terms of standard deviation, the results obtained on the ReHo technique has provided the highest stability in the experiment, with a value of 0.6% and a mean accuracy rate measured at 68% (a value recorded both on average among the different data scalers and looking at the robust scaling technique, which provided the best result on ReHo). When applied to the more accurate results obtained on ALFF and VMHC, the same approach

returns a standard deviation respectively equal to 1% and 3%, leading us to conclude that the SVM algorithm provides more reliable results on the ALFF technique.

By looking at the performances on different scalers, the robust methodology has outperformed the standard and min-max techniques, having an average accuracy across all the MRI normalization methodologies equal to 71%, against, respectively, 69% and 68%. This result is certainly interesting, as it translates into a higher efficiency of the SVM algorithm when the dataset is elaborated in order to handle outliers in the dataset. The strategy to introduce a degree of distortion in the data by locating all the voxels in a space with mean equal to 0 and a fixed standard deviation has proved to be less effective than the robust scaling. The same is true for the attempt performed by the min-max technique to scale the data while preserving their original structure.

The use of an RF methodology has provided overall better results with respect to SVM across all the techniques and scaling methodologies, with an average accuracy on the whole dataset on all the pre-processing techniques and scalers equal to 75% and a slight increase in the standard deviation of the outcomes, which has passed from 6% to 7% (see Fig. 4.1b and 4.1d). In order to compare the results obtained with the two algorithms, the experiments run with the RF strategy have been of course validated with the same SSS validation function used on the SVM algorithm.

While ALFF and VMHC have been confirmed to have a valid informational value by using an RF algorithm, other techniques have managed to lead the classifier to an even better output. In general terms, an improvement has been recorded across all the MRI normalization techniques and on each scaling methodology.

Furthermore, the creation of a random forest of decision trees has proved to find an actual improvement in the passage from ALFF to fALFF data elaboration. Specifically, the first technique has led the classifier to record a 74% accuracy as average value across all the scalers (peaking at 76% on the robust strategy), while the additional data normalization layer brought by fALFF has allowed the RF reach an 80% average accuracy. In both cases, the standard deviation has been recorded at a low level, leading us to validate the interpretation of a prediction improvement. Actually, the standard deviation has decreased following mean accuracy enhancement, moving from 1% to 0.5%.

An interesting confirmation of the results obtained through SVM comes from the evaluation that has been performed on the DC normalization technique. While the mean accuracy has improved (reaching 64% at 1% standard deviation, with robust scaling recording a 65% accuracy), the classifier has proved to encounter significantly more complexities (meaning a decreased accuracy) while looking for correct class discrimination on these images, compared to the other normalization techniques. Therefore, the conclusion on the seemingly low

Table 4.1 Recap of SVM (top) and RF (bottom) accuracy for each fMRI normalization technique for the distinct feature standardization methods. The mean and standard deviation are also reported.

| | ALFF | DC | fALFF | ReHo | VMHC |
|---|---|---|---|---|---|
| Standard scaler | 0.740 | 0.570 | 0.690 | 0.680 | 0.760 |
| | 0.740 | 0.630 | 0.800 | 0.800 | 0.800 |
| Robust Scaler | 0.750 | 0.650 | 0.710 | 0.680 | 0.750 |
| | 0.760 | 0.650 | 0.800 | 0.780 | 0.790 |
| Min-Max Scaler | 0.730 | 0.560 | 0.690 | 0.670 | 0.710 |
| | 0.740 | 0.630 | 0.790 | 0.800 | 0.780 |
| Mean | 0.740 | 0.593 | 0.697 | 0.677 | 0.740 |
| | 0.747 | 0.637 | 0.797 | 0.793 | 0.790 |
| Standard deviation | 0.010 | 0.049 | 0.012 | 0.006 | 0.026 |
| | 0.012 | 0.012 | 0.006 | 0.012 | 0.010 |

importance of studying the single nodes composing the whole-brain network appears to be confirmed by the evidence on the RF algorithm.

As previously mentioned, VMHC has returned positive results on the RF framework too, reaching a 79% average accuracy at 1% standard deviation, one of the limited cases in which the standard scaler has outperformed the robust approach (with accuracies equal to, respectively, 80% and 79%). The same can be said about ReHo images (where the standard method has recorded an 80% accuracy, against the 78% value obtained with the robust scaler), with the observation that, while this technique has appeared to have a below-average predictive power on SVM, the RF algorithm has allowed to obtain a 79% average accuracy (with 1% standard deviation) on this subset of information.

Finally, the RF classifier has confirmed the evidence obtained from the SVM algorithm on the different scalers, with the robust technique recording a 76% accuracy against lower results obtained by the robust and min-max scalers (both amounting to 75%). For the sake of readability, we summarize all the results also in Tab. 4.1.

### 4.1.1 Further evaluation

We have extended our evaluation, by comparing the results of this work with those obtained by Premi et al. [4]. While the goals of that paper are not perfectly comparable to those of our research, a certain degree of symmetry can be identified between the two sets of results. More precisely, Premi et al. have studied only SVM as classification method, however they have

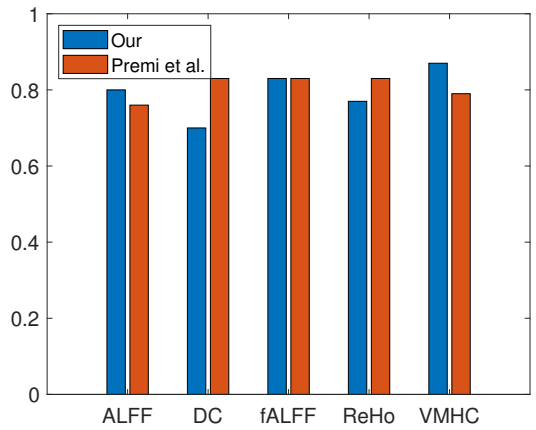



Fig. 4.2 Comparison with the work by Premi et al. [4] using LOOCV.

not exploited any data scaling methodology. Furthermore, the $C$ parameter of the algorithm has been manually tested in their work, not considering any hyper-parameter optimization.

Note that Premi et al. [4] have implemented a Leave-one-out cross-validation (LOOCV), with $n$ runs (where $n$ is equal to the number of subjects in the dataset) having a single sample (singleton) as test, and the remaining data used to train the SVM. On the contrary, we have decided to exploit the advantages of SSS as a validation system instead of the LOOCV. Indeed, SSS works in such a way that the dataset is randomized and can be split into training and test subsets in a balanced way even thousands of times, leading the overall accuracy to possess a potentially larger statistical significance. This approach should provide more reliable results, considering that LOOCV is only composed of 30 (i.e., the number of subjects in the dataset) rounds of unbalanced split and validation.

However, in order to keep the comparison as fair as possible, we have evaluated our methods by considering LOOCV as validation. Fig. 4.2 and Tab. 4.2 report the results for each MRI normalization technique. Our accuracy refers to the RF algorithm and the best scaler for each normalization method. Remarkably, our highest accuracy has been obtained for VMHC technique (87%), and it outperforms Premi's best performance (83%). For the sake of completeness, we remark that in [4] the authors have applied the PCA technique to the data, since the corresponding dimensionality reduction has brought to accuracy improvements in their model. On the contrary, our experimental results have shown that training our models without PCA leads to more stable and discriminative results with respect to the use of the PCA.

Table 4.2 Comparison with the work by Premi et al. [4] using LOOCV. In bold the highest accuracy returned by our method with VMHC.

| | ALFF | DC | fALFF | ReHo | VMHC |
|---|---|---|---|---|---|
| Our | 0.80 | 0.70 | 0.83 | 0.77 | **0.87** |
| Premi et al. | 0.76 | 0.83 | 0.83 | 0.83 | 0.79 |

### 4.1.2 Further experiments

To investigate the possibility of increasing the accuracy of our methods, we have tried to concatenate ALFF, fALFF, DC, ReHo and VMHC into one 3D array in the $z$-direction. This idea was to exploit the entire information contained in all the MRI normalization techniques, instead of performing a separate analysis. However, in this case the highest accuracy (76%) has been reached using the SVM algorithm with Standard Scaler, proving a decay in the results obtained with respect to the use of a single normalization MRI technique. Interestingly, this result indicates that integrating the information contained in the different normalization MRI techniques leads to a less discriminative feature space with respect to the feature space associated to the distinct MRI pre-elaborations.

Furthermore, the same experiments described in Subsec. 3.1.3 have been performed on another dataset of 34 subjects: 17 asymptomatic carriers of GRN Thr272fs mutation and 17 young non-carriers belonging to the same families. Also this dataset has been acquired by the Centre of Aging and Neurodegenerative Diseases, and similarly divided into the ALFF, fALFF, DC, ReHo and VMHC normalization techniques. However, the accuracy for both SVM and RF algorithms has been proved to be consistently close to 50%; this statistic implies that, with the data currently available, the SVM and RF algorithms do not possess the capacity to effectively differentiate between the two MRI classes.

## 4.2 Ultrasonic vocalizations in CS and VH mice and their correlation with the behavior

The current study of USVs in mice has unlocked interesting patterns and correlations associated with their behavior, particularly in response to treatments like CS. The observed data showcases a significant quantitative and qualitative disparity in the vocalizations of CS-treated mice and vehicle-treated mice.

An inspection of our collected data reveals that CS mice emitted an abundance of syllables, reaching a total count of 4,489, a number significantly higher than the 1,529

Table 4.3 Abbreviation corresponding to each syllable.

| **Syllable** | Chevron | Complex | Composite | Downward | Flat | Frequency steps | Harmonics | Short | Two-syllable | Upward |
|---|---|---|---|---|---|---|---|---|---|---|
| **Abb.** | CH | CX | CM | D | F | FS | H | S | TS | U |

syllables emitted by VH mice. This considerable difference suggests that CS treatment may influence vocalizations in mice, implicating potential alterations in their neural or physiological structures.

Taking our analysis a step further, a detailed examination of the types of sounds produced by the two groups of mice was conducted. The data on counts of syllables and variable-length patterns (up to four consecutive syllables, Fig. 4.3) were analyzed, revealing significant disparities between these distributions. This finding was statistically validated using the Chi-square test, which resulted in a p-value less than 0.001, indicating a high degree of statistical significance.

Tab. 4.3 reports the abbreviations corresponding to the names of the syllables, that will be used in the figures for the rest of the work.

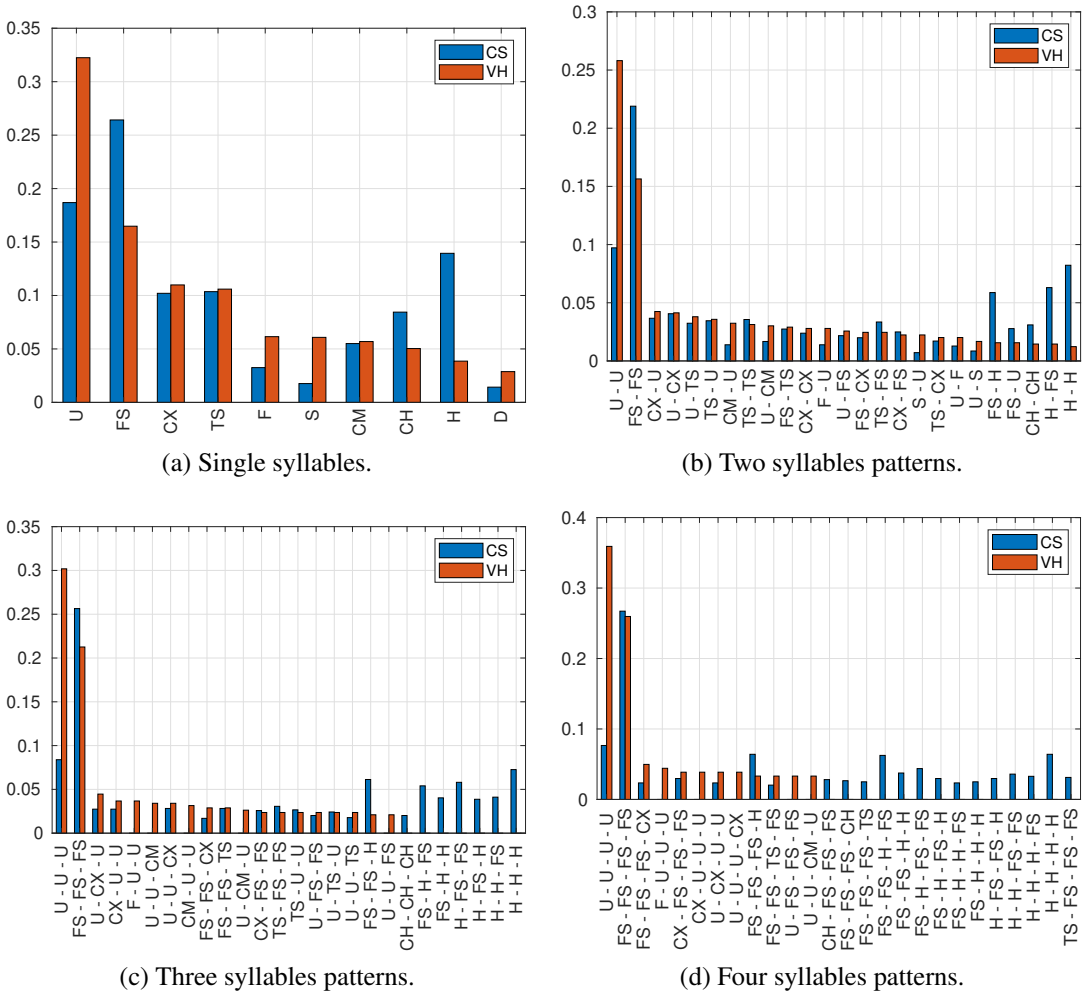
Upon examining the distribution of single syllable types in Fig. 4.3a, we find that both CS and VH mice exhibit a high occurrence of *upward* and *frequency steps* syllables. However, an interesting observation is the higher occurrence of *harmonics* syllables emitted by CS mice. This type of syllable is considerably rarer in VH mice vocalizations.

Furthermore, the occurrence of *chevron* and *composite syllables* is more frequent in CS mice while being sporadic in VH mice. This variation indicates a broader repertoire of vocalization capabilities in CS mice, potentially attributable to the effects of CS treatment.

The examination of ultrasonic vocalizations in mice and their correlation with behavior presents enlightening findings, particularly when comparing sequences of syllable patterns between different groups of mice. The disparities in the distributions of these patterns, as presented in Fig. 4.3b, become more pronounced with longer sequences.

In the case of CS mice, the most frequent pattern features two consecutive *frequency steps*. Other prevalent combinations include *upward - upward* and *harmonics - harmonics*, indicating a certain preference in syllable combinations within this group. On the other hand, VH mice exhibit a different distribution, with the pair *upward - upward* and *frequency steps - frequency steps* being the most common. Notably, this group shows a more even spread of frequency counts across various consecutive sounds, hinting at a broader range of vocalization patterns employed by VH mice.

Our study further extends into examining patterns of longer sequences, as illustrated in Fig. 4.3c and 4.3d. Both groups of mice show a preference for repeating *upward* and *frequency steps* sounds. However, the frequency gap within the syllables of *frequency steps* notably reduces in the 4-syllables pattern. Strikingly, the right-hand side patterns of both




(a) Single syllables.

(b) Two syllables patterns.

(c) Three syllables patterns.

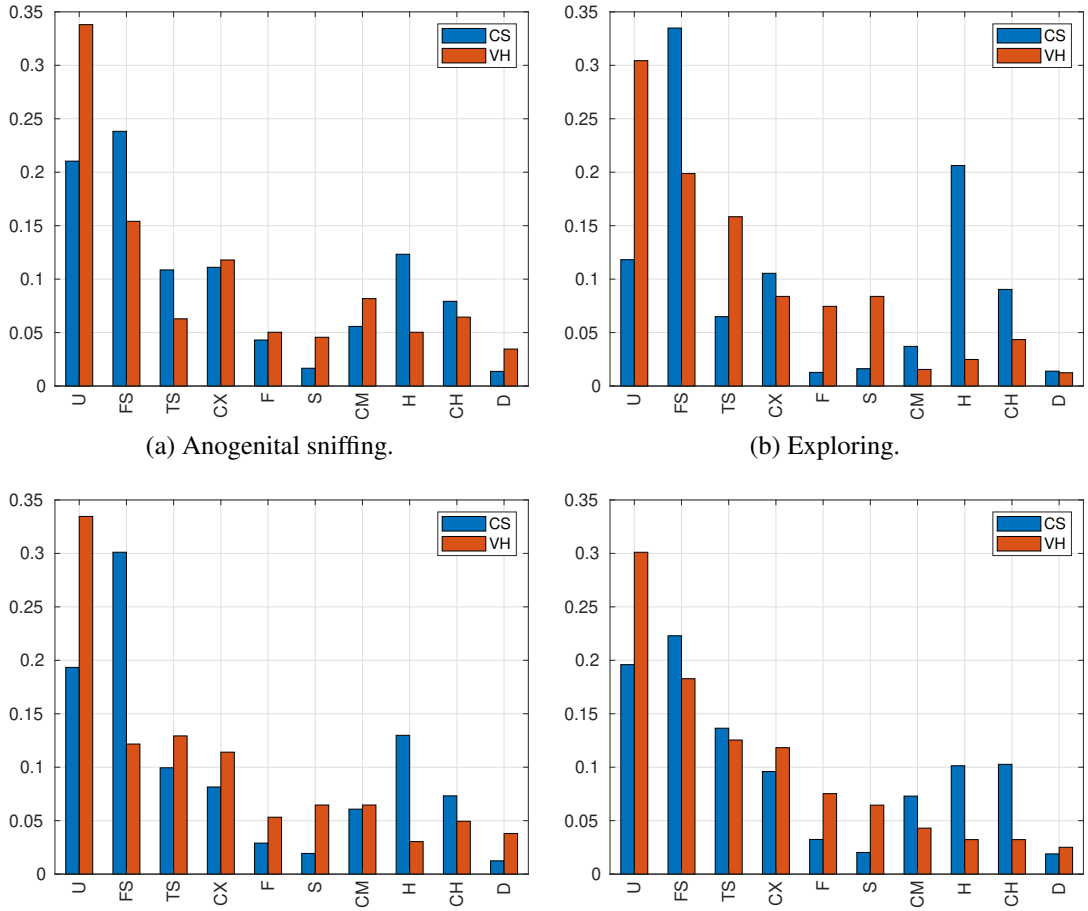
(d) Four syllables patterns.

Fig. 4.3 Distributions of variable-size patterns of syllables for CS and VH.

figures predominantly appear in the CS mice group, with minimal representation in the VH group. This finding suggests an interesting distinction between the two groups in their utilization of these specific syllable pattern types.

The correlation between ultrasonic vocalizations and mice behaviors further enhances our understanding of the pronounced differences between the CS and VH mice groups. Behaviors such as anogenital sniffing, exploring, nose-nose sniffing, and body sniffing were observed and recorded. The CS mice exhibited a higher frequency of both behaviors and vocalizations when compared to VH mice, hinting at a potentially more complex communication system within the CS group.

Moreover, the distribution of sounds and behaviors in CS mice appears to be more evenly spread. This finding indicates greater variability in their sound-producing behaviors, which may suggest a higher level of complexity or flexibility in their communication methods.




(a) Anogenital sniffing.

(b) Exploring.

(c) Nose-nose sniffing.

(d) Body sniffing.

Fig. 4.4 Distributions of syllables. Comparison between CS and VH for each behavior.

Fig. 4.4 illustrates the distributions of the syllables for both CS and VH mice, analyzed separately based on specific behaviors. The significant difference between these distributions, regardless of the considered behavior (Chi-square goodness-of-fit test, $p < 0.001$), further underscores the importance of extended investigations into the correlation between ultrasonic vocalizations and behavior in mice.

The study under review offers an in-depth examination of USVs in mice, with a specific focus on the correlations between these vocalizations and the associated behaviors. A primary finding of this study is the marked difference in vocalization distributions between two groups of mice, namely CS-treated and VH-control mice.

Interestingly, these differences in vocalizations were not uniform but varied significantly depending on the behavior being exhibited. For instance, during exploring behavior, the syllable *harmonics* was notably prevalent in CS mice, while it was largely negligible in their

Table 4.4 Cosine similarity between CS and VH distributions assessed at varying of the pattern lengths and behaviors. Higher values indicate a stronger similarity.

| | 1 syllable | 2 syllables | 3 syllables | 4 syllables |
|---|---|---|---|---|
| Overall | 0.87 | 0.79 | 0.68 | 0.66 |
| Anogenital sniffing | 0.90 | 0.84 | 0.83 | 0.83 |
| Exploring | 0.72 | 0.61 | 0.54 | 0.47 |
| Nose-nose sniffing | 0.80 | 0.60 | 0.49 | 0.42 |
| Body sniffing | 0.91 | 0.82 | 0.79 | 0.77 |

VH counterparts. Similarly, the *two syllable* vocalization exhibited a higher probability in CS mice during anogenital sniffing. Conversely, this trend reversed during exploring behavior.

The study also noted minor variations in body sniffing and nose-nose sniffing behaviors, hinting at nuanced differences in vocalization patterns between the two groups of mice. The examination of longer patterns underscores these differences, emphasizing the distinctive features that set apart CS and VH mice.

Of particular interest was the higher probability of *frequency steps* syllables in CS mice compared to VH mice across varied behaviors. This pattern held true even when each behavior was analyzed in isolation, underscoring the robustness of the observed trend. In contrast, VH mice primarily emitted the syllable *upward*, reinforcing the separate trends observed across the behaviors.

To objectively compare these observations, the study utilized CoS and JSD metrics. These were calculated for various pattern lengths and behaviors and presented in Tabs 4.4 and 4.5, respectively. Both measures underscored an interesting finding - as the pattern length increased, so did the distinctions between CS and VH mice.

Behavior-specific peculiarities were also detected in the study. The behaviors of exploring and nose-nose sniffing showcased the greatest diversity in communication between the two groups. This was reflected in lower CoS values and higher JSD. On the other hand, anogenital sniffing and body sniffing exhibited the highest similarity, indicating shared communication attributes.

These findings provide a nuanced understanding of the complexity of USV communication in mice. They highlight that while the communication patterns certainly vary between the CS and VH groups, this variability is influenced by the particular behavior being displayed.

Table 4.5 Jensen-Shannon divergence between CS and VH distributions assessed at varying of the pattern lengths and behaviors. Lower values indicate a stronger similarity.

| | 1 syllable | 2 syllables | 3 syllables | 4 syllables |
|---|---|---|---|---|
| Overall | 0.06 | 0.13 | 0.35 | 0.49 |
| Anogenital sniffing | 0.04 | 0.11 | 0.16 | 0.15 |
| Exploring | 0.12 | 0.30 | 0.41 | 0.53 |
| Nose-nose sniffing | 0.08 | 0.19 | 0.36 | 0.47 |
| Body sniffing | 0.06 | 0.16 | 0.34 | 0.43 |

## 4.3 USVs dataset preparation and preliminary automatic behavior classification results

### 4.3.1 Dataset preparation

The sophisticated data collection and preprocessing procedures employed have yielded a comprehensive and well-structured dataset. High-frequency microphones were effective in capturing USVs from male mice, with the resultant audio recordings duly saved in WAV format and systematically organized within unique directories for each subject. Concurrently, meticulous behavioral data, including precise temporal records of specific behaviors, were recorded and preserved in a Microsoft Excel file for subsequent analysis.

The imported audio recordings were processed using the librosa library. Careful attention was given to configuring specific parameters to slice the audio signals into consistent-duration segments. A crucial part of this process was the application of the STFT algorithm from the scipy.signal library [103], a method well-suited to the analysis of non-stationary signals such as animal vocalizations. This resulted in the generation of spectrograms for each of these segments.

In the presented computational process, several key parameters are defined to facilitate the precise manipulation of digital audio signals. These include:

- *n_fft = 511*: this parameter defines the length of the Fast Fourier Transform (FFT) window, implying that the FFT algorithm is applied to 511 samples of our audio signal at a time. This window length is a crucial factor that balances the trade-off between frequency resolution and time resolution.

- *n_window = 511*: this value represents the length of the windowed signal before applying the FFT. By setting this equal to “n_fft”, we ensure that the entire segment of our audio signal is considered for the FFT computation.

- *hop_length = 128*: this parameter determines the number of audio samples between successive FFT frames. A smaller hop length value leads to a higher overlap between successive frames and more time resolution, at the expense of increased computational load.

- *window = scipy.signal.kaiser(n_window, 5)*: this statement invokes the Kaiser window function from the scipy library to reduce the spectral leakage in our FFT computation. The second parameter in the Kaiser function, 5, is the beta parameter, which shapes the window and, ultimately, influences the trade-off between the main lobe width and side lobe level.

These parameters were chosen based on empirical trials and the specific requirements of our audio processing task, with the aim of achieving an optimal balance between frequency and time resolution.

Aligning these spectrograms with behavioral annotations was an essential step to ensure the cohesiveness and relevance of the dataset. These annotations, derived from the respective Microsoft Excel files for each mouse, detailed the initiation times, durations, and classifications of behavior observed during the recording sessions. Only behaviors that exceeded a predetermined duration threshold of 0.3 seconds were considered in the analysis, given the focus of this study on behaviorally relevant USVs. It is worth underscoring that the 0.3-second cutoff was determined based on manual timing of the behaviors using video footage and a stopwatch, given the focus of this study on behaviorally relevant USVs. Behaviors with durations less than this threshold could not be accurately discerned via our manual timing method.

A one-hot encoding scheme (Fig. 4.5) was applied to prepare the behavioral annotations for analysis. A predefined set of behavior categories, including anogenital sniffing, contact, following, and others, were used to generate binary labels for each chunk of audio. The resulting labels were merged with the audio chunks, creating a structured dataset for further analysis. Even if we had 8 behaviors, for the final experiments, we decided to use only 4 of them: anogenital sniffing, nose-nose sniffing, exploring, and sniffing body. The remaining four behaviors (contact, following, immobile, and self-grooming) required a larger frequency of occurrence to ensure a robust dataset representation.


Fig. 4.5 One-hot encoding scheme for the behavioral labeling.

An interesting finding revealed by these annotations pertains to the substantial variability in behavior durations among mice, even within the same behavior category, as depicted in Figure 4.6. This variability underscores the diverse range of behavioral responses exhibited by the subjects, thereby introducing an additional layer of complexity when associating these behaviors with corresponding ultrasonic vocalizations. The graphical representation in Fig. 4.6 employs a box plot, where the red marker signifies the 1st quartile of the duration distribution, the blue marker represents the median, and the green marker corresponds to the 3rd quartile.

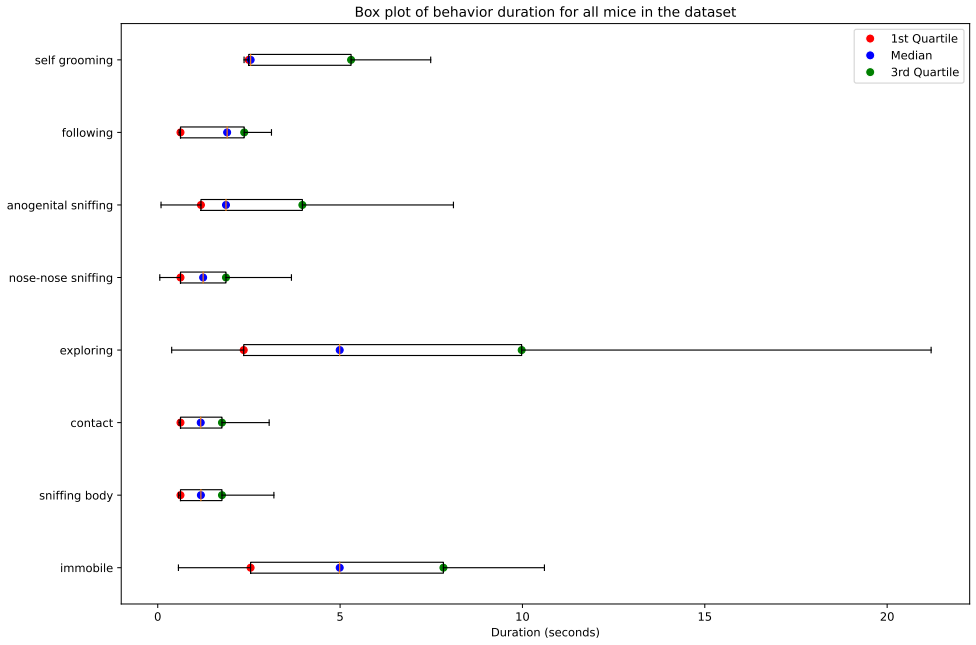



Fig. 4.6 A box plot representation of the sound duration distribution among all mice across different behaviors.

Before deploying our DL models, we ensured the quality and consistency of our dataset by plotting and visualizing the spectrograms. This process included the creation of both a standard view and an inverted grayscale version to spotlight the higher decibel levels (Fig. 4.7). The dataset was then ready for training the DL models, which promised to provide valuable insights into the mapping of rodent USVs to their corresponding behaviors.

We adopted stringent criteria for the creation of the dataset. Any instances where the mice did not exhibit a specific behavior were identified and systematically eliminated. This rigorous process ensured that our dataset was both pure and precise, exclusively representing cases where the mice exhibited discernible behaviors.

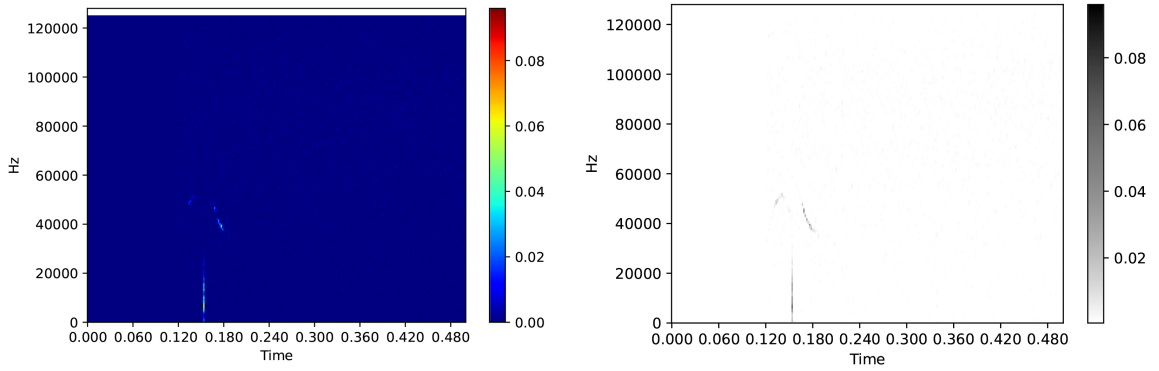



Fig. 4.7 Gray-scaled inverted spectrogram visualization of a mouse ultrasonic vocalization chunk. The high-contrast grayscale representation enhances the visibility of acoustic features in the vocalization.

In relation to the .png datasets, an unexpected complication was observed. A number of images, when produced using the PIL library, appeared as featureless black rectangles. This unusual occurrence was notable given that the corresponding matrices from which these images were derived contained non-null data. This phenomenon was not observed in the images created with the Matplotlib library, suggesting a potential disparity in how these libraries handle image data. Further investigation is required to definitively determine the cause of this issue and develop an appropriate solution.

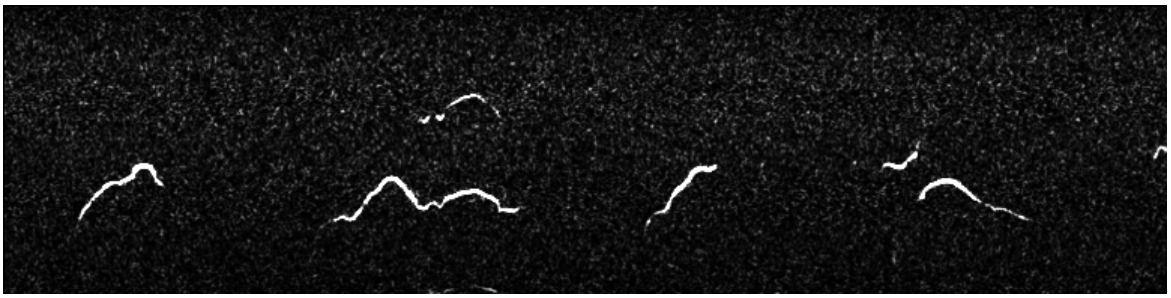


Fig. 4.8 Example of a spectrogram from .png images dataset created with Matplotlib library

### 4.3.2 CNN: 8 convolutional layers

The model (Fig. 4.9) follows a pattern of convolutional, max pooling, batch normalization, and fully connected dense layers, with dropout layers to prevent overfitting. The model parameters, totaling 80,700, are largely trainable (80,268), with a negligible portion (432) being non-trainable.

At the onset, the model employs Conv2D, a spatial convolution over images, resulting in an output shape of None, 176, 3907, 8 (with None being equal to the number of chunks), with 80 parameters. It utilizes 8 filters (indicated by the last digit in the output shape) to extract spatial features. This is followed by another convolution layer with 8 filters and 584 parameters.

Subsequently, the MaxPooling2D layer is implemented, which drastically reduces the spatial dimensions (from 176, 3907 to 88, 1953) by applying a max filter to non-overlapping subregions of the initial representation. This is followed by Batch Normalization, a technique to standardize the inputs to a layer, improving the overall performance and stability of the model.

The model then repeats the pattern of Conv2D layers, MaxPooling, and Batch Normalization, each time increasing the number of filters (16, then 32, then 64) used in the convolution layers and halving the spatial dimensions through max pooling.

After several iterations of this pattern, the model deploys a GlobalAveragePooling2D layer. Unlike Max Pooling, which uses the maximum, this layer takes the average of the spatial dimensions, collapsing them into a single dimension and producing an output of shape None, 64.

At this point, we insert a fully connected (Dense) layer with 64 units and 4160 parameters. Another Batch Normalization layer is introduced here, before a Dropout level, which randomly sets input units to 0 at each update during training to prevent overfitting.

The model concludes with another Dense layer with 32 units and 2080 parameters, a final Batch normalization and Dropout layer, and a final Dense layer with 4 units and 132 parameters.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 176, 3907, 8)      80

 conv2d_1 (Conv2D)           (None, 176, 3907, 8)      584

 max_pooling2d (MaxPooling2  (None, 88, 1953, 8)       0
 D)

 batch_normalization (Batch  (None, 88, 1953, 8)       32
 Normalization)

 conv2d_2 (Conv2D)           (None, 88, 1953, 16)      1168

 conv2d_3 (Conv2D)           (None, 88, 1953, 16)      2320

 max_pooling2d_1 (MaxPoolin  (None, 44, 976, 16)       0
 g2D)

 batch_normalization_1 (Bat  (None, 44, 976, 16)       64
 chNormalization)

 conv2d_4 (Conv2D)           (None, 44, 976, 32)       4640

 conv2d_5 (Conv2D)           (None, 44, 976, 32)       9248

 max_pooling2d_2 (MaxPoolin  (None, 22, 488, 32)       0
 g2D)

 batch_normalization_2 (Bat  (None, 22, 488, 32)       128
 chNormalization)

 conv2d_6 (Conv2D)           (None, 22, 488, 64)       18496

 conv2d_7 (Conv2D)           (None, 22, 488, 64)       36928
```

```
max_pooling2d_3 (MaxPoolin  (None, 11, 244, 64)       0
g2D)

batch_normalization_3 (Bat  (None, 11, 244, 64)       256
chNormalization)

global_average_pooling2d (  (None, 64)                0
GlobalAveragePooling2D)

dense (Dense)               (None, 64)                4160

batch_normalization_4 (Bat  (None, 64)                256
chNormalization)

dropout (Dropout)           (None, 64)                0

dense_1 (Dense)             (None, 32)                2080

batch_normalization_5 (Bat  (None, 32)                128
chNormalization)

dropout_1 (Dropout)         (None, 32)                0

dense_2 (Dense)             (None, 4)                 132

=================================================================
Total params: 80700 (315.23 KB)
Trainable params: 80268 (313.55 KB)
Non-trainable params: 432 (1.69 KB)
_________________________________________________________________
```

Fig. 4.9 Model summary of a 8 convolutional layers network

The training of the model encompassed 100 epochs, although early stopping occurred at the 25th epoch (Fig. 4.10). The model was trained with Tensorflow [104], and CUDA Deep Neural Network library (cuDNN) [105] version 8600 was loaded for GPU-accelerated operations.

During the first epoch, the model achieved an accuracy of 0.2520 with a loss of 2.7811. Initial accuracy on the training set was 0.2597, and the associated loss was 2.2315. This

discrepancy suggests that the model was still in its early stages of learning, with room for optimization.

TensorFloat-32 was used for matrix multiplication, a choice that was logged in the second epoch. This precision enhancement over regular 32-bit floating point format is designed to accelerate training on NVIDIA GPUs specifically.

The model's performance saw gradual improvements: the second epoch ended with a loss of 2.2344 and an accuracy of 0.2598. By the third epoch, the loss decreased to 1.9544, and accuracy increased to 0.2891. This trend of improvement continued until the fifth epoch, where the accuracy dipped to 0.1914. Interestingly, the sixth epoch marked a significant leap in accuracy to 0.4082, with a corresponding loss of 1.6422.

From the seventh epoch onward, the model's accuracy began exhibiting fluctuations, suggestive of a potential overfitting situation. Such a scenario implies the model might be learning the noise within the training data, rather than discerning the inherent pattern. Possible causes for this outcome could stem from the model's complexity compared to the training data's diversity, or from the presence of redundant or irrelevant features in the dataset. Despite these fluctuations in accuracy, the model's loss metric continued to trend downward, reaching a minimum of 1.4189 in the 15th epoch, which also coincided with the peak accuracy of 0.3320.

Following this peak performance, the accuracy began to decline — a clear indicator of overfitting. To prevent further overfitting and unnecessary computation, early stopping was implemented, terminating training at the 25th epoch.

This training session is a clear example of the inherent challenges in AI model training, particularly with regards to balancing accuracy and loss, and managing the risks of overfitting. Future work on this model may involve fine-tuning the hyperparameters, expanding the dataset, or exploring alternative model architectures to improve generalization and prediction on unseen data.

```
Epoch 1/100
2023-09-08 15:21:23.192472: I tensorflow/compiler/xla/stream_executor
                                /cuda/cuda_dnn.cc:432] Loaded
                                cuDNN version 8600
2023-09-08 15:21:25.310361: I tensorflow/compiler/xla/stream_executor
                                /cuda/cuda_blas.cc:606]
                                TensorFloat-32 will be used for
                                the matrix multiplication. This
                                will only be logged once.
```

```
672/672 [==============================] - 30s 34ms/step - loss: 2.
                                    2315 - accuracy: 0.2597 - val_loss
                                    : 2.7811 - val_accuracy: 0.2520
Epoch 2/100
672/672 [==============================] - 19s 29ms/step - loss: 2.
                                    1667 - accuracy: 0.2517 - val_loss
                                    : 2.2344 - val_accuracy: 0.2598
Epoch 3/100
672/672 [==============================] - 19s 29ms/step - loss: 2.
                                    1231 - accuracy: 0.2519 - val_loss
                                    : 1.9544 - val_accuracy: 0.2891
Epoch 4/100
672/672 [==============================] - 19s 28ms/step - loss: 2.
                                    0108 - accuracy: 0.2552 - val_loss
                                    : 2.0331 - val_accuracy: 0.2422
Epoch 5/100
672/672 [==============================] - 36s 53ms/step - loss: 1.
                                    9384 - accuracy: 0.2607 - val_loss
                                    : 1.6818 - val_accuracy: 0.1914
Epoch 6/100
672/672 [==============================] - 19s 28ms/step - loss: 1.
                                    8959 - accuracy: 0.2647 - val_loss
                                    : 1.6422 - val_accuracy: 0.4082
Epoch 7/100
672/672 [==============================] - 19s 28ms/step - loss: 1.
                                    8705 - accuracy: 0.2496 - val_loss
                                    : 1.8510 - val_accuracy: 0.3398
Epoch 8/100
672/672 [==============================] - 19s 28ms/step - loss: 1.
                                    7965 - accuracy: 0.2618 - val_loss
                                    : 1.6927 - val_accuracy: 0.1699
Epoch 9/100
672/672 [==============================] - 19s 29ms/step - loss: 1.
                                    7875 - accuracy: 0.2445 - val_loss
                                    : 1.5670 - val_accuracy: 0.3477
Epoch 10/100
672/672 [==============================] - 19s 28ms/step - loss: 1.
                                    7239 - accuracy: 0.2562 - val_loss
                                    : 1.5212 - val_accuracy: 0.2227
Epoch 11/100
672/672 [==============================] - 19s 28ms/step - loss: 1.
                                    6873 - accuracy: 0.2670 - val_loss
                                    : 1.4784 - val_accuracy: 0.2637
Epoch 12/100
```

```
672/672 [==============================] - 19s 29ms/step - loss: 1.
                                    6811 - accuracy: 0.2615 - val_loss
                                    : 1.5281 - val_accuracy: 0.2773
Epoch 13/100
672/672 [==============================] - 19s 28ms/step - loss: 1.
                                    6617 - accuracy: 0.2577 - val_loss
                                    : 1.7838 - val_accuracy: 0.2715
Epoch 14/100
672/672 [==============================] - 19s 28ms/step - loss: 1.
                                    6396 - accuracy: 0.2539 - val_loss
                                    : 1.5445 - val_accuracy: 0.2695
Epoch 15/100
672/672 [==============================] - 19s 29ms/step - loss: 1.
                                    6104 - accuracy: 0.2713 - val_loss
                                    : 1.4189 - val_accuracy: 0.3320
Epoch 16/100
672/672 [==============================] - 19s 28ms/step - loss: 1.
                                    6148 - accuracy: 0.2644 - val_loss
                                    : 1.4501 - val_accuracy: 0.2754
Epoch 17/100
672/672 [==============================] - 19s 28ms/step - loss: 1.
                                    5916 - accuracy: 0.2513 - val_loss
                                    : 1.5724 - val_accuracy: 0.1953
Epoch 18/100
672/672 [==============================] - 19s 29ms/step - loss: 1.
                                    5824 - accuracy: 0.2630 - val_loss
                                    : 1.4513 - val_accuracy: 0.2637
Epoch 19/100
672/672 [==============================] - 19s 28ms/step - loss: 1.
                                    5788 - accuracy: 0.2596 - val_loss
                                    : 1.4611 - val_accuracy: 0.2715
Epoch 20/100
672/672 [==============================] - 19s 28ms/step - loss: 1.
                                    5698 - accuracy: 0.2657 - val_loss
                                    : 1.5045 - val_accuracy: 0.2852
Epoch 21/100
672/672 [==============================] - 19s 28ms/step - loss: 1.
                                    5471 - accuracy: 0.2715 - val_loss
                                    : 1.4471 - val_accuracy: 0.2480
Epoch 22/100
672/672 [==============================] - 19s 28ms/step - loss: 1.
                                    5454 - accuracy: 0.2642 - val_loss
                                    : 1.4428 - val_accuracy: 0.3340
Epoch 23/100
```

```
672/672 [==============================] - 19s 28ms/step - loss: 1.
                                    5409 - accuracy: 0.2778 - val_loss
                                    : 1.5145 - val_accuracy: 0.2520
Epoch 24/100
672/672 [==============================] - 19s 28ms/step - loss: 1.
                                    5402 - accuracy: 0.2710 - val_loss
                                    : 2.1001 - val_accuracy: 0.2285
Epoch 25/100
671/672 [==============================>.] - ETA: 0s - loss: 1.5370 -
                                    accuracy: 0.2680Restoring model
                                    weights from the end of the best
                                    epoch: 15.
672/672 [==============================] - 19s 29ms/step - loss: 1.
                                    5371 - accuracy: 0.2678 - val_loss
                                    : 1.4238 - val_accuracy: 0.1445
Epoch 25: early stopping
```

Fig. 4.10 Epoch timeline of a 8 convolutional layers network

Fig. 4.11 presents the confusion matrix that pertains to a classification model test set, which is designed to differentiate between four types of behavioral patterns: “anogenital sniffing”, “nose-nose sniffing”, “exploring”, and “sniffing body”. The matrix represents the actual labels versus the predicted labels by the model.

The diagonal elements of the matrix represent accurately classified instances, while off-diagonal elements signify misclassifications. For instance, “anogenital sniffing” was correctly identified 107 times, but was also mistaken for “nose-nose sniffing” 70 times, “exploring” 36 times, and “sniffing body” 4 times.

Similarly, the model predicted “nose-nose sniffing” 25 times correctly but also misclassified it as “anogenital sniffing” 16 times, “exploring” 23 times, and “sniffing body” 4 times. Looking at “exploring”, this behavior was recognized correctly 26 times, however, it was mislabeled as “anogenital sniffing” 61 times, “nose-nose sniffing” 39 times, and “sniffing body” 2 times. For the “sniffing body” behavior, the model gave accurate predictions 12 times and incorrect predictions 30 times for “anogenital sniffing”, 38 times for “nose-nose sniffing”, and 19 times for “exploring”.

The overall test accuracy of the model is 33.2%. This low accuracy suggests that the model has difficulty distinguishing between these four types of behaviors, often confusing one for another. It is indicative of the complex nature of the behavioral patterns under study and the challenges inherent in accurate classification. Further refinement and optimization

of the model are necessary to improve its predictive accuracy. These enhancements could potentially be achieved through the use of larger training datasets, more sophisticated feature extraction, or employing different machine learning algorithms better suited to handling high-dimensional behavioral data. Same results were obtained using .png datasets.

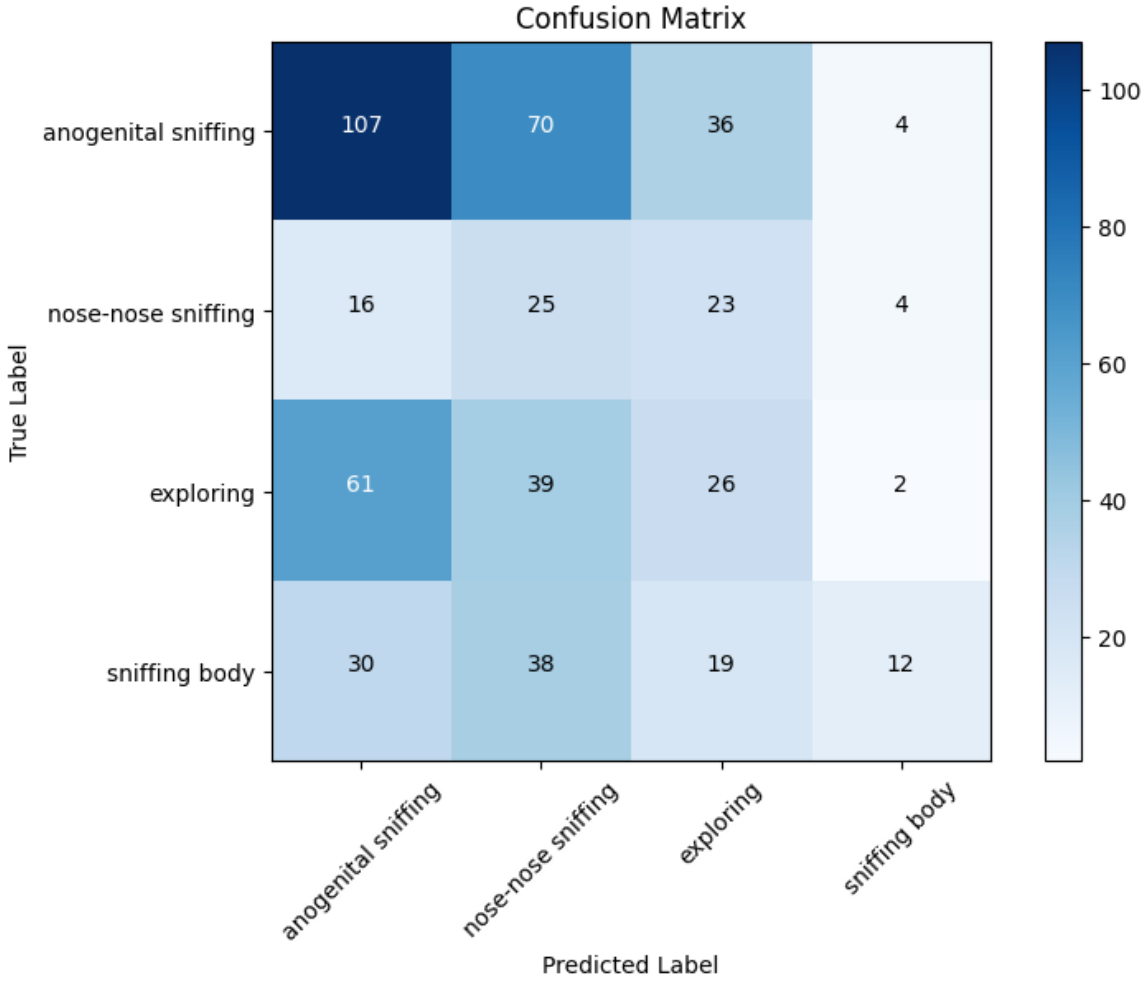



Fig. 4.11 8 layers CNN confusion matrix.

### 4.3.3 CNN: 10 convolutional layers

In order to add a different dimension to our analysis, a model with a higher number of trainable parameters was used (Fig. 4.12). The architecture comprises several Convolutional 2D, Max Pooling, Batch Normalization, and Dense layers, each contributing uniquely to

learning from the input data. The Conv2D layers, with their convolutional operations, enable the model to learn complex pattern hierarchies from the data.

The architecture begins with two Conv2D layers, each outputting a shape of 176x3907x8, with 80 and 584 parameters respectively. These layers function to detect low-level features such as edges, lines, and simple textures. Following the Conv2D layers, a MaxPooling2D layer is applied to reduce spatial dimensions (by half in this case), improving the model's robustness to scale and translation variations and reducing computational complexity. The first Batch Normalization layer is then applied for internal covariate shift reduction [106], aiming to improve the model's training speed and stability.

The pattern of Conv2D pairs followed by MaxPooling2D and Batch Normalization continues, with the number of filters (i.e., feature maps) doubling at each iteration, enabling the detection of increasingly complex patterns. The increase in parameters as the number of filters increases reflects the model's expanding capacity to learn from the data.

Towards the end of the model, a GlobalAveragePooling2D layer is applied, transforming the two-dimensional feature maps into a single vector per feature map. This operation retains the most essential information, reducing the number of parameters and thus the risk of overfitting.

Finally, Dense layers, followed by Batch Normalization and Dropout layers, are used. The Dropout layers help prevent overfitting by randomly setting a fraction of input units to 0 at each update during training time.

In summary, the model architecture demonstrates a careful balance between learning capacity (as evidenced by the Conv2D and Dense layers) and model generalization (as seen with the MaxPooling2D, Dropout, and GlobalAveragePooling2D layers). The model's total parameters amount to 221,564, with 220,940 of these being trainable.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 176, 3907, 8)      80

 conv2d_1 (Conv2D)           (None, 176, 3907, 8)      584

 max_pooling2d (MaxPooling2  (None, 88, 1953, 8)       0
 D)

 batch_normalization (Batch  (None, 88, 1953, 8)       32
```

```
Normalization)

conv2d_2 (Conv2D)           (None, 88, 1953, 16)      1168

conv2d_3 (Conv2D)           (None, 88, 1953, 16)      2320

max_pooling2d_1 (MaxPoolin  (None, 44, 976, 16)       0
g2D)

batch_normalization_1 (Bat  (None, 44, 976, 16)       64
chNormalization)

conv2d_4 (Conv2D)           (None, 44, 976, 32)       4640

conv2d_5 (Conv2D)           (None, 44, 976, 32)       9248

max_pooling2d_2 (MaxPoolin  (None, 22, 488, 32)       0
g2D)

batch_normalization_2 (Bat  (None, 22, 488, 32)       128
chNormalization)

conv2d_6 (Conv2D)           (None, 22, 488, 64)       18496

conv2d_7 (Conv2D)           (None, 22, 488, 64)       36928

max_pooling2d_3 (MaxPoolin  (None, 11, 244, 64)       0
g2D)

batch_normalization_3 (Bat  (None, 11, 244, 64)       256
chNormalization)

conv2d_8 (Conv2D)           (None, 11, 244, 96)       55392

conv2d_9 (Conv2D)           (None, 11, 244, 96)       83040

max_pooling2d_4 (MaxPoolin  (None, 11, 122, 96)       0
g2D)

batch_normalization_4 (Bat  (None, 11, 122, 96)       384
chNormalization)

global_average_pooling2d (  (None, 96)                0
GlobalAveragePooling2D)
```

```
 dense (Dense)                (None, 64)                6208

 batch_normalization_5 (Bat   (None, 64)                256
 chNormalization)

 dropout (Dropout)            (None, 64)                0

 dense_1 (Dense)              (None, 32)                2080

 batch_normalization_6 (Bat   (None, 32)                128
 chNormalization)

 dropout_1 (Dropout)          (None, 32)                0

 dense_2 (Dense)              (None, 4)                 132

=================================================================
Total params: 221564 (865.48 KB)
Trainable params: 220940 (863.05 KB)
Non-trainable params: 624 (2.44 KB)
_________________________________________________________________
```

Fig. 4.12 Model summary of a 10 convolutional layers network

The epoch timeline presented on Fig. 4.13 provides a detailed account of a machine learning model's training process. The model is trained for 26 epochs, with each epoch representing a full pass through the entire training dataset.

In the initial epoch, the model exhibits a considerable loss value of 2.4661 on the training set, reflecting a significant discrepancy between the model's predictions and the actual outcomes. While the accuracy, at 0.2608, indicates that approximately 26% of the model's predictions are correct. The loss and accuracy, 1.8759 and 0.3027 respectively, suggest a similar performance on the unseen data.

As the model continues to learn from the training data, there is a noticeable decrease in the loss value, indicating that the model's predictions are progressively aligning with the actual outcomes. By epoch 5, the training loss has reduced to 2.0568 and the accuracy has marginally improved to 0.2571. In this epoch, the model demonstrates a notable improvement with the loss dropping to 1.5798 and accuracy rising to 0.2617.

However, this linear improvement doesn't continue indefinitely. From epochs 6 to 11, the model seems to oscillate in its performance, with the loss fluctuating between 1.4921 and 1.5513, and the accuracy between 0.2285 and 0.2402. This suggests a possible overfitting to the training data, where the model might be learning noise and outliers, becoming less effective on unseen data.

The model achieves its best performance on the set at epoch 16, where the loss is at its lowest at 1.3980 and accuracy at 0.2812. Despite further training, the model is unable to surpass this performance. The training is halted at epoch 26 due to early stopping - a regularization method to prevent overfitting. The early stopping is triggered as no improvement in loss is observed for 10 consecutive epochs post the best epoch.

This thorough examination of the epoch timeline offers a nuanced understanding of the model's learning journey. While the model demonstrates initial progress in learning from the data, it starts to stagger in its performance later, indicating potential overfitting. Early stopping assists in preventing an over-complicated model and helps in retaining a version that performs best on unseen data.

```
Epoch 1/100
672/672 [==============================] - 23s 33ms/step - loss: 2.
                                    4661 - accuracy: 0.2608 - val_loss
                                    : 1.8759 - val_accuracy: 0.3027
Epoch 2/100
672/672 [==============================] - 19s 29ms/step - loss: 2.
                                    3493 - accuracy: 0.2601 - val_loss
                                    : 1.9576 - val_accuracy: 0.2832
Epoch 3/100
672/672 [==============================] - 19s 29ms/step - loss: 2.
                                    2481 - accuracy: 0.2437 - val_loss
                                    : 2.1163 - val_accuracy: 0.2148
Epoch 4/100
672/672 [==============================] - 20s 30ms/step - loss: 2.
                                    1364 - accuracy: 0.2504 - val_loss
                                    : 2.0035 - val_accuracy: 0.2168
Epoch 5/100
672/672 [==============================] - 19s 29ms/step - loss: 2.
                                    0568 - accuracy: 0.2571 - val_loss
                                    : 1.5798 - val_accuracy: 0.2617
Epoch 6/100
672/672 [==============================] - 19s 29ms/step - loss: 2.
                                    0071 - accuracy: 0.2525 - val_loss
                                    : 1.4921 - val_accuracy: 0.2285
```

```
Epoch 7/100
672/672 [==============================] - 19s 29ms/step - loss: 1.
                                    9579 - accuracy: 0.2475 - val_loss
                                    : 1.4809 - val_accuracy: 0.2676
Epoch 8/100
672/672 [==============================] - 19s 29ms/step - loss: 1.
                                    8993 - accuracy: 0.2530 - val_loss
                                    : 1.5077 - val_accuracy: 0.3711
Epoch 9/100
672/672 [==============================] - 19s 29ms/step - loss: 1.
                                    8376 - accuracy: 0.2558 - val_loss
                                    : 1.4064 - val_accuracy: 0.3262
Epoch 10/100
672/672 [==============================] - 19s 29ms/step - loss: 1.
                                    8637 - accuracy: 0.2547 - val_loss
                                    : 1.5009 - val_accuracy: 0.3008
Epoch 11/100
672/672 [==============================] - 19s 28ms/step - loss: 1.
                                    7543 - accuracy: 0.2707 - val_loss
                                    : 1.5513 - val_accuracy: 0.2402
Epoch 12/100
672/672 [==============================] - 20s 29ms/step - loss: 1.
                                    7361 - accuracy: 0.2732 - val_loss
                                    : 1.5111 - val_accuracy: 0.2520
Epoch 13/100
672/672 [==============================] - 20s 29ms/step - loss: 1.
                                    7004 - accuracy: 0.2548 - val_loss
                                    : 1.6189 - val_accuracy: 0.2949
Epoch 14/100
672/672 [==============================] - 20s 29ms/step - loss: 1.
                                    7003 - accuracy: 0.2617 - val_loss
                                    : 1.4191 - val_accuracy: 0.2480
Epoch 15/100
672/672 [==============================] - 20s 30ms/step - loss: 1.
                                    6736 - accuracy: 0.2708 - val_loss
                                    : 1.4207 - val_accuracy: 0.2852
Epoch 16/100
672/672 [==============================] - 20s 29ms/step - loss: 1.
                                    6636 - accuracy: 0.2571 - val_loss
                                    : 1.3980 - val_accuracy: 0.2812
Epoch 17/100
672/672 [==============================] - 19s 29ms/step - loss: 1.
                                    6472 - accuracy: 0.2531 - val_loss
                                    : 1.4470 - val_accuracy: 0.3184
```

```
Epoch 18/100
672/672 [==============================] - 19s 29ms/step - loss: 1.
                                    6185 - accuracy: 0.2656 - val_loss
                                    : 1.4793 - val_accuracy: 0.2695
Epoch 19/100
672/672 [==============================] - 19s 29ms/step - loss: 1.
                                    6183 - accuracy: 0.2659 - val_loss
                                    : 1.5370 - val_accuracy: 0.2930
Epoch 20/100
672/672 [==============================] - 20s 29ms/step - loss: 1.
                                    5995 - accuracy: 0.2678 - val_loss
                                    : 1.4643 - val_accuracy: 0.3008
Epoch 21/100
672/672 [==============================] - 19s 29ms/step - loss: 1.
                                    5869 - accuracy: 0.2758 - val_loss
                                    : 1.4788 - val_accuracy: 0.3008
Epoch 22/100
672/672 [==============================] - 20s 30ms/step - loss: 1.
                                    5712 - accuracy: 0.2736 - val_loss
                                    : 1.4068 - val_accuracy: 0.3340
Epoch 23/100
672/672 [==============================] - 20s 29ms/step - loss: 1.
                                    5559 - accuracy: 0.2730 - val_loss
                                    : 1.4564 - val_accuracy: 0.3164
Epoch 24/100
672/672 [==============================] - 21s 31ms/step - loss: 1.
                                    5502 - accuracy: 0.2860 - val_loss
                                    : 1.5483 - val_accuracy: 0.2793
Epoch 25/100
672/672 [==============================] - 19s 29ms/step - loss: 1.
                                    5551 - accuracy: 0.2799 - val_loss
                                    : 1.4865 - val_accuracy: 0.2852
Epoch 26/100
671/672 [==============================>.] - ETA: 0s - loss: 1.5580 -
                                    accuracy: 0.2677Restoring model
                                    weights from the end of the best
                                    epoch: 16.
672/672 [==============================] - 20s 29ms/step - loss: 1.
                                    5580 - accuracy: 0.2676 - val_loss
                                    : 1.5547 - val_accuracy: 0.3281
Epoch 26: early stopping
```

Fig. 4.13 Epoch timeline of a 10 convolutional layers network

The Fig. 4.14 provides a robust and detailed assessment of the performance of our predictive model, specifically highlighting the accuracy of the model's predictions in four distinctive categories of behavior.

Each row of this matrix represents the true labels while each column represents the predicted labels. In the first category, "anogenital sniffing", the model predicted correctly in 53 instances, where the true label and the predicted label aligned. However, there were cases of misclassification. 28 instances were incorrectly classified as "nose-nose sniffing", and 33 instances were wrongly categorized as "exploring", and 103 instances were misclassified as "sniffing body".

In the "nose-nose sniffing" category, the model's performance was less accurate. 24 instances were correctly classified, while the remaining were misclassified in various categories. Similarly, the model's performance on "exploring" and "sniffing body" classes demonstrated a similar pattern of correct classifications and misclassification.

The test accuracy for the model is 35.7%. This percentage represents the ratio of correctly predicted instances to the total instances in the dataset. The accuracy gives us a rudimentary understanding of the model's performance but does not provide a comprehensive description. The detailed breakdown of the performance across different categories provided by the confusion matrix offers a more comprehensive view. Same results were obtained using .png datasets.

In conclusion, these results highlight the strengths and weaknesses of the current model. The data suggests that the model performs differently across the various categories, providing a potential area for further research and model refinement. By addressing the misclassification issues, the model's performance and accuracy could be significantly improved.

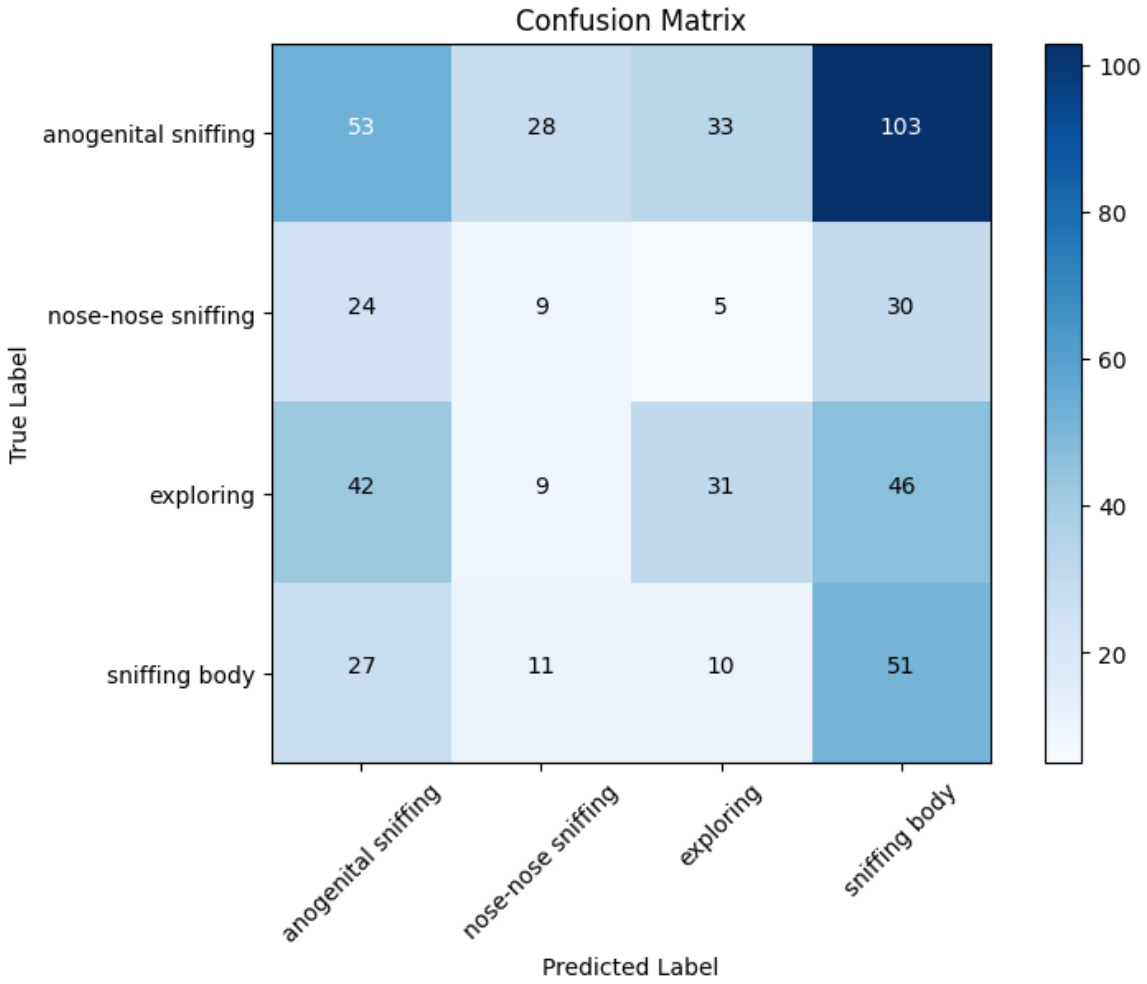



Fig. 4.14 10 layers CNN confusion matrix.

# Chapter 5

# Conclusion

## 5.1 Evaluating the Accuracy of Machine Learning Algorithms for Detecting Frontotemporal Lobar Degeneration: Conclusion

The MRI classification between healthy and FTD-diagnosed subjects is a promising technique that can help experts in providing an early diagnosis which could be beneficial for the life quality of a patient. In this work, we explored the performance of Support Vector Machine and Random Forest to formulate a correct FTD diagnosis. The dataset consisted of MRI data acquired on 30 subjects, and each set of data was elaborated with different MRI techniques based on a Multi-Voxel Pattern Analysis approach. The results obtained were validated by a Stratified Shuffle Split process and are quite consistent across the methodologies, even if Random Forest achieved the best accuracy. When compared our results with those of Premi et al. [4], a 4% accuracy improvement was obtained when using the VMHC technique. An increase in the original size of the dataset may lead to an interesting application of Deep Learning algorithmic techniques, with the expectation of obtaining better results in terms of overall test accuracy.

## 5.2 Ultrasonic vocalizations in mice and correlation with their behavior: conclusion

Our investigation into the ultrasonic communication between two groups of mice, the CS and VH mice, has yielded notable findings. The analysis presented in this PhD thesis aids in the comprehensive understanding of the distinct USV communication differences between

these two groups and their correlation with specific behaviors. Our detailed examination involved recording and analyzing the distribution of sounds and patterns of consecutive syllables. What we discovered is a clear divergence in the sonic patterns between the two groups, reinforcing the idea that ultrasonic communication is deeply intertwined with the social interaction dynamics among mice. Along with the sonic patterns, we also observed disparities in the behaviors of the two groups. This highlights a robust connection between the general sounds produced and the corresponding behavioral context.

These findings provide an impetus for further research. The correlation between ultrasonic communication and behavior in mice, as demonstrated in this study, is a fascinating area of exploration that has significant potential for understanding the intricacies of animal communication. Moreover, the results of this study emphasize the importance of viewing animal communication from a holistic perspective, where both the acoustic features and the associated behaviors are considered.

In light of the possible intersections between AI and behavioral neuroscience, insights from this study might help enrich our understanding of animal behavior. This could foster the development of sophisticated instruments for behavioral analysis. There remains a potential for these methods to contribute to human conditions analysis by examining acoustic communication patterns in the future.

## 5.3 Dataset preparation and preliminary machine learning results: conclusion

In conclusion, this research successfully established a rigorous and methodical process for the collection, preprocessing, and analysis of USVs from male mice, with meticulous attention dedicated to the simultaneous recording of corresponding behavioral data. The judicious use of high-frequency microphones ensured the quality and completeness of the audio data, while the consistent and careful annotation of behaviors allowed for a high level of granularity in the subsequent analysis.

The preprocessing stage employed sophisticated techniques to extract meaningful information from the raw audio recordings. The application of the STFT algorithm was particularly significant in this regard, facilitating the conversion of non-stationary signals into structured spectrograms. The selection of parameters for STFT was a crucial aspect of the methodology, with the chosen values demonstrating a successful balance between resolution and computational efficiency.

The resulting spectrograms were then paired with relevant behavioral data, a task that necessitated precise alignment of the two data types. The thresholding of behaviors by duration served to eliminate potential noise in the dataset, focusing the analysis on substantial and behaviorally relevant USVs. Moreover, the application of a one-hot encoding scheme to the behavioral data ensured that this information could be readily incorporated into the DL models.

Additionally, the recognition of inherent variability in behavioral responses among the mice subjects underscored the complexity of the study. This variability, while introducing additional challenges, also emphasized the potential for rich insights that could be gained from the successful mapping of USVs to behaviors.

In the course of this exploratory research, it was observed that the correlation between mice USVs and their respective behaviors yielded less than optimal results when analyzed using CNN. The data did not present a clear or strong correlation, which could be attributed to a multitude of factors including the complexity of animal behavior, the intricacies of USVs, and the limitations of the current methodological approach.

However, this does not conclusively negate the existence of a potential relationship between mice USVs and behavior. It is important to note that these preliminary findings serve as a stepping stone to further investigations, rather than a definitive conclusion. Future research endeavors may benefit from the application of alternative preprocessing techniques or the utilization of distinct deep learning methodologies.

Deep learning techniques offer a broad spectrum of applications and have proven effective in tackling complex and non-linear relationships. Exploring different architectures or training methods could potentially improve the performance. For instance, recurrent neural networks which are designed to recognize patterns over time, or transformer-based models which have shown promise in understanding sequence data, could be employed.

Furthermore, it is also worth considering a more comprehensive preprocessing stage. This could involve more nuanced feature extraction or noise reduction techniques, aimed at refining the quality of the input data fed into the model.

In conclusion, while the initial findings from the CNN application were not as anticipated, the potential for uncovering significant correlations between mice USVs and behavior remains promising. This study underscores the need for continued exploration, experimentation, and refinement in the quest for understanding the complex interplay between mouse vocalization and behavior.

Future research can explore the following:

- **Machine Learning Integration**: Automating behavior classification based on vocalizations for real-time insights.

- **Contextual Analysis**: Investigating how environmental factors affect communication dynamics.
- **Longitudinal Studies**: Tracking behavioral changes over time to understand social interactions.
- **Cross-Species Comparisons**: Extending our approach to compare communication across species, revealing commonalities and differences.

# References


[1] F. Chollet, *Deep learning mit python und keras: das praxis-handbuch vom entwickler der keras-bibliothek*. MITP-Verlags GmbH & Co. KG, 2018.

[2] PubChem, "Cannabidiol, national library of medicine." https://pubchem.ncbi.nlm.nih.gov/compound/Cannabidiol, 2022.

[3] M. L. Scattoni, S. U. Gandhy, L. Ricceri, and J. N. Crawley, "Unusual repertoire of vocalizations in the BTBR T+ tf/J mouse model of autism," *PloS one*, vol. 3, no. 8, p. e3067, 2008.

[4] E. Premi, F. Cauda, T. Costa, M. Diano, S. Gazzina, V. Gualeni, A. Alberici, S. Archetti, M. Magoni, R. Gasparotti, *et al.*, "Looking for neuroimaging markers in frontotemporal lobar degeneration clinical trials: a multi-voxel pattern analysis study in granulin disease," *Journal of Alzheimer's Disease*, vol. 51, no. 1, pp. 249–262, 2016.

[5] D. S. Knopman and R. O. Roberts, "Estimating the number of persons with frontotemporal lobar degeneration in the US population," *Journal of Molecular Neuroscience*, vol. 45, no. 3, pp. 330–335, 2011.

[6] N. T. Olney, S. Spina, and B. L. Miller, "Frontotemporal dementia," *Neurologic clinics*, vol. 35, no. 2, pp. 339–374, 2017.

[7] B. F. Boeve, A. L. Boxer, F. Kumfor, Y. Pijnenburg, and J. D. Rohrer, "Advances and controversies in frontotemporal dementia: diagnosis, biomarkers, and therapeutic considerations," *The Lancet Neurology*, vol. 21, no. 3, pp. 258–272, 2022.

[8] Q. Yu, Y. Mai, Y. Ruan, Y. Luo, L. Zhao, W. Fang, Z. Cao, Y. Li, W. Liao, S. Xiao, *et al.*, "An MRI-based strategy for differentiation of frontotemporal dementia and alzheimer's disease," *Alzheimer's research & therapy*, vol. 13, pp. 1–12, 2021.

[9] S. L. Risacher and L. G. Apostolova, "Neuroimaging in dementia," *CONTINUUM: Lifelong Learning in Neurology*, vol. 29, no. 1, pp. 219–254, 2023.

[10] D. A. Pisner and D. M. Schnyer, "Support vector machine," in *Machine learning*, pp. 101–121, Elsevier, 2020.

[11] A. Parmar, R. Katariya, and V. Patel, "A review on random forest: An ensemble classifier," in *International conference on intelligent data communication technologies and internet of things (ICICI) 2018*, pp. 758–763, Springer, 2019.

[12] M. Wöhr, T. M. Kisko, and R. K. Schwarting, "Social behavior and ultrasonic vocalizations in a genetic rat model haploinsufficient for the cross-disorder risk gene cacna1c," *Brain Sciences*, vol. 11, no. 6, p. 724, 2021.

[13] L. A. Screven and M. L. Dent, "Social isolation produces no effect on ultrasonic vocalization production in adult female cba/caj mice," *PLoS One*, vol. 14, no. 3, p. e0213068, 2019.

[14] F. de Chaumont, N. Lemière, S. Coqueran, T. Bourgeron, and E. Ey, "LMT USV toolbox, a novel methodological approach to place mouse ultrasonic vocalizations in their behavioral contexts—a study in female and male c57bl/6j mice and in shank3 mutant females," *Frontiers in behavioral neuroscience*, p. 241, 2021.

[15] J. Chabout, A. Sarkar, S. R. Patel, T. Radden, D. B. Dunson, S. E. Fisher, and E. D. Jarvis, "A foxp2 mutation implicated in human speech deficits alters sequencing of ultrasonic vocalizations in adult male mice," *Frontiers in behavioral neuroscience*, vol. 10, p. 197, 2016.

[16] D. T. Sangiamo, M. R. Warren, and J. P. Neunuebel, "Ultrasonic signals associated with different types of social behavior of mice," *Nature neuroscience*, vol. 23, no. 3, pp. 411–422, 2020.

[17] S. Hertz, B. Weiner, N. Perets, and M. London, "Temporal structure of mouse courtship vocalizations facilitates syllable labeling," *Communications biology*, vol. 3, no. 1, pp. 1–13, 2020.

[18] S. Von Merten, S. Hoier, C. Pfeifle, and D. Tautz, "A role for ultrasonic vocalisation in social communication and divergence of natural populations of the house mouse (mus musculus domesticus)," *PloS one*, vol. 9, no. 5, p. e97244, 2014.

[19] K. Schulze, F. Vargha-Khadem, and M. Mishkin, "Phonological working memory and foxp2," *Neuropsychologia*, vol. 108, pp. 147–152, 2018.

[20] F. Cabitza and G. Banfi, "Machine learning in laboratory medicine: waiting for the flood?," *Clinical Chemistry and Laboratory Medicine (CCLM)*, vol. 56, no. 4, pp. 516–524, 2018.

[21] A. J. Hughes, S. E. Daniel, Y. Ben-Shlomo, and A. J. Lees, "The accuracy of diagnosis of parkinsonian syndromes in a specialist movement disorder service," *Brain*, vol. 125, no. 4, pp. 861–870, 2002.

[22] C. Barbieri, M. Molina, P. Ponce, M. Tothova, I. Cattinelli, J. I. Titapiccolo, F. Mari, C. Amato, F. Leipold, W. Wehmeyer, *et al.*, "An international observational study suggests that artificial intelligence for clinical decision support optimizes anemia management in hemodialysis patients," *Kidney international*, vol. 90, no. 2, pp. 422–429, 2016.

[23] C. Barbieri, L. Neri, M. Chermisi, E. Bolzoni, I. Cattinelli, W. Decker, S. Stuard, J. D. Martín-Guerrero, and F. Mari, "How to assess the risks associated with the usage of a medical device based on predictive modeling: the case of an anemia control model certified as medical device," *Expert Review of Medical Devices*, vol. 18, no. 11, pp. 1117–1121, 2021.

[24] F. Hatib, Z. Jian, S. Buddi, C. Lee, J. Settels, K. Sibert, J. Rinehart, and M. Cannesson, "Machine-learning algorithm to predict hypotension based on high-fidelity arterial pressure waveform analysis," *Anesthesiology*, vol. 129, no. 4, pp. 663–674, 2018.

[25] B. J. Mortazavi, N. S. Downing, E. M. Bucholz, K. Dharmarajan, A. Manhapra, S.-X. Li, S. N. Negahban, and H. M. Krumholz, "Analysis of machine learning techniques for heart failure readmissions," *Circulation: Cardiovascular Quality and Outcomes*, vol. 9, no. 6, pp. 629–640, 2016.

[26] J. G. Nam, S. Park, E. J. Hwang, J. H. Lee, K.-N. Jin, K. Y. Lim, T. H. Vu, J. H. Sohn, S. Hwang, J. M. Goo, *et al.*, "Development and validation of deep learning–based automatic detection algorithm for malignant pulmonary nodules on chest radiographs," *Radiology*, vol. 290, no. 1, pp. 218–228, 2019.

[27] M. Silva, C. M. Schaefer-Prokop, C. Jacobs, G. Capretti, F. Ciompi, B. van Ginneken, U. Pastorino, and N. Sverzellati, "Detection of subsolid nodules in lung cancer screening: complementary sensitivity of visual reading and computer-aided diagnosis," *Investigative Radiology*, vol. 53, no. 8, pp. 441–449, 2018.

[28] P. Lakhani and B. Sundaram, "Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks," *Radiology*, vol. 284, no. 2, pp. 574–582, 2017.

[29] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, *et al.*, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *arXiv preprint arXiv:1711.05225*, 2017.

[30] E. J. Hwang, S. Park, K.-N. Jin, J. Im Kim, S. Y. Choi, J. H. Lee, J. M. Goo, J. Aum, J.-J. Yim, J. G. Cohen, *et al.*, "Development and validation of a deep learning–based automated detection algorithm for major thoracic diseases on chest radiographs," *JAMA network open*, vol. 2, no. 3, pp. e191095–e191095, 2019.

[31] D. Fooshee, A. Mood, E. Gutman, M. Tavakoli, G. Urban, F. Liu, N. Huynh, D. Van Vranken, and P. Baldi, "Deep learning for chemical reaction prediction," *Molecular Systems Design & Engineering*, vol. 3, no. 3, pp. 442–452, 2018.

[32] W. Jin, C. Coley, R. Barzilay, and T. Jaakkola, "Predicting organic reaction outcomes with Weisfeiler-Lehman Network," *Advances in neural information processing systems*, vol. 30, 2017.

[33] M. H. Segler, M. Preuss, and M. P. Waller, "Planning chemical syntheses with deep neural networks and symbolic AI," *Nature*, vol. 555, no. 7698, pp. 604–610, 2018.

[34] E. Martin, P. Mukherjee, D. Sullivan, and J. Jansen, "Profile-QSAR: a novel meta-QSAR method that combines activities across the kinase family to accurately predict affinity, selectivity, and cellular activity," *Journal of chemical information and modeling*, vol. 51, no. 8, pp. 1942–1956, 2011.

[35] E. B. Lenselink, N. Ten Dijke, B. Bongers, G. Papadatos, H. W. Van Vlijmen, W. Kowalczyk, A. P. IJzerman, and G. J. Van Westen, "Beyond the hype: deep neural networks outperform established methods using a ChEMBL bioactivity benchmark set," *Journal of cheminformatics*, vol. 9, no. 1, pp. 1–14, 2017.

[36] A. Korotcov, V. Tkachenko, D. P. Russo, and S. Ekins, "Comparison of deep learning with multiple machine learning methods and metrics using diverse drug discovery data sets," *Molecular pharmaceutics*, vol. 14, no. 12, pp. 4462–4475, 2017.

[37] M. Hartenfeller and G. Schneider, "Enabling future drug discovery by de novo design," *Wiley Interdisciplinary Reviews: Computational Molecular Science*, vol. 1, no. 5, pp. 742–759, 2011.

[38] A. Kadurin, A. Aliper, A. Kazennov, P. Mamoshina, Q. Vanhaelen, K. Khrabrov, and A. Zhavoronkov, "The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology," *Oncotarget*, vol. 8, no. 7, p. 10883, 2017.

[39] A. Gupta, A. T. Müller, B. J. Huisman, J. A. Fuchs, P. Schneider, and G. Schneider, "Generative recurrent networks for de novo drug design," *Molecular informatics*, vol. 37, no. 1-2, p. 1700111, 2018.

[40] M. Vidal, N. Wolf, B. Rosenberg, B. P. Harris, and A. Mathis, "Perspectives on individual animal identification from biology and computer vision," *Integrative and Comparative Biology*, vol. 61, no. 3, pp. 900–916, 2021.

[41] M. Gharagozloo, A. Amrani, K. Wittingstall, A. Hamilton-Wright, and D. Gris, "Machine learning in modeling of mouse behavior," *Frontiers in neuroscience*, vol. 15, 2021.

[42] B. Q. Geuther, S. P. Deats, K. J. Fox, S. A. Murray, R. E. Braun, J. K. White, E. J. Chesler, C. M. Lutz, and V. Kumar, "Robust mouse tracking in complex environments using neural networks," *Communications biology*, vol. 2, no. 1, pp. 1–11, 2019.

[43] J. M. Wotton, E. Peterson, L. Anderson, S. A. Murray, R. E. Braun, E. J. Chesler, J. K. White, and V. Kumar, "Machine learning-based automated phenotyping of inflammatory nocifensive behavior in mice," *Molecular pain*, vol. 16, p. 1744806920958596, 2020.

[44] K. R. Coffey, R. G. Marx, and J. F. Neumaier, "DeepSqueak: a deep learning-based system for detection and analysis of ultrasonic vocalizations," *Neuropsychopharmacology*, vol. 44, no. 5, pp. 859–868, 2019.

[45] J. J. Valletta, C. Torney, M. Kings, A. Thornton, and J. Madden, "Applications of machine learning in animal behaviour studies," *Animal Behaviour*, vol. 124, pp. 203–220, 2017.

[46] A. Liu and B. Ziebart, "Robust classification under sample selection bias," *Advances in neural information processing systems*, vol. 27, 2014.

[47] T. Yamauchi, T. Yoshioka, D. Yamada, T. Hamano, M. Ohashi, M. Matsumoto, K. Iio, M. Ikeda, M. Kamei, T. Otsuki, *et al.*, "Cold-restraint stress-induced ultrasonic vocalization as a novel tool to measure anxiety in mice," *Biological and Pharmaceutical Bulletin*, vol. 45, no. 3, pp. 268–275, 2022.

[48] A. C. Niemczura, J. M. Grimsley, C. Kim, A. Alkhawaga, A. Poth, A. Carvalho, and J. J. Wenstrup, "Physiological and behavioral responses to vocalization playback in mice," *Frontiers in Behavioral Neuroscience*, vol. 14, p. 155, 2020.

[49] I. Urits, K. Gress, K. Charipova, K. Habib, D. Lee, C. Lee, J. W. Jung, H. Kassem, E. Cornett, A. Paladini, *et al.*, "Use of cannabidiol (CBD) for the treatment of chronic pain," *Best Practice & Research Clinical Anaesthesiology*, vol. 34, no. 3, pp. 463–477, 2020.

[50] E. M. Blessing, M. M. Steenkamp, J. Manzanares, and C. R. Marmar, "Cannabidiol as a potential treatment for anxiety disorders," *Neurotherapeutics*, vol. 12, no. 4, pp. 825–836, 2015.

[51] C. P. Grof, "Cannabis, from plant to pill," *British journal of clinical pharmacology*, vol. 84, no. 11, pp. 2463–2467, 2018.

[52] R. Mechoulam and L. Hanuš, "Cannabidiol: an overview of some chemical and pharmacological aspects. part i: chemical aspects," *Chemistry and physics of lipids*, vol. 121, no. 1-2, pp. 35–43, 2002.

[53] S. Atalay, I. Jarocka-Karpowicz, and E. Skrzydlewska, "Antioxidative and anti-inflammatory properties of cannabidiol," *Antioxidants*, vol. 9, no. 1, p. 21, 2020.

[54] E. Gonca and F. Darıcı, "The effect of cannabidiol on ischemia/reperfusion-induced ventricular arrhythmias: the role of adenosine a1 receptors," *Journal of cardiovascular pharmacology and therapeutics*, vol. 20, no. 1, pp. 76–83, 2015.

[55] C. Martínez-Aguirre, F. Carmona-Cruz, A. L. Velasco, F. Velasco, G. Aguado-Carrillo, M. Cuéllar-Herrera, and L. Rocha, "Cannabidiol acts at 5-HT1A receptors in the human brain: relevance for treating temporal lobe epilepsy," *Frontiers in Behavioral Neuroscience*, vol. 14, p. 611278, 2020.

[56] S. Jeong, H. K. Yun, Y. A. Jeong, M. J. Jo, S. H. Kang, J. L. Kim, D. Y. Kim, S. H. Park, B. R. Kim, Y. J. Na, *et al.*, "Cannabidiol-induced apoptosis is mediated by activation of noxa in human colorectal cancer cells," *Cancer letters*, vol. 447, pp. 12–23, 2019.

[57] M. S. García-Gutiérrez, F. Navarrete, A. Gasparyan, A. Austrich-Olivares, F. Sala, and J. Manzanares, "Cannabidiol: a potential new alternative for the treatment of anxiety, depression, and psychotic disorders," *Biomolecules*, vol. 10, no. 11, p. 1575, 2020.

[58] A. Suraev, R. R. Grunstein, N. S. Marshall, A. L. D'Rozario, C. J. Gordon, D. J. Bartlett, K. Wong, B. J. Yee, R. Vandrey, C. Irwin, *et al.*, "Cannabidiol (CBD) and $\delta$9-tetrahydrocannabinol (THC) for chronic insomnia disorder ('CANSLEEP' trial): protocol for a randomised, placebo-controlled, double-blinded, proof-of-concept trial," *BMJ open*, vol. 10, no. 5, p. e034421, 2020.

[59] C. Alexander and M. Vasefi, "Cannabidiol and the corticoraphe circuit in post-traumatic stress disorder," *IBRO Neuroscience Reports*, vol. 11, pp. 88–102, 2021.

[60] A. Arzimanoglou, U. Brandl, J. H. Cross, A. Gil-Nagel, L. Lagae, C. J. Landmark, N. Specchio, R. Nabbout, E. A. Thiele, O. Gubbay, *et al.*, "Epilepsy and cannabidiol: a guide to treatment," *Epileptic Disorders*, vol. 22, no. 1, pp. 1–14, 2020.

[61] F. Patricio, A. A. Morales-Andrade, A. Patricio-Martínez, and I. D. Limón, "Cannabidiol as a therapeutic target: evidence of its neuroprotective and neuromodulatory function in parkinson's disease," *Frontiers in Pharmacology*, vol. 11, p. 595635, 2020.

[62] G. Watt and T. Karl, "In vivo evidence for therapeutic properties of cannabidiol (CBD) for alzheimer's disease," *Frontiers in pharmacology*, vol. 8, p. 20, 2017.

[63] É. Jones and S. Vlachou, "A critical review of the role of the cannabinoid compounds $\delta$9-tetrahydrocannabinol ($\delta$9-THC) and cannabidiol (CBD) and their combination in multiple sclerosis treatment," *Molecules*, vol. 25, no. 21, p. 4930, 2020.

[64] S. Inglet, B. Winter, S. E. Yost, S. Entringer, A. Lian, M. Biksacky, R. D. Pitt, and W. Mortensen, "Clinical data for the use of cannabis-based treatments: a comprehensive review of the literature," *Annals of Pharmacotherapy*, vol. 54, no. 11, pp. 1109–1143, 2020.

[65] P. Balachandran, M. Elsohly, and K. P. Hill, "Cannabidiol interactions with medications, illicit substances, and alcohol: a comprehensive review," *Journal of general internal medicine*, vol. 36, no. 7, pp. 2074–2084, 2021.

[66] D. Barchel, O. Stolar, T. De-Haan, T. Ziv-Baran, N. Saban, D. O. Fuchs, G. Koren, and M. Berkovitch, "Oral cannabidiol use in children with autism spectrum disorder to treat related symptoms and co-morbidities," *Frontiers in pharmacology*, vol. 9, p. 1521, 2019.

[67] A. K. Birnbaum, A. Karanam, S. E. Marino, C. M. Barkley, R. P. Remmel, M. Roslawski, M. Gramling-Aden, and I. E. Leppik, "Food effect on pharmacokinetics of cannabidiol oral capsules in adult patients with refractory epilepsy," *Epilepsia*, vol. 60, no. 8, pp. 1586–1592, 2019.

[68] FDA, "FDA approves first drug comprised of an active ingredient derived from marijuana to treat rare, severe forms of epilepsy." https://www.fda.gov/news-events/press-announcements/fda-approves-first-drug-comprised-active-ingredient-derived-marijuana-treat-rare-severe-forms, 2018.

[69] W. Hall, "Medical use of cannabis and cannabinoids. questions and answers for policymaking 2018," *Publications Office of the European Union, Luxembourg*, 2018.

[70] C. Guiney, "Cannabis legislation in europe: an overview.," *Drugnet Ireland*, pp. 10–11, 2017.

[71] C. Regulation, "Council regulation (EEC) no. 2309/93 laying down community procedures for the authorization and supervision of medicinal products for human and veterinary use and establishing a european agency for the evaluation of medicinal products," *Official Journal of the European Communities*, vol. 214, pp. 0001–0021, 1993.

[72] P. Brunetti, A. F. L. Faro, F. Pirani, P. Berretta, R. Pacifici, S. Pichini, and F. P. Busardò, "Pharmacology and legal status of cannabidiol.," *Annali dell'Istituto Superiore di Sanità*, vol. 56, no. 3, pp. 285–291, 2020.

[73] B. Arosio, C. Abbate, D. Galimberti, P. D. Rossi, S. Inglese, C. Fenoglio, E. Ridolfi, C. Gussago, M. Casati, E. Tedone, *et al.*, "GRN Thr272fs clinical heterogeneity: a case with atypical late onset presenting with a dementia with Lewy bodies phenotype," *Journal of Alzheimer's Disease*, vol. 35, no. 4, pp. 669–674, 2013.

[74] Q.-H. Zou, C.-Z. Zhu, Y. Yang, X.-N. Zuo, X.-Y. Long, Q.-J. Cao, Y.-F. Wang, and Y.-F. Zang, "An improved approach to detection of amplitude of low-frequency fluctuation (ALFF) for resting-state fMRI: fractional ALFF," *Journal of neuroscience methods*, vol. 172, no. 1, pp. 137–141, 2008.

[75] X.-N. Zuo, C. Kelly, A. Di Martino, M. Mennes, D. S. Margulies, S. Bangaru, R. Grzadzinski, A. C. Evans, Y.-F. Zang, F. X. Castellanos, *et al.*, "Growing together and growing apart: regional and sex differences in the lifespan developmental trajectories of functional homotopy," *Journal of Neuroscience*, vol. 30, no. 45, pp. 15034–15043, 2010.

[76] R. L. Buckner, J. Sepulcre, T. Talukdar, F. M. Krienen, H. Liu, T. Hedden, J. R. Andrews-Hanna, R. A. Sperling, and K. A. Johnson, "Cortical hubs revealed by intrinsic functional connectivity: mapping, assessment of stability, and relation to Alzheimer's disease," *Journal of neuroscience*, vol. 29, no. 6, pp. 1860–1873, 2009.

[77] Z. Zhang, Y. Liu, T. Jiang, B. Zhou, N. An, H. Dai, P. Wang, Y. Niu, L. Wang, and X. Zhang, "Altered spontaneous activity in Alzheimer's disease and mild cognitive impairment revealed by regional homogeneity," *Neuroimage*, vol. 59, no. 2, pp. 1429–1440, 2012.

[78] T. Pilipenko, A. Gnutti, A. Silvestri, I. Serina, and R. Leonardi, "ML Techniques for MRI Feature-Based Detection of FTD." https://github.com/PilipenkoTatiana/ML-Techniques-for-MRI-Feature-Based-Detection-of-FTD, 2022.

[79] A. Goel and S. K. Srivastava, "Role of kernel parameters in performance evaluation of svm," in *2016 Second international conference on computational intelligence & communication technology (CICT)*, pp. 166–169, IEEE, 2016.

[80] D. J. Kalita, V. P. Singh, and V. Kumar, "A survey on svm hyper-parameters optimization techniques," in *Social Networking and Computational Intelligence: Proceedings of SCI-2018*, pp. 243–256, Springer, 2020.

[81] R. Amami, D. B. Ayed, and N. Ellouze, "Practical selection of SVM supervised parameters with different feature representations for vowel recognition," *arXiv preprint arXiv:1507.06020*, 2015.

[82] V. K. Ayyadevara and V. K. Ayyadevara, "Random forest," *Pro Machine Learning Algorithms: A Hands-On Approach to Implementing Algorithms in Python and R*, pp. 105–116, 2018.

[83] M. Greenacre, P. J. Groenen, T. Hastie, A. I. d'Enza, A. Markos, and E. Tuzhilina, "Principal component analysis," *Nature Reviews Methods Primers*, vol. 2, no. 1, p. 100, 2022.

[84] W. Salehi, P. Baglat, G. Gupta, S. B. Khan, A. Almusharraf, A. Alqahtani, and A. Kumar, "An approach to binary classification of alzheimer's disease using lstm," *Bioengineering*, vol. 10, no. 8, p. 950, 2023.

[85] A. Mastinu, R. Ascrizzi, G. Ribaudo, S. A. Bonini, M. Premoli, F. Aria, G. Maccarinelli, A. Gianoncelli, G. Flamini, L. Pistelli, *et al.*, "Prosocial effects of nonpsychotropic cannabis sativa in mice," *Cannabis and Cannabinoid Research*, vol. 7, no. 2, pp. 170–178, 2022.

[86] J. L. Silverman, M. Yang, C. Lord, and J. N. Crawley, "Behavioural phenotyping assays for mouse models of autism," *Nature Reviews Neuroscience*, vol. 11, no. 7, pp. 490–502, 2010.

[87] M. Premoli, S. A. Bonini, A. Mastinu, G. Maccarinelli, F. Aria, G. Paiardi, and M. Memo, "Specific profile of ultrasonic communication in a mouse model of neurodevelopmental disorders," *Scientific Reports*, vol. 9, no. 1, p. 15912, 2019.

[88] W. Fyke, M. Premoli, V. Echeverry Alzate, J. A. López-Moreno, V. Lemaire-Mayo, W. E. Crusio, G. Marsicano, M. Woehr, and S. Pietropaolo, "Communication and social interaction in the cannabinoid-type 1 receptor null mouse: Implications for autism spectrum disorder," *Autism Research*, vol. 14, no. 9, pp. 1854–1872, 2021.

[89] D. Phillips, *Python 3 object-oriented programming: Build robust and maintainable software with object-oriented design patterns in Python 3.8*. Packt Publishing Ltd, 2018.

[90] P. Raybaut, "Spyder-documentation," *Available online at: pythonhosted. org*, 2009.

[91] C. A. Mertler, R. A. Vannatta, and K. N. LaVenia, *Advanced and multivariate statistical methods: Practical application and interpretation*. Routledge, 2021.

[92] T. Zhu and J.-T. Zhang, "Cosine similarity-based classifiers for functional data," *Contemporary Experimental Design, Multivariate Analysis and Data Mining: Festschrift in Honour of Professor Kai-Tai Fang*, pp. 277–292, 2020.

[93] F. Nielsen, "On a generalization of the jensen–shannon divergence and the jensen–shannon centroid," *Entropy*, vol. 22, no. 2, p. 221, 2020.

[94] S. Okada, M. Ohzeki, and S. Taguchi, "Efficient partition of integer optimization problems with one-hot encoding," *Scientific reports*, vol. 9, no. 1, p. 13036, 2019.

[95] S. Suman, K. S. Sahoo, C. Das, N. Jhanjhi, and A. Mitra, "Visualization of audio files using librosa," in *Proceedings of 2nd International Conference on Mathematical Modeling and Computational Science: ICMMCS 2021*, pp. 409–418, Springer, 2022.

[96] A. Clark *et al.*, "Pillow (pil fork) documentation," *readthedocs*, 2015.

[97] S. Tosi, *Matplotlib for Python developers*. Packt Publishing Ltd, 2009.

[98] N. Ketkar and N. Ketkar, "Introduction to keras," *Deep learning with python: a hands-on introduction*, pp. 97–111, 2017.

[99] K. Hara, D. Saito, and H. Shouno, "Analysis of function of rectified linear unit used in deep learning," in *2015 international joint conference on neural networks (IJCNN)*, pp. 1–8, IEEE, 2015.

[100] A. Gulli and S. Pal, *Deep learning with Keras*. Packt Publishing Ltd, 2017.

[101] A. Alblwi, *Improving the adaptive moment estimation optimization methods for modern machine learning*. University of Delaware, 2020.

[102] B. Juba and H. S. Le, "Precision-recall versus accuracy and the role of large data sets," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 4039–4048, 2019.

[103] J. Nunez-Iglesias, S. Van der Walt, and H. Dashnow, *Elegant SciPy: The Art of Scientific Python*. " O'Reilly Media, Inc.", 2017.

[104] N. Shukla and K. Fricklas, *Machine learning with TensorFlow*. Manning Greenwich, 2018.

[105] Y. Oyama, T. Ben-Nun, T. Hoefler, and S. Matsuoka, "{\mu}-cudnn: Accelerating deep learning frameworks with micro-batching," *arXiv preprint arXiv:1804.04806*, 2018.

[106] D. Arpit, Y. Zhou, B. Kota, and V. Govindaraju, "Normalization propagation: A parametric technique for removing internal covariate shift in deep networks," in *International Conference on Machine Learning*, pp. 1168–1176, PMLR, 2016.