Raffaella Fiamma Cabini

Computational techniques for biomedical image processing

Thesis submitted for the degree of Philosophiae Doctor

Joint PhD program in Computational Mathematics and Decision Sciences - XXXVI Cycle

Università di Pavia and Università della Svizzera Italiana

Supervisor: Silvia Figini

PhD Coordinator: Prof. Luca Pavarino

2023

Title:
Computational techniques for biomedical image processing

Author:
Raffaella Fiamma Cabini

PhD thesis - Università di Pavia and Università della Svizzera Italiana

Pavia, Italy, September 2023

Abstract

The aim of this PhD thesis is to explore the application of computational methods to address segmentation and image generation problems for different biomedical applications and imaging techniques. The developed algorithms aim to achieve various objectives, including accelerating traditionally manual or computationally slow operations, improving results accuracy, ensuring applicability across different imaging techniques and anatomical areas, and creating transparent models for easy understanding of their functionality. The thesis investigates three main research topics:

- *Statistical mechanics-based segmentation*: we propose a new method based on statistical mechanics for biomedical image segmentation. This approach conceptualizes each pixel as a particle with evolving positions and static gray levels, which interact with each other to form regions of segmentation. A key aspect of this model is the integration of a dynamic diffusion term, which quantifies stochastic variations arising during image acquisition. The Boltzmann formulation of the model is efficiently simulated using a Monte Carlo approach. An optimization strategy is proposed to fine-tune the system's internal parameters. The method is evaluated on different biomedical datasets, achieving segmentation performances in terms of Dice similarity coefficient of at least 0.91 for low-complexity segmentation tasks and at least 0.67 for high-complexity datasets. Future research should aim to enhance segmentation performance in more complex segmentation tasks.

- *COVID-19 lung lesion segmentation*: we present the LungQuant system, a fully-automatic deep learning (DL) pipeline designed for segmenting and quantifying COVID-19 lung lesions in computed tomography (CT) images. This system is composed of a cascade of two U-nets, a specialized convolutional neural network architecture designed for image segmentation tasks. The LungQuant system produces as output lung and COVID-19 lesion segmentation masks, the percentage of affected lung and the corresponding CT-Severity Score (CT-SS). We trained and tested all the DL models exclusively on publicly available datasets, achieving a 90% accuracy in CT-SS classification. We are currently developing various extensions of the study, which include technical improvements of the system, a multicenter validation and a radiomics study for clinical outcome prediction.

- *Optimized magnetic resonance fingerprinting (MRF)*: we propose an optimized MRF framework for generating quantitative multiparametric maps in preclinical studies. This method is composed by a DL model

and a hyperparameter tuning strategy that enables the simultaneous optimization of the neural network architecture, the structure of the DL model, and the supervised learning algorithm. The system reduces the mean percentage relative error of the computed maps by a factor of at least 2 and improves the computational time by at least a factor of 37, compared to the traditional reconstruction algorithm. Furthermore, our findings demonstrate that DL method allows the use of fewer MRF images and a reduced k-space sampling percentage, making MRF examinations more efficient. Future developments of this research may involve extending the proposed system to different anatomical regions or applying it to in vivo preclinical MRF.

Contents

Abstract **i**

Contents **iii**

List of Figures **vii**

List of Tables **ix**

1 Introduction **1**
1.1 Biomedical imaging 1
1.1.1 Computational techniques for biomedical image analysis 1
1.1.2 Applications 3
1.1.3 Challenges in biomedical image analysis 4
1.2 Thesis aim and scope 5
References 6

2 Background on Machine Learning and Deep Learning **9**
2.1 Classes of machine learning algorithms 9
2.1.1 Setting up a supervised learning problem 10
2.1.2 Gradient descent method 11
2.2 Deep Learning 12
2.2.1 Feedforward Neural Networks 13
2.2.2 Convolutional Neural Networks 15
2.2.3 Recurrent Neural Networks 16
2.3 Training a neural network 18
2.4 Explainable Deep Learning 19
References 21

A Segmentation of biomedical images **23**

Introduction **25**
References 32

I A kinetic approach to consensus-based segmentation of biomedical images **37**
I.1 Introduction 38
I.2 Modelling consensus dynamics 39
I.2.1 The bounded confidence model 40

I.2.2 Consensus models in segmentation problems . . . 41
I.3 Kinetic models for image segmentation . . . 42
I.3.1 Boltzmann-type derivation . . . 42
I.3.2 DSMC method for Boltzmann-type equations . . 46
I.3.3 Numerical examples for the Hegselmann-Krause dynamics . . . 47
I.4 Application to biomedical images . . . 49
I.4.1 Parameter identification . . . 51
I.4.2 2D biomedical image segmentation . . . 52
I.4.3 Patch-based 2D biomedical Image segmentation . 56
I.4.4 Comparison of different diffusion functions . . . 58
References . . . 61

II Quantification of pulmonary involvement in COVID-19 pneumonia by means of a cascade of two U-nets: training and assessment on multiple datasets using different annotation criteria 67
II.1 Introduction . . . 68
II.2 Material and Methods . . . 69
II.2.1 Datasets . . . 69
II.2.2 *LungQuant*: a DL based quantification analysis pipeline . . . 70
II.2.3 Training details and evaluation strategy for the U-nets . . . 73
II.3 Results . . . 74
II.3.1 U-net$_1$: Lung segmentation performance . . . 74
II.3.2 U-net$_2$: COVID-19 lesion segmentation performance 74
II.3.3 Evaluation of the quantification performance of the *LungQuant* system on a completely independent set 75
II.4 Discussion and Conclusion . . . 77
II.A
Additional descriptions of Materials and Methods . . . 79
II.A.1 Characteristics of the public datasets used in the study . . . 79
II.A.2 Additional training details and evaluation strategy for the U-nets . . . 81
II.A.3 Morphological refinement of U-net$_1$ lung segmentation . . . 83
II.A.4 Generation of a set of reference lung segmentation for model training . . . 83
References . . . 85

B Fast DL reconstruction techniques for preclinical MRF 87

Introduction 89

References 94

III Fast Deep Learning reconstruction techniques for preclinical Magnetic Resonance Fingerprinting 99

III.1 Introduction 100
III.1.1 Magnetic resonance imaging acquisition 102
III.1.2 Input data 103
III.1.3 Deep Learning reconstruction 103
III.1.4 Hyperparameter optimization 106
III.1.5 Evaluation strategy 106
III.1.6 Sequence compression methods 107
III.2 Results 108
III.2.1 Hyperparameter optimization results 108
III.2.2 Phantom results 108
III.2.3 Length of the MRF sequence results 111
III.2.4 K-space undersampling results 113
III.2.5 Computation time 114
III.2.6 Comparison with other DL methods 114
III.3 Discussion and conclusions 115
III.A Additional descriptions of DL methods compared 119
References 121

Conclusions 123

List of Papers 127

List of Figures

1.1 Image Processing block diagram 1

2.1 Block diagram of a supervised learning algorithm 11
2.2 Feedforward neural network 13
2.3 The basic unit of a NN 14
2.4 An example of 2D convolution 15
2.5 Illustration of a Recurrent Neural Network 17
2.6 Long Short-Term Memory block 18
2.7 Different applications of biomedical image segmentation 25

I.1 Results of the Hegselmann-Krause bounded confidence model . . 41
I.2 Transient solutions of the Fokker-Planck equation 48
I.3 Asymptotic solutions of the Fokker-Plank equation computed with the SP scheme 49
I.4 Comparison between the numeric solution of the Fokker-Planck equation and of the Boltzmann-type equation 50
I.5 Summary of the segmentation process 51
I.6 Results of the optimization process 54
I.7 Results of the segmentation process 54
I.8 Segmentation results for the “tumor core” task 55
I.9 Segmentation results for the “whole tumor” task 56
I.10 Results of the segmentation process for the *thigh muscles dataset* 57
I.11 Results of the segmentation process with the patch-based method 58
I.12 Diffusion functions 59
I.13 Segmentation results for the “tumor core” task with the two different diffusion functions 60

II.1 A summary of the whole analysis pipeline 71
II.2 U-net scheme 72
II.3 Three axial slices of a CT scan 76
II.4 Estimated percentages P of affected lung volume versus the ground truth percentages 77
II.5 Data augmentation to increase the diversity of dataset 83
II.6 Morphological refinement of the U-net$_1$ output 84
II.7 T_1 and T_2 relaxation times 90
II.8 T_1 and T_2 mapping. 91
II.9 MRF sequence acquisition 92
II.10 Overview of the MRF analysis framework 94

III.1 Flip angle (FA)/repetition time (TR) profiles used for the magnetic resonance fingerprinting acquisitions and dictionary generation. . 104
III.2 Schematic representation of the two architectures 105
III.3 Schematic representation of k-fold cross-validation 107
III.4 Results of the optimization process for the Gao and the Zhao datasets 109
III.5 Comparison of the results 111
III.6 Boxplots of the percentage relative errors 112
III.7 Boxplots of the percentage relative errors for two regions of interest 112
III.8 Boxplots of the percentage relative errors as a function of different sequence lengths 113
III.9 Boxplots of the percentage relative errors as a function of the k-space sampling percentage 115

List of Tables

I.1 Best configurations of the parameters given by the optimization process . . . 53
I.2 Estimated values of the parameters given by the optimization process for two different diffusion functions . . . 59
I.3 Segmentation accuracy before and after the morphological refinement steps . . . 59

II.1 A summary of the datasets used in this study . . . 69
II.2 Number of CT scans assigned to the train, validation (val) and test sets . . . 73
II.3 Performances achieved by U-net$_1$ in lung segmentation on different test sets . . . 74
II.4 Performances achieved by U-net$_2$ in COVID-19 lesion segmentation 75
II.5 Performances of the *LungQuant* system on the independent COVID-19-CT-Seg test dataset . . . 76
II.6 Classification performances of the whole system . . . 78
II.7 MosMed severity categories defined on the basis of the percentage P of lung volume affected by COVID-19 lesions. The correspondence to the CT-SS scale is reported. . . . 80

III.1 Best configuration of the parameters obtained using the optimization process . . . 110
III.2 Comparison with some of the state-of-the-art models . . . 116

Chapter 1

Introduction

1.1 Biomedical imaging

Biomedical imaging is widely recognized as one of the most important diagnostic tools in modern medicine. It encompasses various methods and techniques that enable the non-invasive acquisition of precise representations of the internal structures of the body, providing valuable information about anatomical components, physiological processes and pathological conditions. Biomedical images are obtained from a wide range of acquisition modalities such as X-ray, Computed Tomography (CT), Magnetic Resonance Imaging (MRI), nuclear imaging (including positron emission tomography (PET)) and ultrasound. Each imaging modality provides different information about the region under examination and is chosen based on the specific clinical question and patient requirements [1].

Biomedical image interpretation is the process of extracting valuable information from medical images. By carefully analyzing and interpreting medical images, healthcare professionals can identify and characterize anatomical structures, detect abnormalities, and evaluate the progression or regression of diseases. This information is essential for making accurate diagnoses, determining optimal treatment strategies, and monitoring the effectiveness of interventions over time. In clinical practice, the interpretation of medical images has mostly been carried out by healthcare professionals, such as radiologists and physicians. However, due to the wide range of pathologies and imaging modalities, manual or visual interpretation of images requires significant and tedious effort on the part of medical experts. Moreover, the subjective nature of the work makes it prone to human error and may introduce considerable variations in interpretations among different experts. As a result, in recent years there is growing interest in the potential benefits of integrating computational methods to support medical image interpretation [2].

1.1.1 Computational techniques for biomedical image analysis

Image processing can be conceptualized as an input-output system [3]. Figure 1.1 illustrates the block diagram representation.

Figure 1.1: Image Processing block diagram. Source [3].

Where, τ represents an image processor (such as denoising, deblurring,

segmentation, compression, or inpainting). The input data is denoted as Q_0 and represents a measured single image or a sequence of images. The output $Q = (q_1, q_2, \dots)$ contains all the desired image features. The aim of mathematical research in image processing is to model the processing operator τ, to automatically associate the input data Q_0 with the required output features Q. By constructing accurate and efficient models for these processing operators, researchers can facilitate the extraction of meaningful information from images, thereby simplifying tasks that are typically performed manually or visually.

In the field of biomedical image analysis, various processing operators can be applied to images, including registration, segmentation, classification, denoising, pattern recognition, and more. For all these operators, the extraction of valuable information from medical images can be approached using two primary computational methodologies: *mechanistic modelling* and *data-driven modelling* or *machine learning* (ML) [4].

A *mechanistic model* is a mathematical function that predicts outcomes based on a theoretical understanding of the system. It relies on the creation of new hypotheses about causal mechanisms, which are generated through observations of the phenomenon under investigation. Mechanistic modelling follows a two-stage process: initially, a subset of available data is utilized to construct and calibrate the model and subsequently additional data is employed to validate and refine the model to improve its accuracy. The aim is to develop models that can be utilized in situations where conducting experiments is either impossible or challenging, allowing for practical applications based on the acquired understanding of the underlying mechanisms. One main advantage of mechanistic models is their intrinsic interpretability, as the results given by the model can be explained based on the underlying mathematical principles. However, it can be challenging to construct accurate models for complex or ambiguous image patterns.

On the other hand, *ML* has gained significant popularity in image analysis, particularly with the emergence of *Deep Learning* (DL) algorithms. ML can be defined as a computer-based process that improves its performance on a given task through experience [5]. This experience can be acquired through two main sources: interactions with labeled data or interactions with the environment while actively performing the task. ML is a programming approach that is completely different from mechanistic modeling because, instead of explicitly designing a mathematical model, it uses algorithms that learn from experiences to improve their performance in executing a specific task. In the field of ML, DL leverages neural networks (NNs) to approximate intricate, nonlinear relationships between inputs and outputs. ML and DL techniques are well-suited for handling complex and unstructured data such as biomedical images, allowing them to learn intricate patterns and generalize to unseen inputs. However, the lack of interpretability in some ML and DL models poses challenges in understanding the rationale behind specific decisions.

1.1.2 Applications

The applications of computational techniques for biomedical image analysis are numerous and encompass a wide range of tasks. This thesis focuses on two fundamental topics in this research area: *biomedical image segmentation* and *biomedical image generation*.

Image segmentation refers to the process of partitioning an image into distinct objects or regions of interest (ROIs). In the biomedical imaging field these ROIs may correspond to organs, tissues, pathologies or other biologically relevant structures. Image segmentation plays a crucial role in the clinical workflow for several reasons. First, by segmenting images, clinicians can measure the size, shape, and spatial distribution of anatomical structures, tissues, or lesions, aiding in quantitative analysis and assessment. This information provides valuable insights for diagnostic purposes, treatment planning, and monitoring disease progression. For example, by segmenting an organ and measuring the volume of its damaged portion, it is possible to estimate the percentage of organ impairment [6], [7]. This information can be valuable in assessing the extent of organ damage and determining appropriate treatment or intervention strategies. Moreover, the segmentation process is the first step for the extraction of region-specific features, enabling further analysis, such as texture characterization or functional mapping. One of these techniques is *radiomics*, a biomedical imaging analysis which is used to derive a wide range of quantitative features from a segmented ROI [8]. These features include intensity-based attributes, shape descriptors, and texture characteristics, among others. The extraction of such features has the potential to enable a comprehensive characterization of the ROI, providing valuable information for diagnostic, prognostic, and treatment-related purposes. Traditionally, healthcare professionals or radiologists perform segmentation by visually analyzing the images and manually delineating the ROIs. However, this manual approach is time-consuming, subjective, and can introduce inter-observer variability. To address these challenges, various algorithms have been proposed. This thesis will address two specific segmentation approaches, DL and clustering, which will be extensively described in sections I and II.

Biomedical image generation represents one of the most challenging applications among computational techniques. It involves the process of computationally synthesizing new visual representations that mimic real-world biomedical data. The complexity of this task arises from the elaborate and highly dimensional nature of biomedical images, as well as the demand for accuracy and reliability for their effective use in medical diagnosis and treatment. However, despite these challenges, biomedical image synthesis has numerous applications, such as superresolution [9], denoising and contrast enhancement [10], transforming images between different modalities [11], accelerating image acquisition and reconstruction [12], motion correction [13], among many others. To achieve these goals, a wide range of computational techniques is utilized. These techniques are almost exclusively based on ML methods, particularly on DL. In recent years, there has been extensive testing of various DL models for image generation, characterized by different architectures with varying

levels of complexity. These models span from straightforward pixel-to-pixel transformation methods to more sophisticated Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs) [14]. Specifically, GANs are a special type of DL algorithm used for image generation and data synthesis. GANs consist of two main parts: the generator and the discriminator. The generator creates images that should resemble real ones, while the discriminator's job is to distinguish between real and generated images. They work together, with the generator trying to improve its output to fool the discriminator, and the discriminator becoming better at choosing real from fake. This competition leads to the generation of high-quality and realistic data. However, training GANs effectively requires a vast collection of real-world biomedical images. Acquiring a sufficiently large dataset for GAN training is challenging in the biomedical domain, posing a significant obstacle to the advancement of GAN-based biomedical image generation. Selecting the appropriate complexity for the model is crucial and should be based on the specific difficulty of the problem and the available data resources.

1.1.3 Challenges in biomedical image analysis

As we have seen above, computational techniques have a wide range of applications and utilities in the field of biomedical imaging. However, their applicability is hindered by several challenges, which vary according to the specific research question and the computational approach used.

The first challenge in biomedical image analysis is the requirement of sufficiently large annotated datasets to build and test mathematical models. The scarcity of labeled medical images obstacles the development and generalization of algorithms, particularly in rare diseases or specialized imaging domains. Data privacy and ethical considerations also must be carefully addressed to ensure confidentiality and patient consent when using medical images for analysis.

Another significant challenge in the field of biomedical image analysis is data standardization. The existence of different data sources, hardware systems and imaging centers contributes to high variations in acquisition protocols, image formats, and metadata. This variability complicates the development of algorithms and computational tools that can effectively handle and analyze these heterogeneous datasets. Harmonizing data standards becomes crucial to ensure the correct integration of biomedical images from different sources. However, it is important to underline that issues related to image acquisition, such as noise, artifacts and patient movement, as well as variations in annotation styles, generate a variability that cannot be completely overcome through data standardization alone. Robust algorithms are required to handle variations in image quality and effectively address these issues in biomedical image analysis.

Moreover, the interpretability and explainability of computational models are essential in biomedical image analysis. DL models, although highly effective, are often regarded as black boxes due to their complex architectures. Understanding how and why these models make inferences is crucial for building trust and confidence in their use in clinical practice.

Overall, addressing these challenges is crucial for advancing biomedical image analysis and leveraging its potential for improved diagnostics, treatment planning, and research in healthcare.

1.2 Thesis aim and scope

This thesis proposes new computational methods to overcome the limitations of image segmentation and generation for different biomedical applications and imaging techniques. To accomplish these tasks, both mechanistic models and data-driven approaches will be introduced and compared. The biomedical challenges tackled in this thesis can be summarized as follows: i) improve operational speed, ii) refine accuracy and alignment with desired outcomes, iii) achieve scalability and applicability across different anatomical regions and imaging modalities and iv) provide explainable results.

More in detail, the first part A focuses on image segmentation problems using two different computational approaches. The first approach I utilizes a clustering method based on statistical mechanics, which provides a non-trainable model for general biomedical imaging problems. The key idea of the method is to represent pixels as particles that interact with each other following a revised consensus dynamic model. To address stochastic variations in the image acquisition process, we incorporate a diffusion term that enhances the model's flexibility. The unsupervised nature of clustering provides several advantages over supervised ML and DL models for image segmentation, particularly when addressing segmentation challenges that are not excessively complex. It doesn't rely on labeled training data, making it suitable for small datasets and adaptable to various imaging techniques. In addition to these aspects, our proposed method is based on a statistical mechanics model, providing inherent interpretability to the segmentation process.

The second approach II involves DL techniques for segmenting COVID-19 lung lesions from CT images, with the aim of quantifying the percentage of lung impairment. Given the complexity of discerning COVID-19 lesions from healthy lung tissue in CT images, attributed to the diverse texture, shape, and distribution patterns exhibited among various patients, mechanistic segmentation techniques are not appropriate for fulfilling this objective. On the other hand, DL-based approaches such as CNNs are valuable tools in tackling this complex segmentation task effectively. We propose the LungQuant system, a DL-based approach for automated segmentation and quantification of COVID-19 lung lesions from CT scans. The system is composed of a cascade of two CNNs developed specifically for image segmentation, known as U-nets. The first network segments lung tissue, while the second identifies COVID-19 lesions within the segmented lungs. The system calculates the compromised lung volume as a percentage of the total lung volume and converts it into the CT-Severity Score (CT-SS).

The second part B is focused on the task of image generation to optimize the process of acquiring and reconstructing Quantitative MRI (QMRI) images using

magnetic resonance fingerprinting (MRF) methodology. MRF is a QMRI [15] technique able to acquire multiple tissue properties in one-shot measurement. However, while the MRF data acquisition protocol is fast, the traditional post-processing procedure to extract tissue properties is relatively slow and requires significant storage capacity. We developed an optimized MRF DL-based framework to provide quantitative multiparametric maps for preclinical studies. The primary objective of the DL model is to generate accurate quantitative maps from raw MRF acquisitions. Through this approach, we aim to improve both the accuracy of the parameter reconstruction and the computational efficiency of the computation process.

In conclusion, this thesis aims to propose novel computational methods for addressing challenges in image segmentation and image generation. Through such contributions, the thesis aims to advance the accuracy, efficiency and usability of computational techniques in these imaging domains. However it's important to underline that the included papers do not aim to provide complete solutions to the image segmentation or generation problems but rather focus on enhancing specific aspects within the larger framework. These methods are developed for research contexts, lacking practical software solutions applicable to everyday scenarios, such as clinical care. Consequently, for the proposed methods to become viable in practice, a thorough evaluation on larger datasets is required.

References

[1] Morris, P., *Biomedical imaging: applications and advances*. Elsevier, 2014.

[2] Shen, D., Wu, G., and Suk, H.-I., "Deep learning in medical image analysis," *Annual Review of Biomedical Engineering*, vol. 19, pp. 221–248, 2017.

[3] Chan, T. F., Shen, J., and Vese, L., "Variational pde models in image processing," *Notices AMS*, vol. 50, no. 1, pp. 14–26, 2003.

[4] Baker, R. E., Pena, J.-M., Jayamohan, J., and Jérusalem, A., "Mechanistic models versus machine learning, a fight worth fighting for the biological community?" *Biology Letters*, vol. 14, no. 5, p. 20 170 660, 2018.

[5] Goodfellow, I., Bengio, Y., and Courville, A., *Deep learning*. MIT press, 2016.

[6] Lizzi, F., Agosti, A., Brero, F., *et al.*, "Quantification of pulmonary involvement in COVID-19 pneumonia by means of a cascade of two u-nets: Training and assessment on multiple datasets using different annotation criteria," *International journal of Computer Assisted Radiology and Surgery*, pp. 1–9, 2022.

[7] Lizzi, F., Postuma, I., Brero, F., *et al.*, "Quantification of pulmonary involvement in COVID-19 pneumonia: An upgrade of the LungQuant software for lung CT segmentation," *The European Physical Journal Plus*, vol. 138, no. 4, pp. 1–10, 2023.

[8] Van Timmeren, J. E., Cester, D., Tanadini-Lang, S., Alkadhi, H., and Baessler, B., "Radiomics in medical imaging—"how-to" guide and critical reflection," *Insights into Imaging*, vol. 11, no. 1, pp. 1–16, 2020.

[9] Ledig, C., Theis, L., Huszár, F., *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2017, pp. 4681–4690.

[10] Yi, X. and Babyn, P., "Sharpness-aware low-dose CT denoising using conditional generative adversarial network," *Journal of Digital Imaging*, vol. 31, pp. 655–669, 2018.

[11] Wolterink, J. M., Dinkla, A. M., Savenije, M. H., Seevinck, P. R., Berg, C. A. van den, and Išgum, I., "Deep MR to CT synthesis using unpaired data," in *Simulation and Synthesis in Medical Imaging*, Springer, 2017, pp. 14–23.

[12] Quan, T. M., Nguyen-Duc, T., and Jeong, W.-K., "Compressed sensing MRI reconstruction using a generative adversarial network with a cyclic loss," *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, pp. 1488–1497, 2018.

[13] Oksuz, I., Clough, J., Bustin, A., *et al.*, "Cardiac MR motion artefact correction from k-space using deep learning-based reconstruction," in *Machine Learning for Medical Image Reconstruction*, Springer, 2018, pp. 21–29.

[14] Goodfellow, I., Pouget-Abadie, J., Mirza, M., *et al.*, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[15] Keenan, K. E., Biller, J. R., Delfino, J. G., *et al.*, "Recommendations towards standards for quantitative MRI (qMRI) and outstanding needs," *Journal of Magnetic Resonance Imaging: JMRI*, vol. 49, no. 7, e26, 2019.

Chapter 2

Background on Machine Learning and Deep Learning

Machine learning (ML) is a subfield of Artificial Intelligence with the goal of developing algorithms capable of automatically learning from data and making predictions or decisions without being explicitly programmed. The fundamental concept of ML is to build algorithms that are capable of improving a computer's program performance at some task through the experience learned from the data. Many different tasks can be solved using ML techniques. Three of the most common tasks are:

- *Classification*: the goal of this task is to assign inputs to specific categories or classes. ML algorithms are trained to generate a function that maps continuous and/or categorical variables to categorical values.
- *Regression*: the purpose of this task is to predict numerical values based on input data. ML algorithms are used to build models that can map continuous and/or categorical variables to continuous outputs.
- *Clustering*: is a task where the objective is to group similar inputs together based on patterns and similarities within the data. Unlike in classification, the specific groups or categories are not predefined.

In addition to these tasks, there are many other possible applications of ML. A complete overview of ML algorithms and applications can be found in the references [1]–[3].

2.1 Classes of machine learning algorithms

ML algorithms can be categorized based on the type of experience they learn during the training process. They can be broadly classified into three categories:

- *Supervised learning:* is a type of ML where the algorithm learns from labeled data. The training dataset consists of examples where each data point is associated with its correct label. Common tasks in supervised learning include classification and regression, where the algorithm learns to predict labels or values for new and unseen data based on the patterns and relationships learned from the labeled training data.
- *Unsupervised learning:* is a class of ML where the algorithm focuses on discovering patterns and structures in unlabeled data. Unlike supervised learning, there are no predefined labels or target outputs. Instead,

unsupervised learning algorithms aim to identify inherent patterns, clusters, or relationships within the data itself.

- *Reinforcement learning:* is a category of ML where the algorithm learns by interacting with an environment and receiving feedback in the form of rewards or penalties. The objective of the learning system is to find an optimal strategy that maximizes rewards over time. The algorithm learns through trial and error, adjusting its behavior based on the feedback received from the environment.

In addition to these three main categories of ML algorithms, there exist numerous alternative approaches that combine elements from different learning paradigms. One such notable approach is active learning [4], [5], which represents a subtle modification of traditional supervised learning. Active learning is an interactive ML approach where the model or an external supervisor actively selects examples for effective learning. Unlike supervised traditional methods, active learning involves model initialization with initial training on a small dataset, selection of informative examples for further learning, and iterative model retraining until satisfactory performance is achieved. The goal is to minimize manual data annotation by focusing on specific examples that provide the most information for the model. This could be particularly useful in situations where manual data annotation is costly or time-consuming. Active learning has gained particular attention in recent years, particularly in biomedical imaging, where the acquisition and annotation of high-quality labeled data can be extremely expensive and time-consuming.

In the next paragraph, we will provide a more detailed description of the class of supervised learning algorithms, as they will be extensively utilized in the subsequent chapters of this thesis.

2.1.1 Setting up a supervised learning problem

Many problems in supervised ML start with the same fundamental elements. The first element is the dataset $D = (\boldsymbol{x}, \boldsymbol{y})$ where $\boldsymbol{x}$ is a vector of independent variables and $\boldsymbol{y}$ is a vector of dependent variables. The dataset is a collection of examples composed by inputs $\boldsymbol{x}$ and outputs $\boldsymbol{y}$. The second element is the model $f(\boldsymbol{x}; \boldsymbol{\theta})$, which is a function $f : \boldsymbol{x} \rightarrow \boldsymbol{y}$ parameterized by $\boldsymbol{\theta}$. f is used to predict output values based on input variables. The final element is the loss function $L(\boldsymbol{y}, f(\boldsymbol{x}; \boldsymbol{\theta}))$, which allows us to assess the performance of the model by measuring the discrepancy between the predicted output $f(\boldsymbol{x}; \boldsymbol{\theta})$ and the true output $\boldsymbol{y}$. The model is fit by finding the optimal values of $\boldsymbol{\theta}$ that minimize the loss function. The goal of the procedure is therefore to find the function f that approximate the unknown relationship f^* between the inputs and the outputs. A schematic representation of these elements is provided by Figure 2.1.

What distinguishes ML from simple optimization is the emphasis on generalization. The trained model should perform well not only on the training data but also on new and previously unseen inputs. The ability to generalize is

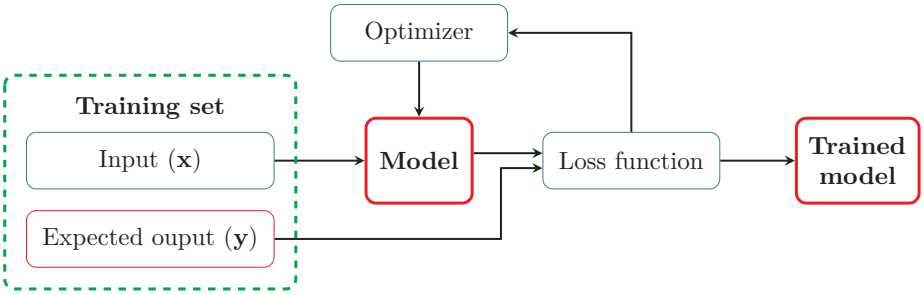


Figure 2.1: Block diagram of a supervised learning algorithm.

crucial. The main challenge in ML is to achieve low generalization error, which is the value of the loss function on new inputs.

To estimate the generalization error of a ML model, a common approach is to randomly divide the dataset D into two mutually exclusive groups: the training set D_{train} and the test set D_{test}. The model is trained by minimizing the loss function using only the data in the training set. Then, the predictive performance of the model is evaluated by computing the loss function using the test set. By splitting the data into separate training and test sets, we obtain an unbiased estimate of the model's predictive performance. This practice, known as cross-validation, helps assess how well the model generalizes to new and unseen data.

2.1.2 Gradient descent method

We have previously seen that in supervised ML algorithms, optimization of the loss function involves adjusting the model's parameters. One of the most widely used methods for this purpose is the gradient descent (GD).

The fundamental idea behind gradient descent is to reduce the loss function $L(\boldsymbol{y}, f(\boldsymbol{x}; \boldsymbol{\theta}))$ by updating the parameters $\boldsymbol{\theta}$ in small steps with the opposite direction of the gradient. This iterative process ensures that the parameters move towards a local minimum of the loss function. In the simplest form of GD algorithm, we update the parameters as follows: we initialize the parameters to some value $\boldsymbol{\theta_0}$ and then iteratively update them using the equation:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \epsilon \nabla_\theta L(\boldsymbol{\theta}) \tag{2.1}$$

where $\nabla_\theta L(\boldsymbol{\theta})$ represents the gradient of the loss function $L(\boldsymbol{\theta}) = L(\boldsymbol{y}, f(\boldsymbol{x}; \boldsymbol{\theta}))$, and ϵ is the learning rate, i.e. a positive scalar determining the step size.

However, the simple gradient descent algorithm can be slow and computationally expensive, particularly for large datasets. Additionally, in many ML algorithms, the loss function is a sum of terms, with each term corresponding to a data point:

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{n} l_i(\boldsymbol{x}_i, \boldsymbol{\theta}) \tag{2.2}$$

Computing the gradient by summing over all n data points at each GD step can be computationally inefficient. To address this issue, the *Stochastic Gradient Descent* (SGD) method was introduced. Instead of computing the actual gradient over the entire dataset, SGD approximates the gradient using a small partition of the data known as a "mini-batch":

$$\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) = \sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} l_i(\boldsymbol{x}_i, \boldsymbol{\theta}) \rightarrow \sum_{i \in B_K}^{n} \nabla_{\boldsymbol{\theta}} l_i(\boldsymbol{x}_i, \boldsymbol{\theta}) \tag{2.3}$$

where B_k denotes the mini-batches. The algorithm cycles over all $k = 1, \ldots, n/M$ (with M as the mini-batch size) mini-batches one at a time and uses the mini-batch approximation to update the parameters $\boldsymbol{\theta}$ at each step k. Completing a full iteration over all n data points, using all n/M mini-batches, is referred to as an *epoch*. SGD significantly speeds up the calculation as it avoids the need to use all n data points to approximate the gradient.

One limitation of GD and SGD algorithms is their sensitivity to the choice of learning rate. A very small learning rate leads to extremely slow training, while larger learning rates may cause GD to diverge and yield poor results. Moreover, the learning rate is constrained by the steepest direction, which can vary depending on the current position in the loss function landscape.

To overcome this issue, there has been recent progress in introducing methods that adaptively adjust the step size based not only on the gradient but also on the second moment of the gradient i.e. the uncentered variance (the mean is not subtracted during variance computation). Examples of such methods include AdaGrad, AdaDelta, RMSprop, and Adam [1]. These techniques allow for more efficient and effective optimization of the loss function.

2.2 Deep Learning

Deep learning (DL) is a subfield of ML that leverages neural networks (NNs) to tackle complex problems. NNs are nonlinear models for supervised and unsupervised learning, drawing inspiration from the biological neural networks found in animal brains.

Conceptually, NNs can be categorized into three main types:

1. *Feedforward Neural Networks:* these models are versatile and widely used for supervised learning tasks. They consist of an input layer (i.e. an array of learnable parameters), one or more hidden layers and an output layer. Information flows in a forward direction, from the input layer through the hidden layers to the output layer.

2. *Convolutional Neural Networks:* CNNs are specifically designed for image processing tasks. They use convolutional layers, consisting of arrays of learnable parameters that filter input images, to extract local patterns and features from input images, enabling effective analysis and understanding of visual data.

3. *Recurrent Neural Networks:* RNNs are specialized NNs that excel in handling sequential data. They have connections between neurons at different time steps, allowing them to remember past inputs and process sequential information effectively.

Each category of NNs has its own unique characteristics and is suited for different types of data and problem domains. In the following sections, we will explore these three categories of NNs in more detail.

2.2.1 Feedforward Neural Networks

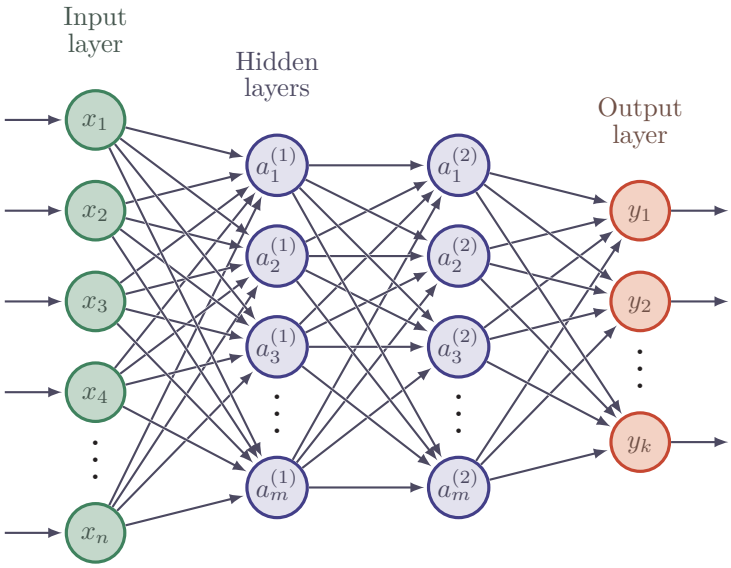


Figure 2.2: Basic architecture of a feedforward NN with two hidden layers. Neurons are arranged into layers with the output of one serving as the input to the next layer. The signal travels from the first layer (the input layer), to the last layer (the output layer), after passing through intermediate layers (the hidden layers).

Feedforward NNs, also known as *multilayer perceptrons* (MLPs), are mathematical functions that map an input vector $\boldsymbol{x}$ to an output vector $\boldsymbol{y}$. The objective of a NN is to approximate a non-linear function $\boldsymbol{y} = f^*(\boldsymbol{x})$. A MLP can be represented as a composition of many different functions:

$$f(\boldsymbol{x}) = f^{(d)} \circ f^{(d-1)} \circ \cdots \circ f^{(2)} \circ f^{(1)} \tag{2.4}$$

The first function $f^{(1)}$ is referred to as the *input layer*, while the final function $f^{(d)}$ is the *output layer*. The functions in between are known as *hidden layers*. The learning algorithm determines how to utilize these layers to best approximate f^*. Since the desired output for these intermediate layers is not provided in the

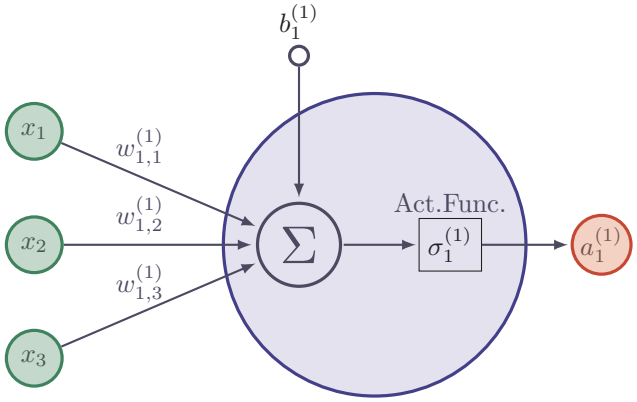


Figure 2.3: The basic unit of a NN, the neuron or node. It consists of a linear transformation that weights the importance of various inputs, followed by a non-linear activation function.

training data (unlike the output layer), they are referred to as "hidden layers". The number of layers determines the *depth* of the network. Modern NNs typically consist of multiple hidden layers, hence the term "deep" in deep DL. Figure 2.2 illustrates an MLP with two hidden layers.

Each function within the network is typically vector-valued. However, instead of considering each layer as a single vector-to-vector function, we can also view the layer as comprising many units that operate in parallel, with each unit representing a vector-to-scalar function. The fundamental unit i of the layer j of the NN, also known as a "neuron" or "node", takes a vector of n input features $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ and produces a scalar output $a_i^{(j)}(\boldsymbol{x})$, referred to as the *activation*.

The function $a_i^{(j)}$ can be decomposed into a linear operation that weighs the relative importance of the inputs using neuron-specific weights $\boldsymbol{w}_i^{(j)} = (w_{i,1}^{(j)}, w_{i,2}^{(j)}, \ldots, w_{i,n}^{(j)})$, followed by re-centering with a neuron-specific bias $b_i^{(j)}$:

$$z_i^{(j)} = {\boldsymbol{w}_i^{(j)}}^T \cdot \boldsymbol{x} + b_i^{(j)} \tag{2.5}$$

The complete input-output function can be expressed as:

$$a_i^{(j)}(\boldsymbol{x}) = \sigma_i^{(j)}(z_i^{(j)}) \tag{2.6}$$

where $\sigma_i(z)^{(j)}$ is the non-linear function, known as the *activation function*, which is typically the same for all neurons within a layer. Figure 2.3 represents a schematic of a node within an MLP, illustrating its components.

Historically, common non-linearities used in neurons include step functions, sigmoids, and hyperbolic tangents. However, more recently, rectified linear units (ReLUs), leaky rectified linear units (leaky ReLUs), and exponential linear units (ELUs) have gained popularity [1].

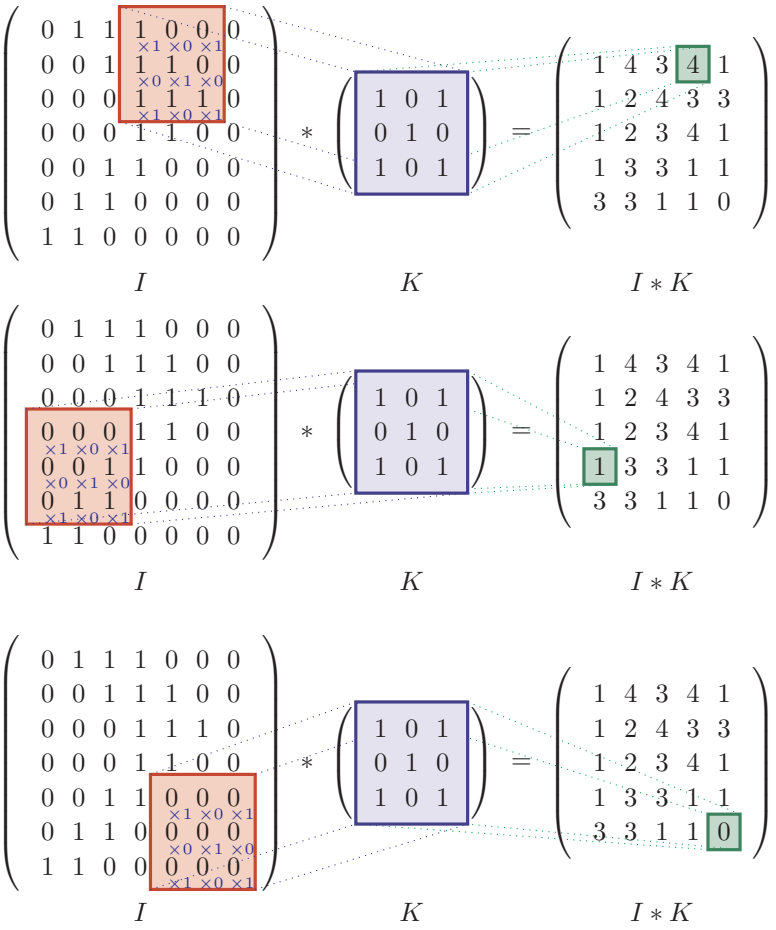


Figure 2.4: An example of 2D convolution with stride equal to 1. In this illustration, we have an input array I with dimensions $7 \times 7 \times 1$ and a 3×3 kernel denoted as K. The result of the convolution operation is referred to as the feature map, which is a 5×5 array.

2.2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) [6] are a class of NNs designed to handle data with a grid-like structure, like images. A CNN typically is composed of two classes of layers: convolutional layers and a pooling layers.

The convolution layer is the core building block of the CNN. Its primary task is taking a multidimensional array of data as input and producing a transformed multidimensional array, known as features map, as output. The convolution transformation consists of a dot product between two arrays: the first array is a set of learnable parameters known as a kernel, while the second array is a

restricted portion of the input data. The kernel is spatially smaller than the input but it has more channels. This means that, if we consider a RGB image as input (three channels), the kernel's height and width will be small, but the depth extends up to all three channels. During the convolution process, the kernel slides across the height and width of the input array. At each spatial position, the corresponding elements of the two grids are multiplied and then summed up to calculate a scalar value. This computation leads to the generation of a multidimensional features map, which results by the convolution operation performed at each spatial position within the input array. The sliding size of the kernel is called stride and determines the dimension of the features map. The choice of kernel size is a key aspect in CNN design. A smaller kernel size captures finer details of the input array, whereas a larger kernel size focuses on larger, more abstract features. A visual representation of convolution applied to a 2D tensor is provided by Figure 2.4.

Since convolution is a linear operation and images contain non-linear patterns, introducing non-linearity functions allows the CNN to capture and process complex features in the feature map. These non-linearity functions are the same activation functions used in feedforward NNs.

The second class of layers of a CNN are the pooling layers. The pooling layer substitutes the output of the CNN at a certain location with a summary statistic computed from nearby outputs. Pooling reduces the spatial dimension of the feature maps and it is applied individually to each slice of the feature maps. Various pooling functions exist, including average pooling, which calculates the average output within the neighborhood, and max pooling, which reports the maximum output within the neighborhood.

2.2.3 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) [7] are a family of NNs designed for processing sequential data. The key idea behind RNNs is inspired by the way humans think. When we read a sentence, our understanding of each word relies on what we've comprehended from the preceding words. This persistence of information is lacking in traditional NNs like MLPs, making them less suitable for processing sequential data. RNNs address this issue. They are networks that share the same weights across several time steps, allowing information to persist.

Suppose to operate on a sequence of vectors $\boldsymbol{x}^{(t)} = (\boldsymbol{x}^{(\tau)}, \boldsymbol{x}^{(\tau-1)}, \ldots, \boldsymbol{x}^{(2)}, \boldsymbol{x}^{(1)})$. In the left graph of Figure 2.5, a single block of RNN, denoted as $\boldsymbol{h}$, takes an input $\boldsymbol{x}$ and generates an output value $\boldsymbol{o}$. The loop in the graph symbolizes how information can be propagated from one time step of the network to the next time step, capturing temporal dependencies within the data. If we unfold the loop, the RNN can be thought of as a chain of multiple copies of the same network, each passing a message to a successor, as depicted in the right graph of Figure 2.5. Specifically, the forward propagation of the RNN shown in Figure 2.5 begins by specifying the initial state $\boldsymbol{h}^{(0)}$. Subsequently, for each time step from

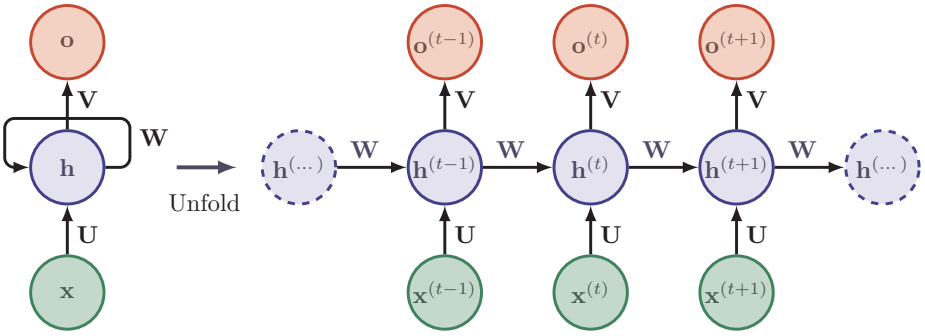


Figure 2.5: Illustration of a RNN. The left graph shows the compact representation of the RNN, while on the right, the same structure is displayed as a time-unfolded computational graph, where each node corresponds to a specific time instance in the sequence. The RNN consists of three sets of connections: input-to-hidden connections parametrized by weight matrix $\mathbf{U}$, hidden-to-hidden recurrent connections parametrized by weight matrix $\mathbf{W}$, and hidden-to-output connections parametrized by weight matrix $\mathbf{V}$.

$t = 1$ to $t = \tau$, we utilize the following update equations:

$$\begin{aligned} \mathbf{a}^{(t)} &= \mathbf{b} + \mathbf{W}\mathbf{h}^{(t-1)} + \mathbf{U}\mathbf{x}^{(t)} \\ \mathbf{h}^{(t)} &= \sigma(\mathbf{a}^{(t)}) \\ \mathbf{o}^{(t)} &= \mathbf{c} + \mathbf{V}\mathbf{h}^{(t)} \end{aligned} \tag{2.7}$$

where $\mathbf{b}$ and $\mathbf{c}$ are the bias vectors while $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$ are the weight matrices for input-to-hidden, hidden-to-output and hidden-to-hidden connections respectively. σ is the activation function for the hidden units.

Despite the theoretical capability of RNNs to handle long-term dependencies, they often have difficulties in effectively learning such dependencies due to the vanishing or exploding gradient problem [8], [9]. When gradients flow backward during the training, they can decrease exponentially over time, causing issues in capturing long-term interactions.

To address this challenge, Long Short-Term Memory networks (LSTMs) were introduced [8], [10]. LSTMs have a chain-like structure similar to RNNs but with a different repeating module. The core element of LSTMs is the cell state, represented by the horizontal line at the top of diagram 2.6. The cell state functions as a conveyor belt, allowing information to flow along it with some minor linear interactions. Moreover, LSTMs have three specialized structures called gates, which selectively control the passage of information by removing or adding information to the cell state. The gates consist of a sigmoid neural net layer followed by a pointwise multiplication operation, which outputs values between zero and one to control the flow. The three gates in an LSTM are responsible for effectively safeguarding and controlling the cell state.

This design allows LSTMs to overcome the limitations faced by traditional

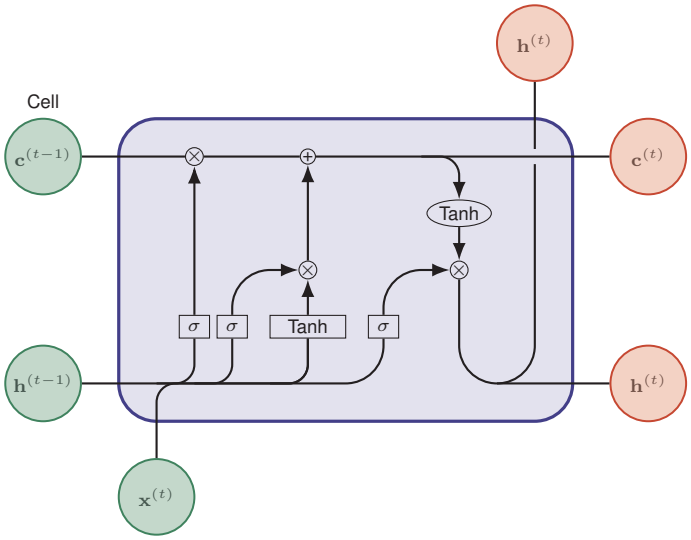


Figure 2.6: The block diagram illustrates the structure of an LSTM recurrent network cell. LSTM cells utilize gating units to control input, state, and output, enabling efficient handling and storage of information over long sequences.

RNNs making them more effective at handling sequential data with long-term dependencies.

2.3 Training a neural network

The process of training a NN is a supervised learning problem. The fundamental steps of NN training are similar to what we described for supervised ML algorithms: defining a loss function and then using the gradient descent method to minimize the loss function and find the optimal weights and biases for the NN. However, the main distinction between linear ML models and NNs lies in the presence of multiple non-linear hidden layers in NNs, making the computation of the gradient more complex.

As with any supervised learning procedure, the first step in training a NN is to define a suitable loss function. Given a data point from the training set $(\boldsymbol{x}_i, y_i)$, where $\boldsymbol{x}_i \in \mathbb{R}^{d+1}$, the NN makes a prediction $\hat{y}_i(\boldsymbol{w})$, where $\boldsymbol{w}$ represents the parameters of the NN. The choice of loss function depends on whether you aim to make continuous or categorical predictions.

For continuous data, common loss functions used to train NNs include the *mean squared error* (MSE):

$$L(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i(\boldsymbol{w}))^2 \tag{2.8}$$

and the *mean absolute error* (MAE):

$$L(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i(\boldsymbol{w})| \tag{2.9}$$

For categorical data, the most commonly used loss function is the *cross-entropy*. In binary data classification, the output of the top layer of the NN is the probability $\hat{y}_i(\boldsymbol{w}) = p(y_i = 1|\boldsymbol{x}_i; \boldsymbol{w})$ that data point i is predicted to belong to category 1. The cross-entropy between the true labels $y_i \in 0, 1$ and the predictions is given by:

$$L(\boldsymbol{w}) = -\sum_{i=1}^{n} y_i \log \hat{y}_i(\boldsymbol{w}) + (1 - y_i) \log[1 - \hat{y}_i(\boldsymbol{w})] \tag{2.10}$$

For categorical data with M possible values, y can take on $0, 1, \ldots, M-1$. To handle this, we define a "one-hot" vector y_{im} for each data point i, such that:

$$y_{im} = \begin{cases} 1, & if\ y_i = m \\ 0, & otherwise. \end{cases} \tag{2.11}$$

We also define the probability that the NN assigns a data point to category m as $\hat{y}_{im}(\boldsymbol{w}) = p(y_i = m|\boldsymbol{x}_i; \boldsymbol{w})$. Then, the categorical cross-entropy is defined as:

$$L(\boldsymbol{w}) = -\sum_{i=1}^{n}\sum_{m=0}^{M-1} y_{im} \log \hat{y}_{im}(\boldsymbol{w}) + (1 - y_{im}) \log[1 - \hat{y}_{im}(\boldsymbol{w})] \tag{2.12}$$

Once the NN architecture and the loss function have been specified, gradient descent-based methods are employed to minimize the loss function and train the model. The basic idea of the gradient descent method is to iteratively update the parameters $\boldsymbol{w}$ by moving in the direction of the gradient of the loss function $\nabla_{\boldsymbol{w}} L(\boldsymbol{w})$. The most challenging aspect of training a NN compared to other ML models is the computation of the gradient.

To calculate the gradient of the loss function concerning the parameters of the NN, the *back-propagation* algorithm is utilized. The key idea behind back-propagation is to apply the chain rule for partial differentiation, starting from the output layer and propagating the gradients backward to the first layer. This process efficiently computes the gradients of the loss function with respect to each parameter in the NN. For a more comprehensive understanding of the back-propagation algorithm, including its mathematical details, you can find a complete description in the reference [1].

2.4 Explainable Deep Learning

Explainable DL (X-DL) refers to the development of ML models that can provide clear and understandable explanations for their decisions and predictions. In image analysis, X-DL aims to correlate the outputs of DL models to

specific biomedical properties of the inputs. The importance of X-DL becomes particularly evident in fields like medical image analysis, where the transparency and interpretability of a model's decision-making process can be critical for gaining the trust of physicians, regulators, and patients. While DL models have shown great success in a variety of medical diagnostic tasks, their black-box nature has limited their clinical use. Recent research in X-DL aims to address this limitation by developing methods for visualizing and interpreting the features that influence a model's decisions. These methods can be broadly classified into following key approaches:

- *Visualization Methods:* Visualization methods employ scientific visualization techniques, such as saliency maps or heatmaps, to highlight input features that significantly influence the output of a NN. These importance maps are generated by analyzing the volume of gradients passed through layers during network training or by comparing the network output of an input with that of a modified copy of the input. By visually representing the relative importance of different input features, these visualizations provide intuitive insights into the model's decision-making process, making it easier to understand how the model arrives to its predictions [11]. For example, in the reference [12], the authors addressed a DL classification problem involving X-rays of patients with COVID-19 or pneumonia. They employed a visualization X-DL approach to generate heatmaps that highlighted the regions of the X-ray that most influenced the DL model's classification.

- *Model Distillation:* Model distillation involves training a separate, "white box" ML model to mimic the input-output behavior of the DL model. Two popular techniques for explaining DL model decisions are LIME (Local Interpretable Model-Agnostic Explanations) [13] and SHAP (SHapley Additive exPlanations) [14].

  LIME is a local approximation method that explains the predictions of any black box model by approximating it with a simpler, interpretable model that is locally faithful to the original model. LIME uses perturbation-based sampling to generate a dataset of interpretable instances around the input instance of interest, and then fits a simplified model to explain the predictions of the black box model on this instance. This information is then used to create an explanation that shows which features of the input were most important for the model's prediction.

  SHAP explains the predictions of any black box model by assigning a value to each input feature, called the Shapley value, indicating its contribution to the prediction. This value is determined using a game-theoretical approach involving perturbing or removing parts of the input to identify regions in the input that the model's output is sensitive to. While SHAP is considered a local approximation method, we can run it multiple times on different input instances to obtain global explanations. For example, in the work [15], the authors introduce an interpretable prediction pipeline for classifying gene mutations based on radiomics MRI data from glioma

patients. They utilize SHAP values to evaluate the global impact of each radiomic feature on model predictions. Since the radiomic features have a specific mathematical definition, it is easier to correlate them with a physical or biological meaning.

- *Intrinsic Methods:* Intrinsic methods are a class of NNs specifically designed to generate explanations along with their predictions. Intrinsic explainability offers a more integrated approach to understanding DL models, because it includes explanation generation into the model's architecture. Two popular methods in intrinsic methods are: i) introducing attention mechanisms to a NN to create a vector of weights that accounts for the relevance of input units to the task [16], and ii) adding additional explanation task to the original model task, and jointly training the explanation task along with the original task [17]. For instance, in the work [18], the authors introduced CA-Net, a DL architecture that utilizes attention mechanisms. This model not only generates segmentation masks but also yields interpretable attention maps that highlight the regions of the input images that most influence the segmentation decision.

In summary, these techniques provide valuable tools for enhancing the explainability of DL models, making them more transparent and trustworthy for deployment in critical applications. For a comprehensive review of explainable DL algorithms and their diverse applications, please refer to the references [19], [20].

References

[1] Goodfellow, I., Bengio, Y., and Courville, A., *Deep learning*. MIT press, 2016.

[2] Mehta, P., Bukov, M., Wang, C.-H., *et al.*, "A high-bias, low-variance introduction to machine learning for physicists," *Physics Reports*, vol. 810, pp. 1–124, 2019.

[3] Géron, A., *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, 2022.

[4] Budd, S., Robinson, E. C., and Kainz, B., "A survey on active learning and human-in-the-loop deep learning for medical image analysis," *Medical Image Analysis*, vol. 71, p. 102 062, 2021.

[5] Ren, P., Xiao, Y., Chang, X., *et al.*, "A survey of deep active learning," *ACM computing surveys (CSUR)*, vol. 54, no. 9, pp. 1–40, 2021.

[6] LeCun, Y., Boser, B., Denker, J. S., *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[7] Rumelhart, D. E., Hinton, G. E., and Williams, R. J., "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[8] Hochreiter, S. and Schmidhuber, J., "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] Bengio, Y., Simard, P., and Frasconi, P., "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[10] Gers, F. A., Schmidhuber, J., and Cummins, F., "Learning to forget: Continual prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[11] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D., "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, IEEE, 2017, pp. 618–626.

[12] Panwar, H., Gupta, P., Siddiqui, M. K., Morales-Menendez, R., Bhardwaj, P., and Singh, V., "A deep learning and grad-CAM based color visualization approach for fast detection of COVID-19 cases using chest X-ray and CT-Scan images," *Chaos, Solitons & Fractals*, vol. 140, p. 110 190, 2020.

[13] Ribeiro, M. T., Singh, S., and Guestrin, C., "Model-agnostic interpretability of machine learning," *arXiv preprint arXiv:1606.05386*, 2016.

[14] Lundberg, S. M. and Lee, S.-I., "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[15] Severn, C., Suresh, K., Görg, C., Choi, Y. S., Jain, R., and Ghosh, D., "A Pipeline for the implementation and visualization of explainable machine learning for medical imaging using radiomics features," *Sensors*, vol. 22, no. 14, p. 5205, 2022.

[16] Vaswani, A., Shazeer, N., Parmar, N., *et al.*, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[17] Hendricks, L. A., Akata, Z., Rohrbach, M., Donahue, J., Schiele, B., and Darrell, T., "Generating visual explanations," in *Computer Vision–ECCV 2016*, Springer, 2016, pp. 3–19.

[18] Gu, R., Wang, G., Song, T., *et al.*, "CA-Net: Comprehensive attention convolutional neural networks for explainable medical image segmentation," *IEEE Transactions on Medical Imaging*, vol. 40, no. 2, pp. 699–711, 2020.

[19] Ras, G., Xie, N., Van Gerven, M., and Doran, D., "Explainable deep learning: A field guide for the uninitiated," *Journal of Artificial Intelligence Research*, vol. 73, pp. 329–396, 2022.

[20] Singh, A., Sengupta, S., and Lakshminarayanan, V., "Explainable deep learning models in medical image analysis," *Journal of imaging*, vol. 6, no. 6, p. 52, 2020.

Part A

Segmentation of biomedical images

Introduction

Image segmentation is the process of dividing an image into distinct and meaningful *regions of interest* (ROIs), each characterized by specific attributes such as proximity, gray level, color, texture, brightness, or contrast [1]–[3]. In the biomedical imaging field these ROIs may correspond to organs, tissues, pathologies or other biologically relevant structures. Popular biomedical image segmentation tasks include the segmentation of brain and brain tumors [4], [5], lungs and pulmonary nodules [6], [7], liver and liver tumors [8], cells [9] and more [10].

Let Ω represent the image domain. The goal of segmentation is to find sets $S_k \subset \Omega$ in such a way that their union covers the entire Ω. This can be expressed mathematically as:

$$\Omega = \bigcup_{k=1}^{K} S_k \tag{2.13}$$

where, $S_k \cap S_j = \varnothing$ for $k \neq j$ and K is the total number of regions. In certain applications, the value of K is assumed to be known based on prior knowledge of the anatomy being considered. For example, in the segmentation of MR brain images, it is common to assume that $K = 3$, representing the gray matter, white matter, and cerebrospinal fluid tissue classes. In some different scenarios, the total number of regions cannot be chosen a priori. This is particularly evident in cases where regions are fragmented or consist of non-contiguous portions due to the intricate nature of tissues or anatomical structures, as shown in Figure 2.7.

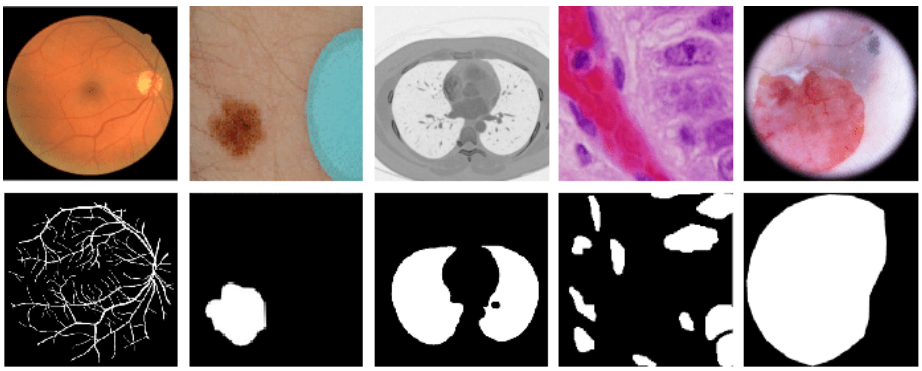


Figure 2.7: Different applications of biomedical image segmentation, i.e. (from left to right) segmentation of blood vessels, skin cancer, lung, cell nuclei and tissue classes. Source [11].

In recent years, numerous algorithms have been developed for the segmentation of biomedical images [12]. These methods can be broadly categorized into the following:

1. *Thresholding:* thresholding is a straightforward and computationally efficient technique that segments an image based on a predefined intensity threshold value. Pixels with intensity values exceeding the threshold are classified as foreground, while those below are considered background. This method is particularly effective for images with clear contrast between foreground and background regions. However, its performance may deteriorate in images with uneven illumination or intricate backgrounds [13].

2. *Region-based methods:* region-based methods group pixels into regions based on their intensity values and spatial relationships. These methods are well-suited for images containing homogeneous regions but may struggle with complex structures or overlapping regions.

    - *Region growing:* this algorithm initiates from a seed pixel and expands the region by incorporating neighboring pixels that meet specific criteria, such as intensity similarity (i.e. similarity in the pixel intensity values) or texture similarity (i.e. similarity in the spatial arrangement of pixel intensities) [14].
    - *Region split and merge:* this method involves dividing an image into quadrants and iteratively merging adjacent regions found to be uniform based on predefined rules, such as intensity, texture, size, and shape [15].
    - *Watershed approach:* watershed segmentation is a method that treats an image like a topography, with brighter pixels representing mountain peaks and darker pixels representing valleys. It identifies boundaries by flooding the image with water (i.e. mask label) from seeds placed in the valleys. Water from different seeds doesn't mix, creating dams that represent object boundaries [16].

3. *Clustering:* clustering algorithms group pixels based on their similarity in intensity values or other features [17], [18]. Techniques like k-means, an iterative method that groups data points into a predefined number of clusters where each data point belongs to the cluster with the nearest mean, and fuzzy c-means, a data clustering algorithm that allows each data point to belong to multiple clusters with varying degrees of membership, are commonly employed for this task. While clustering is beneficial for images with multiple regions, it may not perform well on images with overlapping regions. A more detailed description of clustering models for biomedical image segmentation, including k-means and fuzzy c-means, will be provided in the following sections.

4. *Edge Detection:* edge detection aims to identify the boundaries between distinct regions in an image. These boundaries, or edges, represent the transition points between areas with varying intensity levels. One widely used edge detection algorithm is the Laplacian of Gaussian (LoG) filter [19]. This algorithm operates by convolving the image with a LoG kernel, a combination of the Gaussian filter and the Laplacian filter. The Gaussian filter is first used to smooth the image, which reduces noise. The Laplacian filter is then used to find edges in the smoothed image. This convolution process highlights areas where the intensity values undergo significant changes, effectively identifying the image's edges. The LoG filter is particularly effective for images with clear and distinct boundaries between regions. However, it may encounter challenges when dealing with low-contrast or noisy images.

5. *Graph-based approaches:* these techniques represent the image as a graph, where nodes correspond to pixels and edges represent connections between pixels. The weights of the edges are determined based on the similarity or dissimilarity between neighboring pixels such as the difference in color or intensity values, effectively capturing the relationships within the image. Utilizing the graph structure, the graph-based approach transforms the image segmentation problem into a graph partition problem, enabling the application of diverse graph-partitioning algorithms. For example, the graph-cut algorithm identifies the cut on the graph with the minimum weight, where a cut is a set of edges that divides the graph into two distinct regions: foreground and background [20].

6. *Model-based:* model-based methods segment an image by employing a model that describes the features of the object of interest. These methods are particularly effective for objects with identifiable and consistent visual traits throughout the entire image.

    - *Atlas-based approach:* this method utilizes a reference image, known as atlas, to guide the segmentation of a target image. The atlas contains manually segmented regions of interest, which provide prior knowledge about the expected appearance of objects in the target image. The atlas regions are deformed to align with the target image [21], [22].
    - *Machine Learning and Deep Learning models:* unlike the previously mentioned methods, which depend on manually crafted features and explicit programming, ML and DL models can automatically learn segmentation tasks directly from labeled examples. This is facilitated by their capacity to extract significant features directly from raw images, eliminating the necessity for manual feature engineering. In particular, it has been demonstrated that CNNs excel at identifying intricate patterns and relationships within images, leading to state-of-the-art performance in various biomedical segmentation tasks [10], [23]. A more comprehensive discussion of DL models for biomedical image segmentation will be presented in the following sections.

Nevertheless, the process of segmenting biomedical images presents three fundamental challenges related to specific image characteristics [3]. The first challenge originates from the noise introduced by sensors and electronic systems. This noise can potentially alter pixel intensities and introduce uncertainty into their categorization. The second challenge is related to intensity inhomogeneity, in which homogeneous tissues appear non-uniform in the image due to gradual variations in intensity levels across its extent. Examples of this phenomenon encompass motion artifacts, ring artifacts, or ghosting. The third issue is related to the finite pixel dimensions that lead to the partial volume effect, where individual pixel volumes encompass mixtures of different tissue classes, causing pixel intensities to deviate from a single class assignment. All these factors combine to make biomedical image segmentation a highly complex task.

To effectively assess the performance of biomedical image segmentation models, a large number of metrics have been introduced to evaluate their segmentation quality [10]. These metrics provide quantitative measures of the similarity between the segmented regions and ground truth annotations created by a medical expert. In the following, we give some insights of some popular metrics for evaluating the performance of medical image segmentation. These metrics will be used to evaluate the performance of the segmentation algorithms presented in subsequent works.

The first metric is the Dice Similarity Coefficient (DSC_{metric}), which quantifies the overlap between the segmented region and the ground truth region. It is defined as:

$$DSC_{metric} = \frac{2 \times |S_{true} \cap S_{est}|}{|S_{true}| + |S_{est}|} \tag{2.14}$$

where S_{true} represents the corresponding ground truth region for the segmented region S_{est}. The DSC_{metric} ranges from 0 to 1, where 1 indicates a perfect match between the segmentation and the ground truth.

Another important metric is the Jaccard Index (JI_{metric}), also known as the Intersection over Union (IoU). It is defined as:

$$JI_{metric} = \frac{|S_{true} \cap S_{est}|}{|S_{true} \cup S_{est}|} \tag{2.15}$$

Similar to the DSC_{metric}, the Jaccard Index provides a measure of how well the segmented region aligns with the ground truth region. It also ranges from 0 to 1, with 1 representing a perfect match.

Furthermore, segmentation metrics can also be used in the training phase of neural networks for image segmentation tasks. When these metrics are differentiable with respect to the model's parameters, they can be utilized as loss functions during the training process. In addition to the previously defined metrics (where $DSC_{loss} = 1 - DSC_{metric}$ and $JI_{loss} = 1 - JI_{metric}$), a commonly employed function for training purposes is the Cross-Entropy, as defined in Equation 2.12. In the context of image segmentation, where the task involves classifying each pixel into different categories corresponding to different

ROIs, the Cross-Entropy is used to assess how well the predicted pixel-wise probability distribution aligns with the ground truth distribution. This loss function would have a value of zero only if there is a perfect match between the segmentation mask predicted by the model and the true segmentation mask. Moreover, in cases where classes are imbalanced, which occurs when the ROI area is much smaller than the background area, a weighted Cross-Entropy can be employed. This approach involves assigning distinct weights w_c to each class c, which take into account class frequencies or other weighting factors relevant for the task.

The two works that will be described in the following paragraphs are located in this context. The first study, detailed in paper I, presents a clustering model based on statistical mechanics. The second research project, discussed in paper II, describes a DL model designed for the automatic segmentation and quantification of COVID-19 pulmonary damage. The following paragraphs will provide an overview of clustering methods, as well as an introduction to the U-Net algorithm used for image segmentation. These principles constitute the fundamental basis of the two segmentation models that will be discussed in the next sections.

Clustering for segmentation of biomedical images

In the field of image analysis, a strategy often adopted to perform image segmentation involves the use of clustering algorithms [1], [12]. Clustering is an unsupervised learning technique whose goal is to categorize unlabeled data into clusters, using measures of similarity or distance. Specifically, a cluster is defined as a collection of points that exhibit a common pattern or structure. Clustering is one of the main examples of unsupervised learning models, since it doesn't rely on labeled data for its operation.

The task of image segmentation can be understood as a clustering process, where the goal is to group similar pixels together based on a specific feature vector. This feature vector could represent a range of attributes extracted from the local neighborhood of each pixel, such as pixel intensity, color value, pixel intensity variations, and other characteristics that offer insights into the visual content of the image. By analyzing these feature vectors, the clustering algorithm identifies and groups pixels with similar feature vectors, effectively dividing the image into homogeneous regions.

One widely used clustering algorithm is *k-means*. This algorithm iteratively computes mean intensities for each cluster and subsequently segments the image by attributing each pixel to the cluster whose mean intensity is closest. Another example is the *fuzzy c-means* algorithm, which extends the k-means algorithm's functionality by facilitating adaptable segmentations thanks to the principles of fuzzy set theory. This theory allows data points to exist in multiple clusters with varying degrees of membership, resulting in a more versatile representation of the underlying data structure.

However, it has been shown in the works [1], [24] that clustering algorithms can be sensitive to noise and intensity inhomogeneities, which are features often encountered in biomedical images. In this context, in paper I we propose

a clustering method based on statistical mechanics to specifically segment biomedical images even in the presence of noise or intensity inhomogeneities.

Unet

A different approach to image segmentation involves the use of deep neural networks, especially CNNs. CNNs excel at capturing intricate patterns and relationships within the images, leading to state-of-the-art performance in many biomedical image segmentation tasks. The *U-Net* architecture is a specialized CNN design developed primarily to address image segmentation problems. Its structure takes the form of the letter "U" composed by an encoder path, a decoder path, and a central connecting middle block. This innovative architecture was first introduced in 2015 in the work by Ronneberger et al. [25]

In more detail, the encoder path is constructed with a series of convolutional and pooling layers. This configuration enables the extraction of higher-level features from the input image, progressively reducing its spatial dimensions while enhancing its feature depth. Subsequently, the decoder path is composed of transposed convolutions and upscaling layers. These operations are employed to restore the initial spatial resolution of the encoded features. A middle block, consisting of several convolutional layers, connects the encoder and decoder.

The exceptional segmentation performance is determined by skip connections i.e. links between the encoder and decoder paths. Skip connections consist of concatenating feature vectors from the encoder to the decoder path. These connections facilitate the extraction of local features and play a crucial role in enhancing the network's ability to distinguish intricate boundaries.

Through this architecture, the U-Net integrates high-resolution features with fine-grained details, enhancing the precision of localization in segmentation tasks. The U-Net architecture demonstrates a remarkable capability to handle diverse image sizes while preserving intricate details, making it highly suitable for a variety of biomedical imaging segmentation tasks.

A kinetic approach to consensus-based segmentation of biomedical images

In the first paper I of this chapter, we propose a novel segmentation method that uses a clustering model based on statistical mechanics. This approach is in line with statistical mechanics-based techniques that have been increasingly applied across a diverse range of social and biological applications [26]–[28].

The central concept behind these approaches is to employ the mathematical framework from the physics of gases to describe the behavior of groups of interacting agents. In the dynamics of gases, molecules are characterized by properties such as position and velocity and interact with each other through the exchange of kinetic energy. Similarly, other categories of agents may exhibit distinct microscopic states and interact based on application-specific rules. As in the case of gases collective properties emerge from the interactions of a

multitude of particles, the same phenomenon can occur in social or biological systems. Furthermore, when dealing with a large number of agents, a statistical description becomes feasible. In particular, since the total number of agents is assumed constant, it is possible to introduce a probability distribution function to describe the statistical distribution of the states of the agents. The evolution of this probability distribution function follows the Boltzmann equation, which, however, can pose challenges for analytical solutions. Consequently, numerical techniques are widely adopted, such as Monte Carlo direct simulations (DSMC) in the form of Nanbu-Babovsky [29] or Bird schemes [30].

In more detail, the primary objective of our work is to extend the clustering model proposed in [31] for image segmentation in the biomedical imaging field. The model is based on the Hegselmann-Krause aggregation model, originally designed for opinion dynamics but adapted to handle image data. The key idea of this approach is to represent each pixel as a particle with a time-dependent position vector and a static feature (gray level). These particles interact until they reach an equilibrium state, resulting in the formation of clusters that represent the segmentation regions. To enhance the model's flexibility, we introduce a non-constant diffusion term dependent on the pixel's gray level, allowing us to account for stochastic variations in the image acquisition process. By incorporating the diffusion coefficient into the binary interaction rules, we derive the Boltzmann formulation of the model. This formulation enables the development of a DSMC, making the simulation of binary collision dynamics computationally efficient. The proposed extension overcomes the time-consuming limitations of the random subset algorithm proposed in [31], making it applicable to high-dimensional biomedical images.

Additionally, we propose an optimization strategy for internal parameter configuration. Using a suitable loss function measuring the distance between the ground truth segmentation mask and the evaluated mask, we determine the optimal configuration of the parameters that minimizes the introduced segmentation metric. We test this system on three different biomedical datasets: the HL60 cell nuclei dataset, the brain tumor dataset, and the thigh muscles dataset. Promising results are achieved for the HL60 cell nuclei and brain tumor datasets, while segmentation precision is lower for the thigh muscles dataset. To address this complexity, we propose a patch-based approach, dividing the image into smaller arrays and applying the segmentation system to subregions. This improves the quality of the segmentation mask in the thigh muscles dataset.

Quantification of pulmonary involvement in COVID-19 pneumonia by means of a cascade of two U-nets: training and assessment on multiple datasets using different annotation criteria

In the second paper II of this chapter, we propose a fully automated DL system to segment and quantify coronavirus disease 2019 (COVID-19) pulmonary lesions from CT images. COVID-19 is a highly infectious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). It has been observed

that COVID-19 can affect various organ systems, including the lungs, brain, kidneys, liver, gastrointestinal tract, and cardiovascular system. One of the most common and visible complications of COVID-19, even in the early stages, is pneumonia. Chest CT scans are commonly employed to evaluate COVID-19-related pneumonia due to their significance in diagnosing and assessing various lung conditions. Common findings on CT scans of affected subjects include ground-glass opacities and consolidations. These lesions often exhibit patchy or bilateral distributions and can impact different lung lobes. Moreover, their appearance may evolve over time in response to disease progression or treatment. Quantifying the extent of lung lesions in COVID-19 patients may offer insight into disease severity, assisting in patient management and treatment strategies [32], [33].

A widely adopted evaluation approach is the CT-Severity Score (CT-SS), which evaluates the volume of lesion opacifications and total lung volume, assigning a score based on their percentage of involvement. CT-SS has been demonstrated to directly correlate with disease severity [34]. However, distinguishing COVID-19 lesions from healthy lung tissue on CT images can be challenging due to variations in texture, shape, and distribution within the lung among different patients. These factors pose significant obstacles to mechanistic segmentation methods. Conversely, DL-based approaches such as CNNs are valuable tools for effectively addressing this complex task.

In our work, we propose the DL-based method, named the LungQuant system, for the automated quantification of CT-SS. Our analysis pipeline consists of a cascade of two U-nets. The first U-net is dedicated to identifying the lung parenchyma, while the second one focuses on detecting COVID-19 lesions within the segmented lungs. The system outputs the estimated percentage of compromised lung volume, calculated as the ratio between the volume of COVID-19 lesions and the total lung volume. This percentage is further converted into CT-SS. The system was trained on publicly available data and evaluated using different datasets, heterogeneously populated and annotated according to different criteria. Our results reveal that the performance of the DL-based segmentation is influenced by the quality of annotations in the datasets.

In the following work [35], we refined the LungQuant system by incorporating an additional CNN to crop CT scans closer to the lungs, achieving stable performance across different CT-SS classes. Additionally, the system underwent validation in a multicentric study [36] involving 14 clinical experts, confirming its accuracy and effectiveness in classifying CT-SS.

References

[1] Pham, D. L., Xu, C., and Prince, J. L., "Current methods in medical image segmentation," *Annual Review of Biomedical Engineering*, vol. 2, no. 1, pp. 315–337, 2000.

[2] Pal, N. R. and Pal, S. K., "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277–1294, 1993.

[3] Sharma, N. and Aggarwal, L. M., "Automated medical image segmentation techniques," *Journal of Medical Physics*, vol. 35, no. 1, p. 3, 2010.

[4] Menze, B. H., Jakab, A., Bauer, S., *et al.*, "The multimodal brain tumor image segmentation benchmark (BRATS)," *IEEE Transactions on Medical Imaging*, vol. 34, no. 10, pp. 1993–2024, 2014.

[5] Cherukuri, V., Ssenyonga, P., Warf, B. C., Kulkarni, A. V., Monga, V., and Schiff, S. J., "Learning based segmentation of CT brain images: Application to postoperative hydrocephalic scans," *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 8, pp. 1871–1884, 2017.

[6] Wang, S., Zhou, M., Liu, Z., *et al.*, "Central focused convolutional neural networks: Developing a data-driven model for lung nodule segmentation," *Medical Image Analysis*, vol. 40, pp. 172–183, 2017.

[7] Onishi, Y., Teramoto, A., Tsujimoto, M., *et al.*, "Multiplanar analysis for pulmonary nodule classification in CT images using deep convolutional neural network and generative adversarial networks," *International Journal of Computer Assisted Radiology and Surgery*, vol. 15, pp. 173–178, 2020.

[8] Seo, H., Huang, C., Bassenne, M., Xiao, R., and Xing, L., "Modified U-Net (mU-Net) with incorporation of object-dependent high level features for improved liver and liver-tumor segmentation in CT images," *IEEE Transactions on Medical Imaging*, vol. 39, no. 5, pp. 1316–1325, 2019.

[9] Song, T.-H., Sanchez, V., EIDaly, H., and Rajpoot, N. M., "Dual-channel active contour model for megakaryocytic cell segmentation in bone marrow trephine histology images," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 12, pp. 2913–2923, 2017.

[10] Wang, R., Lei, T., Cui, R., Zhang, B., Meng, H., and Nandi, A. K., "Medical image segmentation using deep learning: A survey," *IET Image Processing*, vol. 16, no. 5, pp. 1243–1267, 2022.

[11] Asadi-Aghbolaghi, M., Azad, R., Fathy, M., and Escalera, S., "Multi-level context gating of embedded collective knowledge for medical image segmentation," *arXiv preprint arXiv:2003.05056*, 2020.

[12] Ramesh, K., Kumar, G. K., Swapna, K., Datta, D., and Rajest, S. S., "A review of medical image segmentation algorithms," *EAI Endorsed Transactions on Pervasive Health and Technology*, vol. 7, no. 27, e6–e6, 2021.

[13] Batenburg, K. J. and Sijbers, J., "Adaptive thresholding of tomograms by projection distance minimization," *Pattern Recognition*, vol. 42, no. 10, pp. 2297–2305, 2009.

[14] Thakur, A. and Anand, R. S., "A local statistics based region growing segmentation method for ultrasound medical images," *International Journal of Medical and Health Sciences*, vol. 1, no. 10, pp. 564–569, 2007.

[15] Manousakas, I., Undrill, P., Cameron, G., and Redpath, T., "Split-and-merge segmentation of magnetic resonance medical images: Performance evaluation and extension to three dimensions," *Computers and Biomedical Research*, vol. 31, no. 6, pp. 393–412, 1998.

[16] Dhage, P., Phegade, M., and Shah, S., "Watershed segmentation brain tumor detection," in *2015 International Conference on Pervasive Computing (ICPC)*, IEEE, 2015, pp. 1–5.

[17] Ahmed, M. N., Yamany, S. M., Mohamed, N., Farag, A. A., and Moriarty, T., "A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data," *IEEE Transactions on Medical Imaging*, vol. 21, no. 3, pp. 193–199, 2002.

[18] Aneja, D. and Rawat, T. K., "Fuzzy clustering algorithms for effective medical image segmentation," *International Journal of Intelligent Systems and Applications*, vol. 5, no. 11, pp. 55–61, 2013.

[19] Kong, H., Akakin, H. C., and Sarma, S. E., "A generalized Laplacian of Gaussian filter for blob detection and its applications," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1719–1733, 2013.

[20] Chen, X. and Pan, L., "A survey of graph cuts/graph search based medical image segmentation," *IEEE Reviews in Biomedical Engineering*, vol. 11, pp. 112–124, 2018.

[21] Isgum, I., Staring, M., Rutten, A., Prokop, M., Viergever, M. A., and Van Ginneken, B., "Multi-atlas-based segmentation with local decision fusion—application to cardiac and aortic segmentation in CT scans," *IEEE Transactions on Medical Imaging*, vol. 28, no. 7, pp. 1000–1010, 2009.

[22] Cabezas, M., Oliver, A., Lladó, X., Freixenet, J., and Cuadra, M. B., "A review of atlas-based segmentation for magnetic resonance brain images," *Computer Methods and Programs in Biomedicine*, vol. 104, no. 3, e158–e177, 2011.

[23] Hesamian, M. H., Jia, W., He, X., and Kennedy, P., "Deep learning techniques for medical image segmentation: Achievements and challenges," *Journal of Digital Imaging*, vol. 32, pp. 582–596, 2019.

[24] Beevi, Z. and Sathik, M., "A robust segmentation approach for noisy medical images using fuzzy clustering with spatial probability," *Methods*, vol. 29, no. 37, p. 38, 2009.

[25] Ronneberger, O., Fischer, P., and Brox, T., "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, Springer, 2015, pp. 234–241.

[26] Preziosi, L., Toscani, G., and Zanella, M., "Control of tumor growth distributions through kinetic methods," *Journal of Theoretical Biology*, vol. 514, p. 110 579, 2021.

[27] Loy, N. and Preziosi, L., "Kinetic models with non-local sensing determining cell polarization and speed according to independent cues," *Journal of Mathematical Biology*, vol. 80, no. 1-2, pp. 373–421, 2020.

[28] Toscani, G., "Kinetic models of opinion formation," *Communications in Mathematical Sciences*, vol. 4, no. 3, pp. 481–496, 2006.

[29] Nanbu, K., "Direct simulation scheme derived from the Boltzmann equation. I. Monocomponent gases," *Journal of the Physical Society of Japan*, vol. 49, no. 5, pp. 2042–2049, 1980.

[30] Bird, G., "Direct simulation and the Boltzmann equation," *The Physics of Fluids*, vol. 13, no. 11, pp. 2676–2681, 1970.

[31] Herty, M., Pareschi, L., and Visconti, G., "Mean field models for large data–clustering problems," *Networks and Heterogeneous Media*, vol. 15, no. 3, pp. 463–487, 2020.

[32] Guan, X., Yao, L., Tan, Y., *et al.*, "Quantitative and semi-quantitative CT assessments of lung lesion burden in COVID-19 pneumonia," *Scientific Reports*, vol. 11, no. 1, p. 5148, 2021.

[33] Ye, Z., Zhang, Y., Wang, Y., Huang, Z., and Song, B., "Chest CT manifestations of new coronavirus disease 2019 (COVID-19): A pictorial review," *European Radiology*, vol. 30, pp. 4381–4389, 2020.

[34] Yang, R., Li, X., Liu, H., *et al.*, "Chest CT severity score: An imaging tool for assessing severe COVID-19," *Radiology: Cardiothoracic Imaging*, vol. 2, no. 2, e200047, 2020.

[35] Lizzi, F., Postuma, I., Brero, F., *et al.*, "Quantification of pulmonary involvement in COVID-19 pneumonia: An upgrade of the LungQuant software for lung CT segmentation," *The European Physical Journal Plus*, vol. 138, no. 4, pp. 1–10, 2023.

[36] Scapicchio, C., Chincarini, A., Ballante, E., *et al.*, "A multicenter evaluation of a deep learning software (LungQuant) for lung parenchyma characterization in COVID-19 pneumonia," *European Radiology Experimental*, vol. 7, no. 1, p. 18, 2023.

Paper I

A kinetic approach to consensus-based segmentation of biomedical images

Raffaella Fiamma Cabini[1,2], Anna Pichiecchio[3,4], Alessandro Lascialfari[5,2], Silvia Figini[6], Mattia Zanella[1]

Under review in *Mathematics and Computers in Simulation*

Abstract

In this work, we apply a kinetic version of a bounded confidence consensus model to biomedical segmentation problems. In the presented approach, time-dependent information on the microscopic state of each particle/pixel includes its space position and a feature representing a static characteristic of the system, i.e. the gray level of each pixel. From the introduced microscopic model we derive a kinetic formulation of the model. The large time behavior of the system is then computed with the aid of a surrogate Fokker-Planck approach that can be obtained in the quasi-invariant scaling. We exploit the computational efficiency of direct simulation Monte Carlo methods for the obtained Boltzmann-type description of the problem for parameter identification tasks. Based on a suitable loss function measuring the distance between the ground truth segmentation mask and the evaluated mask, we minimize the introduced segmentation metric for a relevant set of 2D gray-scale images. Applications to biomedical segmentation concentrate on different imaging research contexts.

Keywords: image segmentation; kinetic modelling; consensus models; particle systems.

[1] Department of Mathematics, University of Pavia, Via Ferrata 5, 27100 Pavia, Italy
[2] INFN, Istituto Nazionale di Fisica Nucleare, Pavia Unit, Via Bassi 6, 27100, Pavia, Italy
[3] Department of Brain and Behavioural Sciences, University of Pavia, Via Mondino 2, 27100 Pavia, Italy
[4] Advanced Imaging and Radiomic Center, IRCCS Mondino Foundation, Via Mondino 2, 27100 Pavia, Italy
[5] Department of Physics, University of Pavia, Via Bassi 6, 27100, Pavia, Italy
[6] Department of Social and Political Science, University of Pavia, Corso Carlo Alberto 3, 27100 Pavia, Italy

I.1 Introduction

In image processing and computer vision, image segmentation is a fundamental process to subdivide images in subsets of pixels that has found application in many research contexts [1]. In the field of medical imaging, the identification of image subregions is a powerful tool for tissue recognition to track pathological changes. Image segmentation can help clinical studies of anatomical structures and in the identification of regions of interest, and to measure tissue volume for clinical purposes.

The main goal of image segmentation is to divide the image into a set of pixel regions that share similar properties such as closeness, gray level, color, texture, brightness, and contrast [2]. By converting an image into a group of segments, it is possible to process only the important areas instead of studying the entire image. To this end, various computational strategies and mathematical methods have been developed in the last decades. Among them, Neural Networks (NNs) are one of the most common strategies used in modern image segmentation problems. These techniques can approximate, starting from a series of examples, the nonlinear function between the inputs and the outputs of interest. It has been observed that well trained NNs can achieve good segmentation accuracy even with complex images [3]–[6]. Anyway, the approximation obtained through NNs may require extensive supervised training procedure. Indeed, the performances of NNs essentially depend on the adopted training procedure and on the availability of unbiased data [7].

A different approach to image segmentation is based on clustering techniques such as the k–means method, the c–means method, hierarchical clustering method and genetic algorithms [8], [9]. We refer to clustering process as a dynamics to identifying groups of similar data points according to some observed characteristics. These strategies belong to the category of the unsupervised algorithms since they do not require a training procedure and do not depend on training datasets. It can be observed that image segmentation is a clustering process where the pixels are classified into multiple distinct regions so that pixels within each group are homogeneous with respect to certain features, while pixels in different groups are different from each other [10]–[13].

In this work, following the approach introduced in the recent work [14] for clustering problems, we adopt a mathematical strategy to image segmentation that is based on consensus dynamics of large systems of agents. In this direction, we adopt a kinetic-type approach by rewriting generalized Hegselmann-Krause (HK) microscopic consensus models in terms of a binary scheme. The evolutions of aggregate quantities are then obtained through a Boltzmann-type model whose steady state can be approximated by means of a quasi-invariant approach [15]. Indeed, in the mentioned scaling, a reduced complexity Fokker-Planck model corresponds, in the zero-diffusion limit, to the mean-field model defined in [14]. Suitable steady state preserving numerical methods can be applied to verify the consistency of the approach.

Following the approach in [14], each pixel is represented by a particle characterized by a time-dependent position vector and a static feature that

describes an intrinsic property of the particle, i.e. the gray level of the pixel. Particles interact with each other until they reach the equilibrium state in which they group into a finite number of clusters. Hence, a segmentation mask is generated by assigning the mean of their gray levels to each cluster of particles and by applying a binary threshold. The two main advantages of this segmentation system compared to other standard clustering techniques such as the k–means method, are that the clustering process also takes into account the gray level of the pixels and not just their reciprocal positions and it is not necessary to select in advance the final number of the clusters.

In the following, we apply a particle-based clustering method to biomedical image segmentation problems. In particular, we will incorporate in the model a non-constant diffusion term which depends on the gray level of the pixel which consent to quantify aleatoric uncertainties in the segmentation pipeline. The aleatoric uncertainties derive from image acquisition processes and can affect the quality of the considered image.

The Boltzmann-type formulation of the problem allows to apply a direct Monte Carlo method to simulate the binary collision dynamics with a computational cost directly proportional to the number of particles N of the system, as described in [15]. The computational efficiency is important for parameter identification purposes since the segmentation masks to optimize corresponds to the large time distribution of the system. Applications to biomedical images will be presented to test the performance of the proposed method. In more detail, the manuscript is organized as follows. In Section I.2 we introduce the generalized Hegselmann-Krause model for image segmentation. The evolution of aggregate quantities are then obtained by means of a Boltzmann-type equation in Section I.3.1. We derived a surrogate Fokker-Planck model for which the large time distribution can be efficiently computed and compared with the one of the Boltzmann-type model. In Section I.4 we describe how to generate segmentation masks and in section I.4.1 the strategy to optimize the model parameters is discussed. In Section I.4.2 we apply the proposed segmentation pipeline to three different biomedical image datasets. Finally, in Section I.4.3 we propose a patch-based version of the method that we apply for the segmentation of Magnetic Resonance Images of the thigh muscles. Our numerical experiments show that the introduced segmentation method achieves good performances for the tested images.

I.2 Modelling consensus dynamics

The study of consensus formation has gained great interest in the field of opinion dynamics to understand the basic ingredients underlying the phenomena of choice formation in connected communities. Several mathematical models of consensus have been proposed in the form of systems of first order ordinary differential equations (ODEs) or binary algorithms describing the behaviour of a finite number of particles/agents. In this direction, we mention the pioneering works [16], [17] where simple agent-based models were introduced to observe the

relative influence among individuals, see also [18] for a stochastic version of the above information exchange processes.

In recent years, several differential models of consensus have been introduced to understand the underlying social forces of social phenomena. Without intending to review all the literature, we point the interested readers to some references for finite systems: in [19], [20] simple Ising spin models have been introduced to mimic the mechanisms of decision making in closed communities, in [21] a binary scheme is adopted to measure the convergence towards consensus under interaction limitations. Furthermore, models incorporating leader-follower effect have been considered in [22] and models of social interaction on networks have been studied in [23]. Finally, in [24] a microscopic modelling approach is considered to measure convergence towards consensus in the case of asymmetric interactions. For a review we mention [25]. We highlight how consensus-like dynamics may model heterogeneous phenomena, see e.g. flocking dynamics [26], [27] or economic interactions [28], [29].

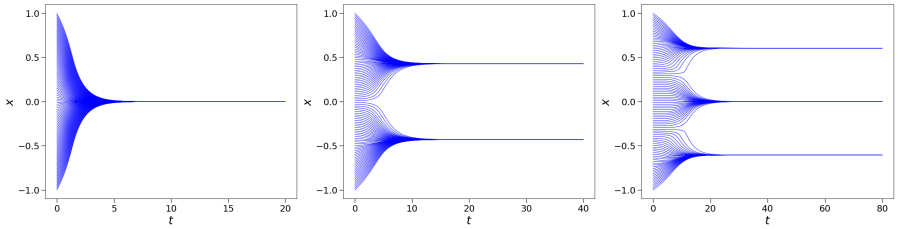
Beside microscopic particle-based models for consensus dynamics, in the limit of infinitely many agents, it is possible to cope with the evolution of distribution functions characterizing the aggregate trends of the interacting systems. These approaches are generally based on kinetic-type partial differential equations (PDEs) and are capable of linking microscopic forces to emerging features of the system. In this direction we mention [30]–[36] and the references therein. For macroscopic models of consensus dynamics we point the interested reader to [37], [38].

I.2.1 The bounded confidence model

In the following, we consider a population of $N \geq 2$ particles characterized by an initial state $x_i(0) \in X \subseteq \mathbb{R}$. Each particle $i \in \{1, \dots, N\}$ modifies its state $x_i \in X$ through the interaction with the particle $j \in \{1, \dots, N\}$, $j \neq i$, only if x_i is sufficiently close to x_j, i.e. $|x_i - x_j| \leq \Delta$, being $\Delta > 0$ a suitable threshold. This model stresses the homophily in learning processes and is known as Hegselmann-Krause (HK) model [39], see [40] for a survey. At the time-continuous level, the dynamics of the ith particle may be suitably defined as follows

$$\frac{d}{dt}x_i = \alpha \sum_{j=1}^{N} \chi(|x_j - x_i| \leq \Delta)(x_j - x_i) \qquad i = 1, \dots, N \tag{I.1}$$

where $\chi(\cdot)$ is a characteristic function and defines the bounded confidence interaction scheme, and $\alpha > 0$ is a suitable scaling constant measuring that contribution of each interaction. Generally, it is assumed $\alpha = 1/N$ such that, if $X = [-1, 1]$ and $\Delta = 2$ the system converges towards the mean $\bar{x} = \frac{1}{N}\sum_{j=1}^{N} x_j(t)$. Furthermore, it has been proved that the HK model converges to a steady configuration where the initial states are grouped in a finite number of clusters, see [24], [41], [42]. In Figure III.1 we show the dynamics of a system of $N = 100$ particles according to the HK model for different values of the threshold $\Delta > 0$. We may easily observe how multiple clusters appear for small values of $\Delta > 0$.



(a) $\Delta = 1$ (b) $\Delta = 0.5$ (c) $\Delta = 0.3$

Figure I.1: Results of the Hegselmann-Krause bounded confidence model for three different values of the Δ threshold. At the initial time we selected $N = 100$ particles equally spaced in $[-1, 1]$.

I.2.2 Consensus models in segmentation problems

An application of the HK model to image segmentation problems has been proposed in [14]. The key idea of this approach is to link each particle $i \in \{1, \dots, N\}$ to a time-dependent position $\mathbf{x}_i = (x_i(y), y_i(t)) \in \mathbb{R}^2$ and a scalar quantity expressing a feature $c_i \in [0, 1]$ and corresponding to a static characteristic of the system. In particular, in the following $c_i \in [0, 1]$ expresses the gray level of the ith pixel. In this setting, the consensus process defined in (I.1) depends also on the feature of the system and assumes the following form

$$\begin{aligned} \frac{d}{dt}\mathbf{x}_i &= \frac{1}{N}\sum_{j=1}^{N} P_{\Delta_1,\Delta_2}(\mathbf{x}_i, \mathbf{x}_j, c_i, c_j)(\mathbf{x}_j - \mathbf{x}_i) \\ \frac{d}{dt}c_i &= 0 \quad i = 1, \dots, N \end{aligned} \tag{I.2}$$

where $\mathbf{x}_i(0) = \mathbf{x}_i^0$, $c_i(0) = c_i^0$ and where we introduced the interaction function

$$P_{\Delta_1,\Delta_2}(\mathbf{x}_i, \mathbf{x}_j, c_i, c_j) = \chi(|\mathbf{x}_j - \mathbf{x}_i| \leq \Delta_1)\chi(|c_j - c_i| \leq \Delta_2) \tag{I.3}$$

and $\Delta_1 \geq 0$ and $\Delta_2 \geq 0$ are the confidence intervals of the position vectors and of the gray levels, respectively. In (I.3) the function $\chi(\cdot)$ is the characteristic function. The two introduced scalar and image-dependent quantities are particularly important to determine the optimal segmentation masks. Indeed, they determine the confidence level under which the ith pixel tends to form a cluster through interactions with the whole set of pixels determining the image.

It is important to highlight that biomedical images are often affected by ambiguities due to several sources of uncertainty linked to both clinical factors and to possible bottlenecks in data acquisition processes [43], [44]. Among them it is possible to distinguish two major types of uncertainty, we may refer to the first as aleatoric uncertainty and is linked to stochasticities in the data collection process. In this case, we have to face reconstruction problems in which the image processing models suffer raw acquisition data. On the other hand,

the second kind of uncertainty is of epistemic-type and determines deviation of model parameters. In particular, in medical imaging MRI (Magnetic Resonance Imaging) scans may lead to ambiguous segmentation outputs [45]. In this regard, the study of uncertainty quantification in image segmentation is a growing field to produce robust segmentation algorithms that are capable of avoiding erroneous results.

Therefore, in order to take into account aleatoric-type uncertainties affecting the correct feature of an image we consider a stochastic consensus model. In particular, we concentrate on the stochastic version of (I.2) whose form is defined as follows

$$\begin{aligned} d\mathbf{x}_i &= \frac{1}{N}\sum_{j=1}^{N} P_{\Delta_1,\Delta_2}(\mathbf{x}_i,\mathbf{x}_j,c_i,c_j)(\mathbf{x}_j-\mathbf{x}_i)dt + \sqrt{2\sigma^2 D(c_i)}d\mathbf{W}_i \\ \frac{d}{dt}c_i &= 0, \quad i=1,\dots,N \end{aligned} \tag{I.4}$$

where the interaction function is compatible with (I.3), $\mathbf{x}_i(0) \in \mathbb{R}^2$, $c_i(0) \in [0,1]$ and $\mathbf{W}_i$'s are independent Wiener processes. Furthermore, since marked deviations can be expected far from the boundaries of the features' domain, we consider a non-uniform impact of the introduced aleatoric uncertainty. To this end, we consider a local diffusion function $D(c)$ such that $D(0) = D(1) = 0$. A possible form of this function is $D(c) = c(1-c)$, $c \in [0,1]$. In (I.4) the diffusion is weighted by the parameter $\sigma^2 > 0$. In Section I.4.4, other choices of the local diffusion function $D(c)$ will be considered.

I.3 Kinetic models for image segmentation

In [14] the mean-field limit of the generalized HK model is studied and it is formally argued that for $N \to \infty$ a system of particles whose dynamics is (I.2) can be described in terms of the following nonlocal partial differential equation

$$\begin{aligned} &\partial_t f(\mathbf{x},c,t) = \nabla_{\mathbf{x}} \cdot [\mathscr{B}[f]_{\Delta_1,\Delta_2}(\mathbf{x},c,t))f(\mathbf{x},c,t)] \\ &f(\mathbf{x},\mathbf{c},0) = f_0(\mathbf{x},\mathbf{c}) \end{aligned} \tag{I.5}$$

where $\mathscr{B}_{\Delta_1,\Delta_2}[f](t,\mathbf{x},c)$ is the operator that describes the interaction dynamics and is defined as

$$\mathscr{B}_{\Delta_1,\Delta_2}[f](\mathbf{x},c,t) = \int_0^1 \int_{\mathbb{R}^2} P_{\Delta_1,\Delta_2}(\mathbf{x},\mathbf{x}_*,c,c_*)(\mathbf{x}-\mathbf{x}_*)f(\mathbf{x}_*,c_*,t)d\mathbf{x}_* dc_*, \tag{I.6}$$

where the interaction function $P_{\Delta_1,\Delta_2}(\mathbf{x},\mathbf{x}_*,c,c_*)$ is defined as in (I.3). In the following, we argue how the same model can be obtained in suitable limits from binary interaction dynamics.

I.3.1 Boltzmann-type derivation

In order to define consensus dynamics from the point of view of kinetic theory we set up a consistent binary scheme defining the interactions between pixels.

To this end, inspired by [15], [46], we consider the dynamics defined in (I.4) for two particles characterized by positions $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^2$ and features $c_i, c_j \in [0,1]$. Hence, we introduce a time discretization at the level of particles with time step $\epsilon > 0$. Setting

$$\begin{aligned} &\mathbf{x} := \mathbf{x}_i(t), \quad \mathbf{x}_* := \mathbf{x}_j(t), \quad \mathbf{x}' = \mathbf{x}_i(t+\epsilon), \quad \mathbf{x}'_* = \mathbf{x}_j(t+\epsilon), \\ &c := c_i(t), \quad c_* := c_j(t), \quad c' := c_i(t+\epsilon), \quad c'_* := c_j(t+\epsilon), \end{aligned}$$

we may discretise the stochastic differential equation with Euler-Maruyama scheme with time step $\epsilon > 0$ to obtain the following binary dynamics

$$\begin{aligned} \mathbf{x}' &= \mathbf{x} + \epsilon P_{\Delta_1,\Delta_2}(\mathbf{x}, \mathbf{x}_*, c, c_*)(\mathbf{x}_* - \mathbf{x}) + \sqrt{2\sigma^2 D(c)}\eta \\ c' &= c \\ \mathbf{x}'_* &= \mathbf{x}_* + \epsilon P_{\Delta_1,\Delta_2}(\mathbf{x}_*, \mathbf{x}, c_*, c)(\mathbf{x} - \mathbf{x}_*) + \sqrt{2\sigma^2 D(c_*)}\eta \\ c'_* &= c_* \end{aligned} \tag{I.7}$$

where $\eta = (\eta_i, \eta_j)$ is a centered 2D Gaussian random variable such that

$$\langle \eta \rangle = (0,0) \qquad \langle \eta_i \eta_j \rangle = \Sigma_{ij}, \qquad i,j = 1,2 \tag{I.8}$$

where $\langle \cdot \rangle$ denotes the integration with respect to the distribution of η and $\Sigma = (\Sigma_{ij})_{i,j=1,2}$ is a diagonal matrix with unitary diagonal components. A first important consequence of the binary scheme (I.7) with $D \equiv 0$ is that the support of the positions can not increase. Indeed, since $P_{\Delta_1,\Delta_2} \in [0,1]$ and $\epsilon \in (0,1)$ we have

$$\begin{aligned} |\mathbf{x}| &= |(1 - \epsilon P_{\Delta_1,\Delta_2})\mathbf{x} + \epsilon P_{\Delta_1,\Delta_2}\mathbf{x}_*| \leq (1 - \epsilon P_{\Delta_1,\Delta_2})|\mathbf{x}| + \epsilon P_{\Delta_1,\Delta_2}|\mathbf{x}_*| \\ &\leq \max\{|\mathbf{x}|, |\mathbf{x}_*|\}. \end{aligned}$$

Furthermore, we can observe that, as in [36], the binary interactions (I.7) are such that

$$\begin{aligned} \langle \mathbf{x}' + \mathbf{x}'_* \rangle &= \mathbf{x} + \mathbf{x}_* + \epsilon \left(P_{\Delta_1,\Delta_2}(\mathbf{x}, \mathbf{x}_*, c, c_*) - P_{\Delta_1,\Delta_2}(\mathbf{x}_*, \mathbf{x}, c_*, c) \right)(\mathbf{x}_* - \mathbf{x}) \\ &= \mathbf{x} + \mathbf{x}_*, \end{aligned} \tag{I.9}$$

meaning that, since the interaction function P_{Δ_1,Δ_2} is symmetric, the mean is conserved on average in a single binary interaction.

In general, for the introduced interaction function (I.3) the energy is not conserved. A particularly simple case may be obtained by considering P_{Δ_1,Δ_2} constant. In this case, from (I.9) we deduce that the mean position is conserved in each binary interaction, whereas if $\sigma^2 = 0$, the mean energy is dissipated

Let us introduce the distribution function $f = f(\mathbf{x}, c, t) : \mathbb{R}^2 \times [0,1] \times \mathbb{R}_+ \to \mathbb{R}_+$ such that $f(\mathbf{x}, c, t)d\mathbf{x}dc$ is the fraction of particles which at time $t \geq 0$ are represented by their position in $\mathbf{x} \in \mathbb{R}^2$ and feature $c \in [0,1]$. The evolution

of f undergoing binary interactions (I.7) can be described in the following Boltzmann-type kinetic equation

$$\begin{aligned}&\partial_t f(\mathbf{x}, c, t)\\ &\quad = \left\langle \int_0^1 \int_{\mathbb{R}^2} \frac{1}{{}'J} (f({}'\mathbf{x}, {}'c, t) f({}'\mathbf{x}_*, {}'c_*, t) - f(\mathbf{x}, c, t) f(\mathbf{x}_*, c_*, t)) d\mathbf{x}_* dc_* \right\rangle,\end{aligned} \tag{I.10}$$

where $({}'\mathbf{x}, {}'\mathbf{x}_*)$ are the pre-interaction positions which generate the postinteraction positions $(\mathbf{x}, \mathbf{x}_*)$ according to the interaction rule (I.7). Similarly, we denoted with $({}'c, {}'c_*)$ the preinteraction features generating the postinteraction features (c, c_*). Anyway, following the scheme (I.7), the dynamics do not prescribe evolution of the features. Finally, we denoted by ${}'J$ the Jacobian of the transformation $({}'\mathbf{x}, {}'\mathbf{x}_*) \to (\mathbf{x}, \mathbf{x}_*)$.

Equation (I.10) may be recast in weak form as follows

$$\begin{aligned}&\frac{d}{dt} \int_{[0,1]} \int_{\mathbb{R}^2} \phi(\mathbf{x}, c) f(\mathbf{x}, c, t) d\mathbf{x} dc\\ &\quad = \int_{[0,1]^2} \int_{\mathbb{R}^4} \langle \phi(\mathbf{x}', c') - \phi(\mathbf{x}, c) \rangle \, f(\mathbf{x}, c, t) f(\mathbf{x}_*, c_*, t) d\mathbf{x} dc d\mathbf{x}_* dc_*,\end{aligned} \tag{I.11}$$

where $\phi(\mathbf{x}, c) : \mathbb{R}^2 \times [0,1] \to \mathbb{R}$ is a test function. Choosing $\phi(\mathbf{x}, c) = 1$, we get that the total mass of f is constant in time, meaning that the total number of particles/pixels are conserved. Choosing instead $\phi(\mathbf{x}, c) = \mathbf{x}$ we get

$$\begin{aligned}&\frac{d}{dt} \int_0^1 \int_{\mathbb{R}^2} \mathbf{x} f(\mathbf{x}, c, t) d\mathbf{x} dc\\ &\quad = \epsilon \int_{[0,1]^2} \int_{\mathbb{R}^4} (P_{\Delta_1, \Delta_2}(\mathbf{x}, \mathbf{x}_*, c, c_*) - P_{\Delta_1, \Delta_2}(\mathbf{x}_*, \mathbf{x}, c_*, c))(\mathbf{x}_* - \mathbf{x})\\ &\qquad \times f(\mathbf{x}, c, t) f(\mathbf{x}_*, c_*, t) d\mathbf{x} d\mathbf{x}_* dc dc_*.\end{aligned}$$

Therefore, the mean position $M = \int_0^1 \int_{\mathbb{R}^2} \mathbf{x} f(\mathbf{x}, c, t) d\mathbf{x} dc$ is conserved in time being P_{Δ_1, Δ_2} symmetric.

In the following, we introduce a suitable scaling under which we can obtain the nonlocal mean-field model (I.5) starting from the binary interaction scheme (I.7). The procedure is based on the quasi-invariant regime introduced in [36]. We introduce the new time scale $\tau = \epsilon t$, we scale the distribution function $g(\mathbf{x}, c, \tau) = f(\mathbf{x}, c, \tau/\epsilon)$, and we introduce the following scaling for the variance

$$\sigma^2 \to \epsilon \sigma^2. \tag{I.12}$$

We observe that $\partial_\tau g = \frac{1}{\epsilon} \partial_t f$ and the equation satisfied by g is

$$\begin{aligned}&\frac{d}{d\tau} \int_0^1 \int_{\mathbb{R}^2} \phi(\mathbf{x}, c) g(\mathbf{x}, c, \tau) d\mathbf{x} dc\\ &\quad = \frac{1}{\epsilon} \int_{[0,1]^2} \int_{\mathbb{R}^4} \langle \phi(\mathbf{x}', c') - \phi(\mathbf{x}, c) \rangle \, g(\mathbf{x}, c, \tau) g(\mathbf{x}_*, c_*, \tau) d\mathbf{x} d\mathbf{x}_* dc dc_*.\end{aligned} \tag{I.13}$$

Hence, if $\epsilon \ll 1$ and if ϕ is sufficiently smooth then the difference $\langle \phi(\mathbf{x}', c') - \phi(\mathbf{x}, c) \rangle$ is small and can be expanded in Taylor series. We obtain

$$\langle \phi(\mathbf{x}', c') - \phi(\mathbf{x}, c) \rangle = \langle \mathbf{x}' - \mathbf{x} \rangle \cdot \nabla_{\mathbf{x}} \phi(\mathbf{x}, c) \tag{I.14}$$

$$+ \frac{1}{2} \langle (\mathbf{x}' - \mathbf{x})^T H[\phi](\mathbf{x}' - \mathbf{x}) \rangle + R_\epsilon(\mathbf{x}, \mathbf{x}_*, c), \tag{I.15}$$

where $R_\epsilon(\mathbf{x}, \mathbf{x}_*, c)$ is the remainder term of the Taylor expansion and $H[\phi]$ is the Hessian matrix. Within the scaling (I.12) we highlight that, by construction, the remainder term R_ϵ depends in a multiplicative way on higher moments of the random variable η su that $R_\epsilon/\epsilon \ll 1$ for $\epsilon \ll 1$. We point the interested reader to [47], [48] for further details. We note that all the terms multiplied by $c' - c$ do not appear in expression (I.14) since $c' = c$.

Hence, by substituting (I.14) into equation (I.13) we have

$$\begin{aligned}
&\frac{d}{d\tau} \int_0^1 \int_{\mathbb{R}^2} \phi(\mathbf{x}, c) g(\mathbf{x}, c, \tau) d\mathbf{x} dc \\
&= \frac{1}{\epsilon} \int_{[0,1]^2} \int_{\mathbb{R}^4} \langle \mathbf{x}' - \mathbf{x} \rangle \cdot \nabla_{\mathbf{x}} \phi(\mathbf{x}, c) g(\mathbf{x}, c, t) g(\mathbf{x}_*, c_*, \tau) d\mathbf{x} d\mathbf{x}_* dc dc_* \\
&\quad + \frac{1}{2\epsilon} \int_{[0,1]^2} \int_{\mathbb{R}^4} \langle (\mathbf{x}' - \mathbf{x})^T H[\phi(\mathbf{x}, c)](\mathbf{x}' - \mathbf{x}) \rangle g(\mathbf{x}, c, \tau) g(\mathbf{x}_*, c_*, \tau) d\mathbf{x} dc d\mathbf{x}_* dc_* \\
&\quad + \frac{1}{\epsilon} \int_{[0,1]^2} \int_{\mathbb{R}^4} R_\epsilon(\mathbf{x}, \mathbf{x}_*, c) \phi(\mathbf{x}, c) g(\mathbf{x}, c, \tau) g(\mathbf{x}_*, c_*, \tau) d\mathbf{x} d\mathbf{x}_* dc dc_*
\end{aligned} \tag{I.16}$$

In the limit $\epsilon \to 0^+$ we formally obtain:

$$\begin{aligned}
&\frac{d}{d\tau} \int_0^1 \int_{\mathbb{R}^2} \phi(\mathbf{x}, c) g(\mathbf{x}, c, \tau) d\mathbf{x} dc \\
&= \int_{[0,1]^2} \int_{\mathbb{R}^4} P_{\Delta_1, \Delta_2}(\mathbf{x}, \mathbf{x}_*, c, c_*)(\mathbf{x}_* - \mathbf{x}) \cdot \nabla_{\mathbf{x}} \phi(\mathbf{x}, c) g(\mathbf{x}, c, t) g(\mathbf{x}_*, c_*, \tau) d\mathbf{x} dc d\mathbf{x}_* dc_* \\
&\quad + \sigma^2 \int_{[0,1]^2} \int_{\mathbb{R}^4} D(c) \nabla_{\mathbf{x}}^2 \phi(\mathbf{x}, c) g(\mathbf{x}, c, \tau) g(\mathbf{x}_*, c_*, \tau) d\mathbf{x} d\mathbf{x}_* dc dc_*.
\end{aligned} \tag{I.17}$$

Next, we may integrate back by parts and restoring the original variables we get the following Fokker-Planck-type equation

$$\partial_t g(\mathbf{x}, c, \tau) = \nabla_{\mathbf{x}} \cdot \left[\mathscr{B}_{\Delta_1, \Delta_2}[g](\mathbf{x}, c, t) g(\mathbf{x}, c, \tau) + \sigma^2 D(c) \nabla_{\mathbf{x}} g(\mathbf{x}, c, \tau) \right], \tag{I.18}$$

provided the following boundary condition is satisfied

$$\mathscr{B}_{\Delta_1, \Delta_2}[g](\mathbf{x}, c, \tau) g(\mathbf{x}, c, \tau) + \sigma^2 D(c) \nabla_{\mathbf{x}} g(\mathbf{x}, c, \tau) \Big|_{|\mathbf{x}| \to +\infty} = 0$$

In (I.18) the nonlocal operator $\mathscr{B}_{\Delta_1, \Delta_2}[\cdot]$ corresponds to the one defined in (I.6). We may observe how the obtained model is equivalent to the Fokker Planck equation (I.5) in absence of the diffusion coefficient, i.e. in the case $\sigma^2 = 0$.

I.3.2 DSMC method for Boltzmann-type equations

In this section, we introduce a direct simulation Monte Carlo (DSMC) approach to Boltzmann-type equations. The numerical approximation of nonlinear Boltzmann-type models is a major task and has been deeply investigated in the recent decades, see e.g. [15], [49]–[51]. The main issue of deterministic methods relies on the so-called curse of dimensionality affecting the approximation of the multidimensional integral of the collision operator. Furthermore, the preservation of relevant physical quantities is challenging at the deterministic level, making the schemes model dependent. On the other hand Monte Carlo methods for kinetic equations naturally employ the microscopic dynamics defining the binary scheme to satisfy the physical constraints and are much less sensitive to the curse of dimensionality. Among the most popular examples of Monte Carlo methods for the Boltzmann equation we mention the DSMC method of Nanbu [52], Bird [53]. Rigorous results on the convergence of the methods have been provided in [54]. Furthermore, the computational cost of the method is $O(N)$, where N is the number of the particles of the system, making this method very efficient.

In the following, we describe the DSMC method based on the Nanbu-Babovsky scheme. This method consists in selecting particles by independent pairs and making them evolve at the same time according to the binary collision rules described in equation (I.7). Let us consider a time interval $[0, T]$ and discretize it in N_t intervals of size Δt. We introduce the stochastic rounding of a positive real number x as

$$\mathrm{Sround}(x) = \begin{cases} \lfloor x \rfloor + 1 & \text{with probability} \quad x - \lfloor x \rfloor \\ \lfloor x \rfloor & \text{with probability} \quad 1 - x + \lfloor x \rfloor \end{cases},$$

where $\lfloor x \rfloor$ denotes the integer part of x. We sample the random variable η from a 2D Gaussian distribution with mean equal to zero and diagonal covariance matrix.

The kinetic distribution, as well as its moments, is then recovered from the empirical density function

$$f_N(\mathbf{x}, c, t) = \frac{1}{N} \sum_{i=1}^{N} \delta(\mathbf{x} - \mathbf{x}_i(t)) \otimes \delta(c - c_i(t)),$$

where $\delta(\cdot)$ is the Dirac delta function. Hence, for any test function ϕ we denote the moments of the distribution f by

$$(\phi, f)(t) = \int_0^1 \int_{\mathbb{R}^2} \phi(\mathbf{x}, c) f(\mathbf{x}, c, t) d\mathbf{x} dc,$$

we have

$$(\phi, f_N)(t) = \frac{1}{N} \sum_{i=1}^{N} \phi(\mathbf{x}_i, c_i).$$

Algorithm 1 Monte Carlo algorithm for Boltzmann equation (I.11)

1: Given N particles $(\mathbf{x}_n^0, c_n^0)$, with $n = 1, \dots, N$ computed from the initial distribution $f_0(\mathbf{x}, c)$;
2: **for** $t = 1$ **to** T **do**
3: set $n_p = \text{Sround}(N/2)$;
4: sample n_p pairs (i, j) uniformly without repetition among all possible pairs of particles at time step t;
5: for each pair (i, j), sample η
6: for each pair (i, j), compute the data change:

$$\begin{aligned} \Delta\mathbf{x}_i^t &= \epsilon P_{\Delta_1,\Delta_2}(\mathbf{x}_i^t, \mathbf{x}_j^t, c_i^0, c_j^0)(\mathbf{x}_j^t - \mathbf{x}_i^t) + \sqrt{2\sigma^2 D(c_i^0)}\boldsymbol{\eta} \\ \Delta\mathbf{x}_j^t &= \epsilon P_{\Delta_1,\Delta_2}(\mathbf{x}_j^t, \mathbf{x}_i^t, c_j^0, c_i^0)(\mathbf{x}_i^t - \mathbf{x}_j^t) + \sqrt{2\sigma^2 D(c_j^0)}\boldsymbol{\eta} \end{aligned} \tag{I.19}$$

compute

$$\mathbf{x}_{i,j}^{t+1} = \mathbf{x}_{i,j}^t + \Delta\mathbf{x}_{i,j}^t \tag{I.20}$$

7: **end for**

In the following, we will evaluate the approximation of the empirical density f_N obtained by

$$f_{N,\Delta\mathbf{x},\Delta c}(\mathbf{x}, c, t) = \frac{1}{N}\sum_{i=1}^{N} S_{\Delta\mathbf{x}}(\mathbf{x} - \mathbf{x}_i(t)) \otimes S_{\Delta c}(c - c_i(t)), \tag{I.21}$$

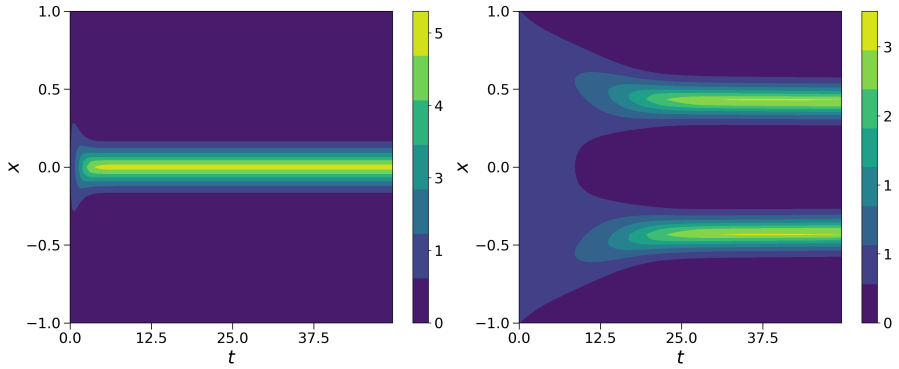
with $S_{\Delta\mathbf{x}}(\mathbf{x})$, $S_{\Delta c}(c)$ suitable mollifications of the indicator function. In the simplest setting, if we consider the indicator function it would lead to the standard histogram.

I.3.3 Numerical examples for the Hegselmann-Krause dynamics

In this section we provide numerical evidence of the consistency of the Boltzmann-type approach with respect to the one introduced in (I.18) corresponding, in the zero-diffusion limit to the model (I.5). In particular, the numerical solution of the Fokker Planck equation (I.18) has been obtained through a structure preserving deterministic scheme [55].This class of schemes are capable of approximating with arbitrary order of accuracy the steady state of a Fokker-Planck-type model and preserve important physical properties like positivity and entropy dissipation. The numerical approximation of the Boltzmann model has been obtained through the Algorithm 1.

We consider in particular $N = 10^5$ particles uniformly distributed in the domain $[-1, 1]^2 \times [0, 1] \subset \mathbb{R}^2 \times [0, 1]$

$$f(\mathbf{x}, c, 0) = \begin{cases} \beta & (\mathbf{x}, c) \in [-1, 1]^2 \times [0, 1] \\ 0 & \text{elsewhere} \end{cases} \tag{I.22}$$



(a) (b)

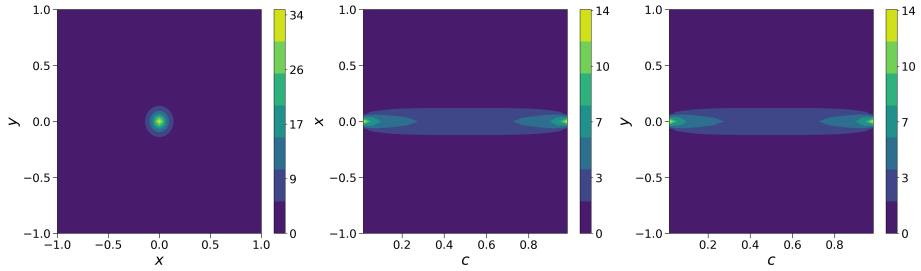
Figure I.2: Transient solutions of the Fokker-Planck equation (I.18) approximated through a semi-implicit SP scheme over $[0,T]$, $T = 50$, with time step $\Delta t = 3 \cdot 10^{-1}$. In the left panel we consider $\Delta_1 = 2$, $\Delta_2 = 1$ and $\sigma^2 = 5 \cdot 10^{-2}$ while in the right panel $\Delta_1 = 0.5$, $\Delta_2 = 1$ and $\sigma^2 = 10^{-2}$. The initial distribution is (I.22).

with $\beta > 0$ a normalization constant such that $\int_0^1 \int_{\mathbb{R}^2} f(\mathbf{x}, c, 0) d\mathbf{x} dc = 1$. Hence, we compare solution of the Boltzmann-type model (I.10) under the scalings $\epsilon = 10^{-1}$ and $\epsilon = 10^{-2}$ with the solution of the Fokker-Planck model (I.18). We consider the local diffusion function $D(c)$ of the form $D(c) = c(1-c)$, $c \in [0,1]$.

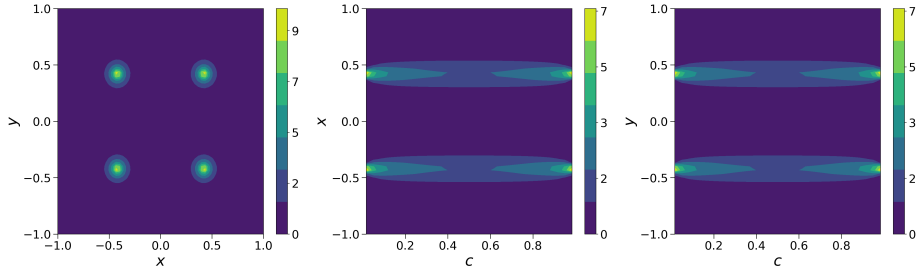
Let us introduce the spatial grid $x_i \in [-L, L]$, $y_j \in [-L, L]$, $c_k \in [0,1]$ such that $x_{i+1} - x_i = \Delta x$, $y_{j+1} - y_j = \Delta y$, $c_{k+1} - c_k = \Delta c$. We further assume $\Delta x = \Delta y$ such that $L = N_x \Delta x = N_y \Delta y$ and $N_x = N_y = 61$, the discretization of the features' interval $[0,1]$ is instead computed with $N_c = 31$ gridpoints. We use a second order semi-implicit method for the time integration of (I.18), we refer to [55] for a detailed description of these methods.

In Figure III.4 we show the transient behaviour of the Fokker-Planck solutions obtained with the semi-implicit SP scheme up in the time interval $[0, 50]$ with time step $\Delta t = 3 \cdot 10^{-1}$ and we fix $\sigma^2 = 5 \cdot 10^{-2}$ for the left panel and $\sigma^2 = 10^{-2}$ for the right panel. We represent the projection of the kinetic density along the x-axis computed as $f_{yc}(x,t) = \int_{-L}^{L} \int_0^1 f(x,y,c,t) dc dy$. In the left panel we fix the confidence coefficients $\Delta_1 = 2$ and $\Delta_2 = 1$ and in the right panel we choose $\Delta_1 = 0.5$ and $\Delta_2 = 1$. We may observe how the qualitative behavior of the solution dramatically changes since multiple clusters appear for large times, see [38].

Similarly, in Figure III.5 we show the projections on the xy, xc and yc planes of the numerical solution of the introduced FP model at time $T = 50$. The projections are computed as $f_c = \int_0^1 f(\mathbf{x}, c, T) dc$, $f_y = \int_{-L}^{L} f(\mathbf{x}, c, T) dy$, $f_x = \int_{-L}^{L} f(\mathbf{x}, c, T) dx$. We may observe how, for large times, the kinetic density displays multiple clusters both the space variable $\mathbf{x}$ and in the features' variable



(a)



(b)

Figure I.3: Asymptotic solutions of the Fokker-Plank equation numerically computed with the SP scheme at the final time $T = 50$. The left column represents the xy-projections, the middle column the xc-projections and the right column the yc-projections. In the top row we use $\Delta_1 = 2$, $\Delta_2 = 1$ and $\sigma^2 = 5 \cdot 10^{-2}$ while in the bottom row $\Delta_1 = 0.5$, $\Delta_2 = 1$ and $\sigma^2 = 10^{-2}$.

c. The parameters Δ_1, Δ_2 and σ^2 are the same used in III.4a and III.4b for the top and bottom panels of Figure III.5 respectively.

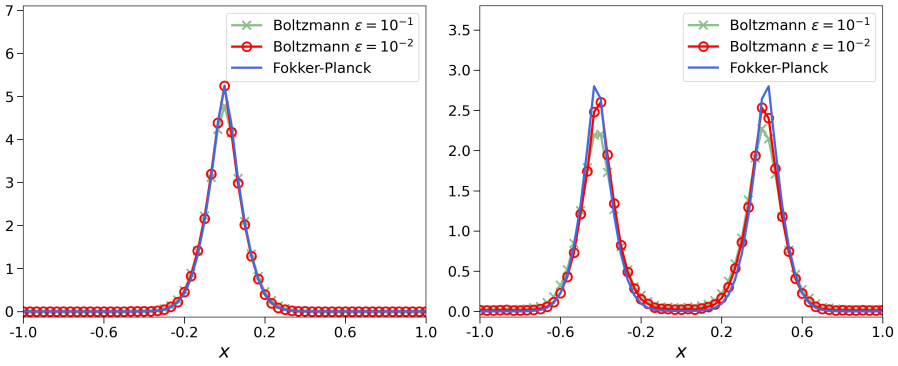
In Figure I.4 we show the x-axis projection of the Fokker-Planck solution provided by the SP scheme and the solutions of the Boltzmann equation obtained with Algorithm 1 with $\epsilon = 10^{-2}, 10^{-1}$. We can also observe that, as expected, the solution of the Boltzmann model is a good approximation of the solution of the Fokker-Planck model for small values of the parameter $\epsilon > 0$.

I.4 Application to biomedical images

In this section we focus on segmentation problems for medical images. In particular, we concentrate on the segmentation of images of cell nuclei, of brain tumour and on the recognition of thigh muscles. In all the aforementioned cases, we apply Algorithm 1 to generate the segmentation masks.

The procedure to generate segmentation masks can be summarized as follows:

i) The pixels of the 2D image are interpreted as uniformly spaced particles, each characterized by the spatial position (x_i, y_i) and with static feature



(a) (b)

Figure I.4: Comparison between the numeric solution of the Fokker-Planck equation computed by the SP scheme (in blue) with final distribution provided by the MC algorithm for the Boltzmann-type equation with two different values of the parameter ϵ (in red and green). Both the panels represent the x-projections of the asymptotic distributions computed for $T = 50$. In the left panel we use $\Delta_1 = 2$, $\Delta_2 = 1$ and $\sigma^2 = 5 \cdot 10^{-2}$ while in the right panel $\Delta_1 = 0.5$, $\Delta_2 = 1$ and $\sigma^2 = 10^{-2}$. The green distribution is computed with $\epsilon = 10^{-1}$ and the red one with $\epsilon = 10^{-2}$.

c_i defined are the gray level of the corresponding pixel. Hence, the initial distribution is reconstructed through (I.21). In all the applications of this work we linearly scale the initial position of the particles on a reference domain $[-1, 1] \times [-1, 1]$ and the initial values of the features to the $[0, 1]$ range.

ii) We numerically determine the large time solution of the Boltzmann-type model in (I.10) by means of the DSMC Algorithm 1. In this way, particles tend to aggregate in a number of finite clusters based on the Euclidean distance and on the difference between the gray levels of the pixels quantifying the features.

iii) The segmentation masks are generated by computing the mean value of the features of pixels in the same cluster. We assign to these values to the initial position of the pixels. Using this method, homogeneous regions with similar characteristics are created in the image and correspond to the segmentation mask.

iv) The obtained multi-level mask can be transformed into a binary mask by fixing a threshold $\tilde{c}$ such that if $c_i < \tilde{c}$ then $c_i = 0$ and if $c_i \geq \tilde{c}$ then $c_i = 1$.

Once the segmentation mask has been computed by this procedure, we apply two morphological refinement steps in order to remove those small regions misclassified as foreground parts and to fill possible small holes that were

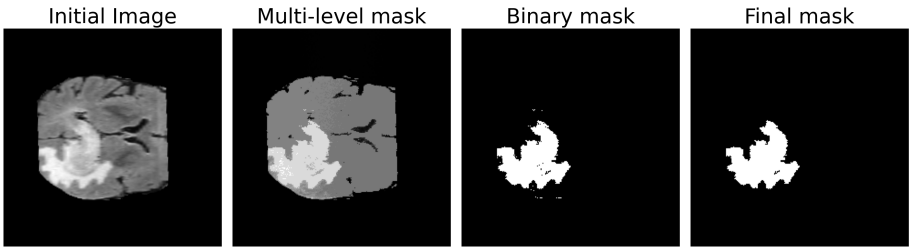


Figure I.5: Summary of the segmentation process. The first panel shows the input image; the second panel displays the multi-level segmentation mask produced by DSMC Algorithm 1; the third panel represents the mask after the binarization process; and the fourth panel shows the binary mask after the two morphological refinement steps.

misidentified background parts. Specifically, for the first step we label all the connected components of the foreground mask and remove the components whose number of pixels is below a fixed threshold. The second stage of morphological refinement works in the same way as the first, except that it is performed in a complementary manner on the background pixels. To implement these operations, we used the scikit-image python library [56] which exploits the graph theory to find distinct objects of a binary image [57]. By using this approach, we were able to obtain more accurate segmentation masks by reducing some small imperfections of the segmentation process. The introduced segmentation procedure and the results of the two morphological steps are sketched in Figure I.12.

I.4.1 Parameter identification

The purpose of this phase of the work is to choose the set of the optimal parameters $\Delta_1 > 0$, $\Delta_2 > 0$ and $\sigma^2 > 0$ for the segmentation problem based on the available data. We perform the optimization procedure by systematically scanning the parameter search space through a random sampling process and measuring the segmentation performance to find the best combination of Δ_1, Δ_2 and σ^2.

For image segmentation problems, the objective function quantifies the goodness of segmentation by measuring the distance or similarity between the ground truth segmentation mask and the evaluated mask. Let us consider as objective function the Dice Similarity Coefficient (DSC_{loss}), which is defined as follows

$$\begin{aligned} DSC_{loss} &= 1 - DSC_{metric} \\ &= 1 - \frac{2|S_{true} \cap S_{est}|}{|S_{true}| + |S_{est}|}, \end{aligned} \tag{I.23}$$

where S_{true} is the true mask and S_{est} is the estimated mask and the DSC_{loss} is computed only on the foreground (white voxels). This index is null if there is a

perfect overlap between the two masks and it is equal to one if the masks are completely disjoint [58]. It follows that the best segmentation will minimize this index. We refer to [59] for a complete overview of segmentation metrics.

Hence, we solve the optimization problem

$$\min_{\Delta_1,\Delta_2,\sigma^2>0} DSC_{loss} \tag{I.24}$$

where the loss function is computed by looking at the numerical equilibrium of the model (I.10) associated to a particular choice of parameters and obtained through Algorithm 1.

To solve the optimization problem Equation (I.24), we used the random sampling algorithm from the Hyperopt python package [60]. The key idea of this class of algorithms is to define a range of potential values for parameters of interest and randomly selecting a number of combinations to test. The performance of the objective function is evaluated for each point tested of the search space and the set of parameters that minimize the objective function represents the best configuration for solving the optimization problem. This technique can be useful for complex functions with non-linear, non-convex, or noisy shapes where gradient-based optimization may be difficult to apply [61]. To determine the bounds for Δ_1, Δ_2 and σ^2, we performed the optimization on a sufficiently large search space and repeated the process by progressively limiting the search space to the most promising area.

I.4.2 2D biomedical image segmentation

We apply the Monte Carlo Algorithm 1 and the optimization strategy described in section I.4.1 to 2D gray-scale biomedical image segmentation. We consider three different datasets:

- The *HL60 cell nuclei dataset* is a public dataset collected for the Cell Tracking Challenge and available at http://celltrackingchallenge.net/. It consists of synthetic 2D time-lapse video sequences of fluorescent stained HL60 nuclei moving on a substrate, realized with the Fluorescence Microscopy (FM) technique. Each image is accompanied with the ground truth segmentation mask which identifies the cell nuclei. To test our 2D segmentation pipeline we use the first time frame of the sequences.

- The *brain tumor dataset* consists 3D in multi-parametric magnetic resonance images (MRI) of patients affected by glioblastoma or lower-grade glioma, publicly available in the context of the Brain Tumor Image Segmentation (BraTS) Challenge http://medicaldecathlon.com/. The acquisition sequences include T_1-weighted, post-Gadolinium contrast T_1-weighted, T_2-weighted and T_2 Fluid-Attenuated Inversion Recovery (FLAIR) volumes. Three intra-tumoral structures were manually annotated by experienced radiologists, namely "tumor core", "enhancing tumor" and "whole tumor". We evaluate the performances of the MC algorithm for two different segmentation tasks: the "tumor core" and the "whole tumor"

annotations. For the first task we use a single slice in the axial plane of the post-Gadolinium contrast T_1-weighted scans while for the second task we use a single slice in the axial plane of the T_2-weighted scans.

- The *thigh muscles dataset* consists of 3D MRI scans of left and right thighs of healthy subjects and facioscapulohumeral dystrophy (FSHD) patients with muscle alterations. All the images were collected on a 3T MRI whole-body scanner (Skyra, Siemens Healthineers AG Erlangen, Germany) at the Mondino Foundation, Pavia Italy. Two acquisition protocols were performed for each subject: 3D six-point multi-echo gradient echo (GRE) sequence with interleaved echo sampling and a 2D multi-slice multi-echo spin echo (MESE) sequence. The dataset includes the ground truth segmentation mask of the 12 muscles of the thighs manually drawn by the radiologist team (we refer to [62] for a complete description of the acquisition settings and the annotation procedure). To test the segmentation pipeline we consider a single slice in the axial plane of the GRE scans of a healthy and an FSHD patient.

We optimize the values of the parameters Δ_1, Δ_2 and σ^2 by solving the minimization problem (I.24). For each dataset and segmentation task, we select the combination of the parameters that minimize the DSC_{loss}. The search space is defined by introducing additional constraints on the choice of parameters, in particular we consider $\Delta x \leq \Delta_1 \leq 0.7$, $0.05 \leq \Delta_2 \leq 0.3$. The values of the parameter σ^2 are determined by sampling the data from a log-uniform distribution with support $[e^{-5}, 1]$. The DSMC Algorithm 1 has been implemented in the quasi-invariant scaling with $\epsilon = 10^{-2}$ with final time $T_{max} = 2000$.

We present in Figure III.6 the results of the optimization process for the *HL60 cell nuclei dataset*. In particular, we show the values of the DSC_{loss} function as a function of Δ_1, Δ_2 and σ^2 parameters. The best configurations of the Δ_1, Δ_2 and σ^2 parameters for each dataset and segmentation task are listed in Table III.1.

Dataset name	$\boldsymbol{\Delta_1}$	$\boldsymbol{\Delta_2}$	$\boldsymbol{\sigma^2}$
HL60 cell nuclei	0.49	0.14	0.84
Brain tumor - tumor core	0.69	0.17	0.03
Brain tumor - whole tumor	0.34	0.28	0.25
Thigh muscles - healthy	0.55	0.13	0.02
Thigh muscles - FSHD	0.57	0.17	0.43

Table I.1: Best configurations of the Δ_1, Δ_2 and σ^2 parameters given by the optimization process.

Once the best configurations of the Δ_1, Δ_2 and σ^2 parameters have been established, we generate the segmentation masks for all the datasets and annotation tasks solving the Boltzmann model in (I.10) with the optimal choice

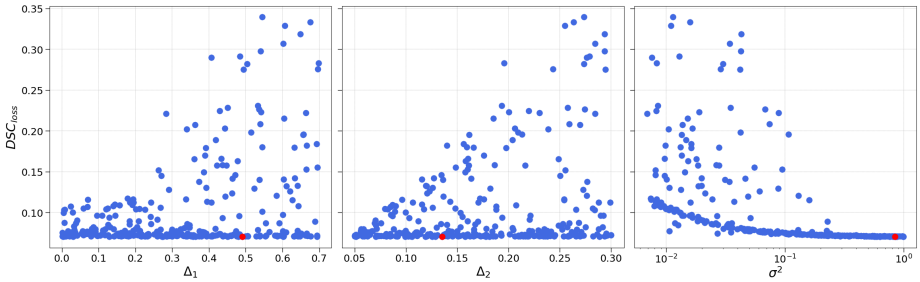


Figure I.6: Results of the optimization process for the *HL60 cell nuclei dataset* as a function of the three parameters Δ_1, Δ_2 and σ^2. We use scatter plots to graphically represent the distribution of the loss (DSC_{loss}) values as a function of the Δ_1, Δ_2 and σ^2 values. Red dots highlight the best configuration.

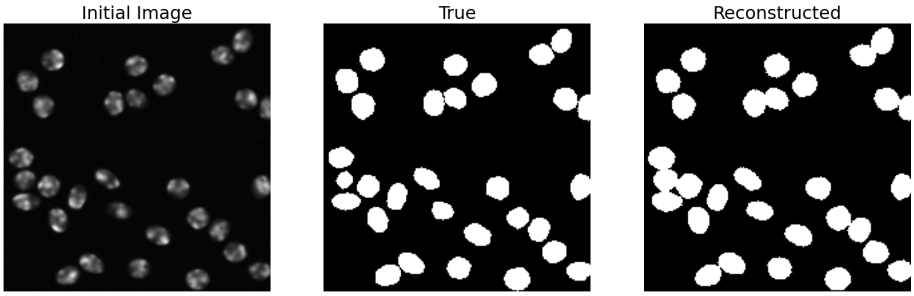


Figure I.7: Results of the segmentation process for the *HL60 cell nuclei dataset* obtained with the optimal set of Δ_1, Δ_2 and σ^2 parameters and with the morphological refinement steps. The three panels represent respectively from the left to the right the original image, the ground truth segmentation mask and the mask computed by the segmentation process.

of parameters. At the end of the segmentation procedure, we apply the two morphological refinement steps as described in Section I.4 to improve the quality of the segmentation masks.

In Figure III.8 we represent the results of the segmentation process obtained for the *HL60 cell nuclei dataset*. We can observe from the initial image that cells have regular borders but they are represented by a wide range of gray levels. By comparing the ground truth mask and the reconstructed one, we may observe that they are in good agreement. The method can provide good results in detecting the location of cells even for those regions whose gray level is close to the background. We evaluate the performance of the segmentation algorithm using the DSC_{metric} coefficient defined in Equation (I.23) which quantifies the overlap between the predicted and the true mask. The method reaches a DSC_{metric} of 0.94 on the image of the *HL60 cell nuclei dataset* reported in Figure III.8.

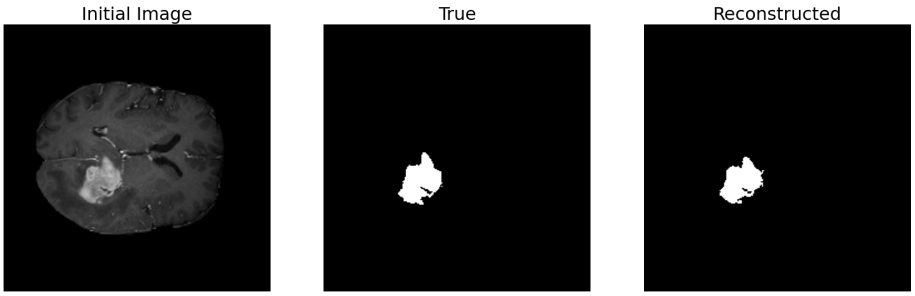


Figure I.8: Segmentation results for the "tumor core" task of the *brain tumor dataset*, obtained with the optimal set of Δ_1, Δ_2 and σ^2 parameters and with the morphological refinement steps.

In Figure III.9 we test the performance of the method in segmenting the "tumor core" region of one image of the *brain tumor dataset*. In this example, the region of interest is characterized by less regular edges, by the presence of cavities but by a quite homogeneous color. We can see from the Figure III.9 that the method identifies with a good precision the shape of the tumor and it preserves the small empty structures within the mass. In terms of the evaluation metric, the segmentation system achieves a DSC_{metric} of 0.93.

In Figure I.9 we present the second segmentation task performed on the *brain tumor dataset* where we apply the method to identify the "whole tumor" region. In this case, the region of interest is characterized by an irregular shape that also contains small holes, concave and convex areas. From the right panel of Figure I.9 we can observe that the reconstructed segmentation mask is very close to the reference one, except of some holes are not entirely identified as well as the small region in the upper area which is misclassified as a tumor part. However, the overall agreement of the two segmentations is good, also considering that the initial image is more complex than the previous ones. The evaluation metric DSC_{metric} reached by the model on the dataset is equal to 0.91.

In Figure I.10 we represent the results of the segmentation process for the *thigh muscles dataset* for a healthy subject I.10a and a FSHD patient I.10b. We can observe from the left panel that this dataset, compared to the previous ones, presents some features that appear more complex. Indeed, it is known how the thigh muscles are not well defined structures. Furthermore, thigh muscles often overlap each other or are separated by very thin segments of other tissues. The regions of interest are not uniform in color and differ slightly from the gray level of the surrounding areas. In particular, if we consider the MRI scan of the FSHD subject, we can observe how the disease has altered some muscles (three muscles of the lower area of the image) making them difficult to distinguish from the surrounding adipose tissues. Scan artifacts are stronger than in the other images, making the homogeneity of the gray levels poorer than in previous ones.

From the left panel of Figure I.10, we may observe that the method recognizes

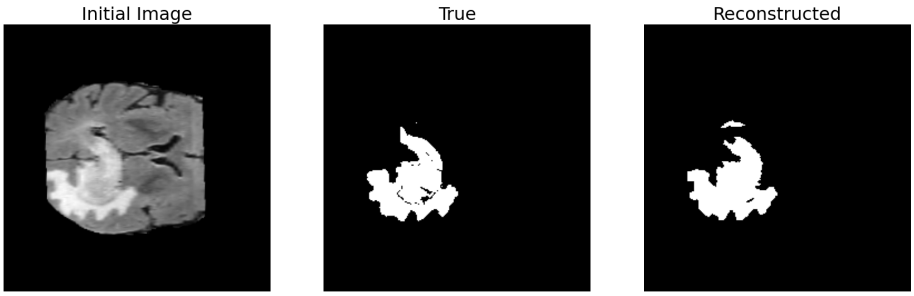


Figure I.9: Segmentation results for the "whole tumor" task of the *brain tumor dataset*, obtained with the optimal set of Δ_1, Δ_2 and σ^2 parameters and with the morphological refinement steps.

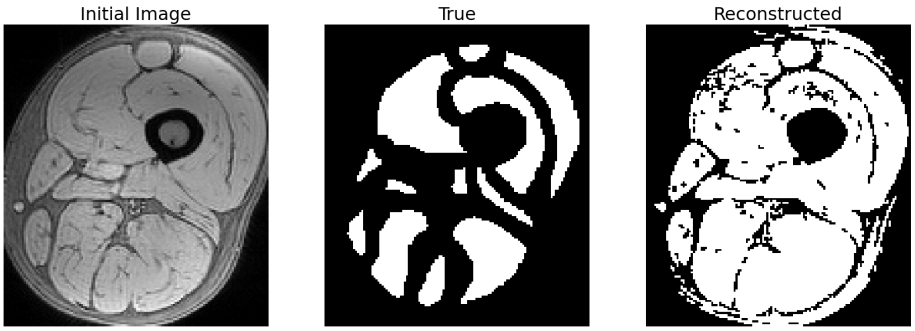
the thigh muscles by separating them from the other anatomical structures. However, from a visual assessment we notice that the segmentation precision achieved by the model on this dataset is lower than the previous ones. Especially in the peripheral areas, the method does not correctly distinguish the surrounding tissues from the muscles. If we consider the performances on the FSHD patient image, we can see how the method fails to recognize the muscles altered by the disease, misclassifying them as adipose tissue. In terms of the evaluation metric, the model reaches a DSC_{metric} equal to 0.60 on the healthy subject and a DSC_{metric} equal to 0.73 for the FSHD one, which are lower values respect the results obtained above.

It is interesting to observe that, by comparing the results obtained for the healthy and the FSHD patient, the optimal values of the confidence intervals Δ_1 and Δ_2 are compatible, while the best value of σ^2 coefficient is greater in the FSHD subject. This happens because the value of the σ^2 diffusion coefficient is directly related to the inhomogeneities of the gray levels of the pixels, which are greater in the image of the sick patient due to muscle damage. The increase in the diffusion parameter is particularly interesting as it could be used as an estimate of the FSHD muscle impairment and therefore an indicator of the progress of the disease.

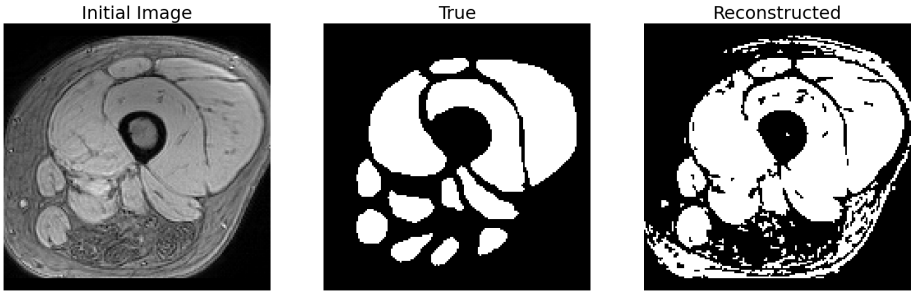
I.4.3 Patch-based 2D biomedical Image segmentation

To alleviate the limitations of the method encountered when dealing with more complex operations and lower quality image data, such as those in the *thigh muscles dataset*, in this section we present an improvement of the segmentation pipeline described in the previous paragraphs.

This new approach relies on the assumption that the method should recognize fine structures more accurately by focusing on smaller regions of the image. Therefore, we decide to apply the segmentation pipeline I.4 and the optimization algorithm I.4.1 to portions of the image that we will call patches (i.e. small



(a) Healthy subject



(b) FSHD subject

Figure I.10: Results of the segmentation process for the *thigh muscles dataset* obtained with the optimal set of Δ_1, Δ_2 and σ^2 parameters and with the morphological refinement steps.

subregions of the image defined as two-dimensional pixel arrays) rather than to the whole matrix of pixels [63], [64].

The method is composed of the following steps. Input images are first converted in square matrices by adding a padding of background pixels on the border of the input array. This step is necessary to obtain square patches, however the segmentation procedure could also be applied to rectangular ones. Images are then divided in non-overlapping patches that are passed as input of the optimization algorithm. In this way, the optimization process will find the best combination of the Δ_1, Δ_2 and σ^2 parameters for each single patch. Once the local optimization has been performed, the segmentation masks of each patch are estimated through the MC algorithm 1 with the best combination of parameters. The two refinement steps are applied to each patch masks in order to improve the quality of the results. Hence, all the patch masks are connected together to create the entire segmentation mask and the two post-processing routines are repeated over the complete mask to get the final segmentation.

We test the patch-based segmentation pipeline on the *thigh muscles dataset* for the healthy subject, which was the worst performing example. Using the padding process, we transform the 210×178 pixel array into a 216×216 pixel

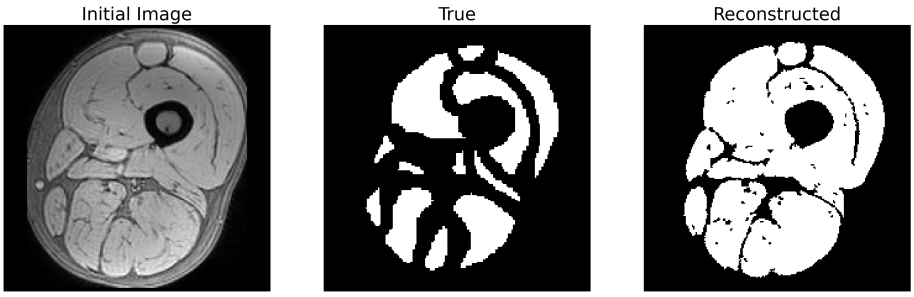


Figure I.11: Results of the segmentation process for the *thigh muscles dataset* obtained with the patch-based method with the optimal set of Δ_1, Δ_2 and σ^2 parameters and the morphological refinement steps.

array to fit four square patches to the width and height of the image. In this way, we generate 16 patches each of 54×54 pixels. Then, for each patch we solve the optimization problem (I.24) to determine the optimal parameters Δ_1, Δ_2 and σ^2 under the constraints $\Delta x \leq \Delta_1 \leq 1.6$, $0.05 \leq \Delta_2 \leq 0.2$. The values of σ^2 are sampled from a log-uniform distribution with support $[e^{-12}, 1]$.

The results of the patch-based segmentation algorithm for the *thigh muscles dataset* are represented in Figure I.11. We can observe that the reconstructed mask is clearly closer to the expected one, compared to the mask in Figure I.10. The method recognizes the thigh muscles with more precision and distinguishes them from the surrounding tissue more accurately. Particularly, if we consider the tissues in the peripheral area of the thigh that were wrongly classified as muscle tissues in Figure I.10, they are now correctly excluded from the muscles region. In terms of the evaluation metric, the segmentation system achieves a DSC_{metric} of 0.67. This result is satisfactory considering the complexity of the image in terms of anatomical regions and scan quality.

I.4.4 Comparison of different diffusion functions

In this section we compare the segmentation results obtained with two different local diffusion functions $D(c)$. In details, we consider the functions $D_1(c) > 0$ and $D_2(c) > 0$ defined as follows

$$D_1(c) = c(1-c), \qquad D_2(c) = \begin{cases} \alpha c & c \leq 0.5 \\ -\alpha c + \alpha & c > 0.5 \end{cases} \tag{I.25}$$

with $c \in [0,1]$ and $\alpha = \frac{1}{2}$ such that $D_1\left(\frac{1}{2}\right) = D_2\left(\frac{1}{2}\right)$, see Figure I.12.

Hence, we implemented the DSMC Algorithm 1 to segment the tumor core region of the *brain tumor dataset* using both $D_1(c)$ and $D_2(c)$. The parameters Δ_1, Δ_2 and σ^2 were optimized independently for both cases, and the best configurations are reported in Table I.2.

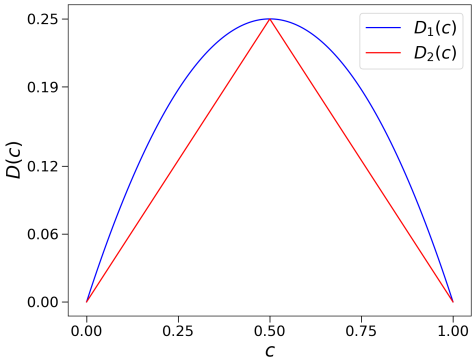


Figure I.12: Diffusion functions considered for image segmentation.

$\mathbf{D(c)}$	$\mathbf{\Delta_1}$	$\mathbf{\Delta_2}$	σ^2
D_1	0.69	0.17	0.03
D_2	0.58	0.27	0.06

Table I.2: Estimated values of Δ_1, Δ_2 and σ^2 by the optimization process for two different diffusion functions.

In Table I.3 we report the segmentation results in terms of DSC_{metric} obtained with local diffusion functions $D_1(c)$ and $D_2(c)$. We may observe that we obtained better performance of the segmentation pipeline defined with $D_1(c)$ with respect to the case $D_2(c)$. It is important to observe also that the difference in performance are alleviated by the morphological refinement step. Figure I.13 shows the segmentation results obtained with both diffusion functions. We may observe how the segmentation mask obtained with D_2 includes two small regions erroneously classified as part of the tumor area. These results suggest that the segmentation performance is influenced by the selection of a different diffusion function and, in this specific example, the quadratic function $D_1(c)$ leads to better results in the considered case.

However, the choice of the optimal diffusion function may be application-specific and be related to various factors such as the nature of the image, the size of the region of interest, and the desired level of detail in the segmentation mask.

Diffusion function	$\mathbf{DSC_{metric}}$ **Row mask**	$\mathbf{DSC_{metric}}$ **Final mask**
D_1	0.87	0.93
D_2	0.79	0.90

Table I.3: Values of DSC_{metric} before (row mask) and after (final mask) the morphological refinement steps.

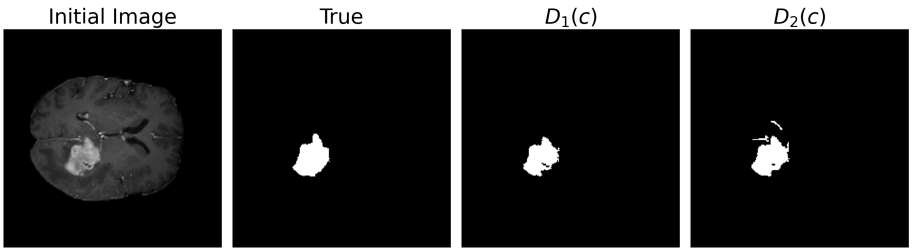


Figure I.13: Segmentation results for the "tumor core" task of the *brain tumor dataset*, obtained with the optimal set of Δ_1, Δ_2 and σ^2 parameters and with the two different diffusion functions D_1 and D_2.

Conclusions

In this work we proposed a kinetic model for consensus-based segmentation tasks inspired by the Hegselmann-Krause (HK) model. We derived a Boltzmann description of the generalized HK model based on a binary interaction scheme for the particles' locations and features. In the quasi-invariant limit we showed the consistency of the approach with existing mean-field modelling approaches by deriving a surrogate Fokker-Planck model with nonlocal drift. The Boltzmann-type description allows us to apply efficient direct simulation Monte Carlo schemes to evaluate the collision dynamics with low computational burden. Hence, we proposed an optimization strategy for the internal parameter configuration.

This model-based segmentation strategy is tested on three different biomedical datasets: a HL60 cell nuclei dataset, a brain tumor dataset and a thigh muscles dataset. The performances of the method are evaluated in terms of the DSC_{metric} that quantifies the overlap between the reconstructed mask and the reference mask. Good results are obtained for the HL60 cell nuclei dataset and the brain tumor dataset, while for the thigh muscle dataset the segmentation accuracy is lower. For this more complex dataset, we proposed a patch-based approach which consists of dividing the image into smaller arrays of pixels and applying the segmentation system to these subregions of the initial image. This second version of the method improves the quality of the segmentation mask.

Several extensions of the presented modelling approach to include color images and modified interaction functions are currently under study and will be discussed in future works.

Acknowledgements

This work has been written within the activities of the GNFM group of INdAM (National Institute of High Mathematics). MZ acknowledges partial support of MUR-PRIN2020 Project No.2020JLWP23 (Integrated Mathematical Approaches to Socio-Epidemiological Dynamics). The research of MZ was partially supported by MIUR, Dipartimenti di Eccellenza Program (2018-2022), and Department

of Mathematics “F. Casorati”, University of Pavia. R.F.C., S.F. and A.L thank the INFN-CSN5 research project *next_AIM (Artificial Intelligence in Medicine: next steps)*, https://www.pi.infn.it/aim/. One of us, A.L., thanks the national project “MIUR, Dipartimenti di Eccellenza Program (2018-2022), project F11I18000680001”.

References

[1] Cheng, H.-D., Jiang, X., Sun, Y., and Wang, J., “Color image segmentation: Advances and prospects,” *Pattern Recognition*, vol. 34, no. 12, pp. 2259–2281, 2001.

[2] Sharma, N. and Aggarwal, L. M., “Automated medical image segmentation techniques,” *Journal of Medical Physics*, vol. 35, no. 1, p. 3, 2010.

[3] Isensee, F., Jaeger, P. F., Kohl, S. A., Petersen, J., and Maier-Hein, K. H., “NnU-Net: A self-configuring method for deep learning-based biomedical image segmentation,” *Nature Methods*, vol. 18, no. 2, pp. 203–211, 2021.

[4] Ronneberger, O., Fischer, P., and Brox, T., “U-net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, Springer, 2015, pp. 234–241.

[5] Zhou, Z., Rahman Siddiquee, M. M., Tajbakhsh, N., and Liang, J., “Unet++: A nested U-net architecture for medical image segmentation,” in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*, Springer, 2018, pp. 3–11.

[6] Hesamian, M. H., Jia, W., He, X., and Kennedy, P., “Deep learning techniques for medical image segmentation: Achievements and challenges,” *Journal of Digital Imaging*, vol. 32, pp. 582–596, 2019.

[7] Liu, X., Song, L., Liu, S., and Zhang, Y., “A review of deep-learning-based medical image segmentation methods,” *Sustainability*, vol. 13, no. 3, p. 1224, 2021.

[8] Jain, A. K., Murty, M. N., and Flynn, P. J., “Data clustering: A review,” *ACM Computing Surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[9] Han, J., Yang, C., Zhou, X., and Gui, W., “A new multi-threshold image segmentation approach using state transition algorithm,” *Applied Mathematical Modelling*, vol. 44, pp. 588–601, 2017.

[10] Mittal, H., Pandey, A. C., Saraswat, M., Kumar, S., Pal, R., and Modwel, G., “A comprehensive survey of image segmentation: Clustering methods, performance parameters, and benchmark datasets,” *Multimedia Tools and Applications*, vol. 81, no. 24, pp. 35 001–35 026, 2022.

[11] Frigui, H. and Krishnapuram, R., "A robust competitive clustering algorithm with applications in computer vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 450–465, 1999.

[12] Yu, Z., Au, O. C., Zou, R., Yu, W., and Tian, J., "An adaptive unsupervised approach toward pixel clustering and color image segmentation," *Pattern Recognition*, vol. 43, no. 5, pp. 1889–1906, 2010.

[13] Pizzagalli, D. U., Gonzalez, S. F., and Krause, R., "A trainable clustering algorithm based on shortest paths from density peaks," *Science Advances*, vol. 5, no. 10, eaax3770, 2019.

[14] Herty, M., Pareschi, L., and Visconti, G., "Mean field models for large data–clustering problems," *Networks and Heterogeneous Media*, vol. 15, no. 3, pp. 463–487, 2020.

[15] Pareschi, L. and Toscani, G., *Interacting multiagent systems: kinetic equations and Monte Carlo msethods*. OUP Oxford, 2013.

[16] French Jr, J. R., "A formal theory of social power," *Psychological Review*, vol. 63, no. 3, p. 181, 1956.

[17] DeGroot, M. H., "Reaching a consensus," *Journal of the American Statistical Association*, vol. 69, no. 345, pp. 118–121, 1974.

[18] Chatterjee, S. and Seneta, E., "Towards consensus: Some convergence theorems on repeated averaging," *Journal of Applied Probability*, vol. 14, no. 1, pp. 89–97, 1977.

[19] Sznajd-Weron, K. and Sznajd, J., "Opinion evolution in closed community," *International Journal of Modern Physics C*, vol. 11, no. 06, pp. 1157–1165, 2000.

[20] Galam, S., "Rational group decision making: A random field Ising model at T=0," *Physica A: Statistical Mechanics and its Applications*, vol. 238, no. 1–4, pp. 66–80, 1997.

[21] Deffuant, G., Neau, D., Amblard, F., and Weisbuch, G., "Mixing beliefs among interacting agents," *Advances in Complex Systems*, vol. 3, no. 01n04, pp. 87–98, 2000.

[22] Ni, W. and Cheng, D., "Leader-following consensus of multi-agent systems under fixed and switching topologies," *Systems & Control Letters*, vol. 59, no. 3-4, pp. 209–217, 2010.

[23] Watts, D. J. and Dodds, P. S., "Influentials, networks, and public opinion formation," *Journal of consumer research*, vol. 34, no. 4, pp. 441–458, 2007.

[24] Motsch, S. and Tadmor, E., "Heterophilious dynamics enhances consensus," *SIAM Review*, vol. 56, no. 4, pp. 577–621, 2014.

[25] Castellano, C., Fortunato, S., and Loreto, V., "Statistical physics of social dynamics," *Reviews of Modern Physics*, vol. 81, no. 2, p. 591, 2009.

[26] Cucker, F. and Smale, S., "Emergent behavior in flocks," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 852–862, 2007.

[27] D'Orsogna, M. R., Chuang, Y.-L., Bertozzi, A. L., and Chayes, L. S., "Self-propelled particles with soft-core interactions: Patterns, stability, and collapse," *Physical Review Letters*, vol. 96, no. 10, p. 104 302, 2006.

[28] Degond, P., Liu, J.-G., and Ringhofer, C., "Evolution of the distribution of wealth in an economic environment driven by local Nash equilibria," *Journal of Statistical Physics*, vol. 154, pp. 751–780, 2014.

[29] Bouchaud, J.-P. and Mézard, M., "Wealth condensation in a simple model of economy," *Physica A: Statistical Mechanics and its Applications*, vol. 282, no. 3–4, pp. 536–545, 2000.

[30] Albi, G., Herty, M., and Pareschi, L., "Kinetic description of optimal control problems and applications to opinion consensus," *Communications in Mathematical Sciences*, vol. 13, no. 6, pp. 1407–1429, 2015.

[31] Albi, G., Pareschi, L., and Zanella, M., "Boltzmann-type control of opinion consensus through leaders," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 372, no. 2028, p. 20 140 138, 2014.

[32] Borra, D. and Lorenzi, T., "Asymptotic analysis of continuous opinion models under bounded confidence," *Communications on Pure & Applied Analysis*, vol. 12, no. 3, pp. 1487–1499, 2013.

[33] Düring, B. and Wolfram, M.-T., "Opinion dynamics: Inhomogeneous boltzmann-type equations modelling opinion leadership and political segregation," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 471, no. 2182, p. 20 150 345, 2015.

[34] Düring, B. and Wright, O., "On a kinetic opinion formation model for pre-election polling," *Philosophical Transactions of the Royal Society A*, vol. 380, no. 2224, p. 20 210 154, 2022.

[35] Piccoli, B., Duteil, N. P., and Trélat, E., "Sparse control of Hegselmann–Krause models: Black hole and declustering," *SIAM Journal on Control and Optimization*, vol. 57, no. 4, pp. 2628–2659, 2019.

[36] Toscani, G., "Kinetic models of opinion formation," *Communications in Mathematical Sciences*, vol. 4, no. 3, pp. 481–496, 2006.

[37] Colombo, R. M. and Garavello, M., "Hyperbolic consensus games," *Communications in Mathematical Sciences*, vol. 17, no. 4, pp. 1005–1024, 2019.

[38] Pareschi, L., Toscani, G., Tosin, A., and Zanella, M., "Hydrodynamic models of preference formation in multi-agent societies," *Journal of Nonlinear Science*, vol. 29, no. 6, pp. 2761–2796, 2019.

[39] Hegselmann, R. and Krause, U., "Opinion dynamics and bounded confidence models, analysis, and simulation," *Journal of Artifical Societies and Social Simulation (JASSS)*, vol. 5, no. 3, 2002.

[40] Lorenz, J., "Continuous opinion dynamics under bounded confidence: A survey," *International Journal of Modern Physics C*, vol. 18, no. 12, pp. 1819–1838, 2007.

[41] Blondel, V. D., Hendrickx, J. M., and Tsitsiklis, J. N., "Continuous-time average-preserving opinion dynamics with opinion-dependent communications," *SIAM Journal on Control and Optimization*, vol. 48, no. 8, pp. 5214–5240, 2010.

[42] Canuto, C., Fagnani, F., and Tilli, P., "An Eulerian approach to the analysis of Krause's consensus models," *SIAM Journal on Control and Optimization*, vol. 50, no. 1, pp. 243–265, 2012.

[43] Barbano, R., Arridge, S., Jin, B., and Tanno, R., "Uncertainty quantification in medical image synthesis," in *Biomedical Image Synthesis and Simulation: Methods and Applications*, ser. MICCAI Society Book Series, Academic Press, 2022, ch. 26, pp. 601–641.

[44] Kendall, A. and Gal, Y., "What uncertainties do we need in Bayesian deep learning for computer vision?" In *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, 2017.

[45] Kwon, Y., Won, J.-H., Kim, B. J., and Paik, M. C., "Uncertainty quantification using bayesian neural networks in classification: Application to biomedical image segmentation," *Computational Statistics & Data Analysis*, vol. 142, p. 106 816, 2020.

[46] Carrillo, J. A., Fornasier, M., Rosado, J., and Toscani, G., "Asymptotic flocking dynamics for the kinetic Cucker–Smale model," *SIAM Journal on Mathematical Analysis*, vol. 42, no. 1, pp. 218–236, 2010.

[47] Cordier, S., Pareschi, L., and Toscani, G., "On a kinetic model for a simple market economy," *Journal of Statistical Physics*, vol. 120, pp. 253–277, 2005.

[48] Pareschi, L. and Toscani, G., "Wealth distribution and collective knowledge: A Boltzmann approach," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 372, no. 2028, p. 20 130 396, 2014.

[49] Albi, G. and Pareschi, L., "Binary interaction algorithms for the simulation of flocking and swarming dynamics," *Multiscale Modeling & Simulation*, vol. 11, no. 1, pp. 1–29, 2013.

[50] Dimarco, G. and Pareschi, L., "Numerical methods for kinetic equations," *Acta Numerica*, vol. 23, pp. 369–520, 2014.

[51] Pareschi, L. and Russo, G., "An introduction to monte carlo method for the boltzmann equation," in *ESAIM: Proceedings*, EDP Sciences, vol. 10, 2001, pp. 35–75.

[52] Nanbu, K., "Direct simulation scheme derived from the Boltzmann equation. I. Monocomponent gases," *Journal of the Physical Society of Japan*, vol. 49, no. 5, pp. 2042–2049, 1980.

[53] Bird, G., "Direct simulation and the Boltzmann equation," *The Physics of Fluids*, vol. 13, no. 11, pp. 2676–2681, 1970.

[54] Babovsky, H. and Illner, R., "A convergence proof for Nanbu's simulation method for the full Boltzmann equation," *SIAM Journal on Numerical Analysis*, vol. 26, no. 1, pp. 45–65, 1989.

[55] Pareschi, L. and Zanella, M., "Structure preserving schemes for nonlinear Fokker–Planck equations and applications," *Journal of Scientific Computing*, vol. 74, no. 3, pp. 1575–1600, 2018.

[56] Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., *et al.*, "Scikit-image: Image processing in Python," *PeerJ*, vol. 2, e453, 2014.

[57] Fiorio, C. and Gustedt, J., "Two linear time union-find strategies for image processing," *Theoretical Computer Science*, vol. 154, no. 2, pp. 165–181, 1996.

[58] Yang, Y., Shu, X., Wang, R., Feng, C., and Jia, W., "Parallelizable and robust image segmentation model based on the shape prior information," *Applied Mathematical Modelling*, vol. 83, pp. 357–370, 2020.

[59] Taha, A. A. and Hanbury, A., "Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool," *BMC Medical Imaging*, vol. 15, no. 1, pp. 1–28, 2015.

[60] Bergstra, J., Komer, B., Eliasmith, C., Yamins, D., and Cox, D. D., "Hyperopt: A python library for model selection and hyperparameter optimization," *Computational Science & Discovery*, vol. 8, no. 1, p. 014 008, 2015.

[61] Andradóttir, S., "An overview of simulation optimization via random search," *Handbooks in operations research and management science*, vol. 13, pp. 617–631, 2006.

[62] Agosti, A., Shaqiri, E., Paoletti, M., *et al.*, "Deep learning for automatic segmentation of thigh and leg muscles," *Magnetic Resonance Materials in Physics, Biology and Medicine*, vol. 35, no. 3, pp. 467–483, 2022.

[63] Coupé, P., Manjón, J. V., Fonov, V., Pruessner, J., Robles, M., and Collins, D. L., "Patch-based segmentation using expert priors: Application to hippocampus and ventricle segmentation," *NeuroImage*, vol. 54, no. 2, pp. 940–954, 2011.

[64] Cordier, N., Delingette, H., and Ayache, N., "A patch-based approach for the segmentation of pathologies: Application to glioma labelling," *IEEE Transactions on Medical Imaging*, vol. 35, no. 4, pp. 1066–1076, 2015.

Paper II

Quantification of pulmonary involvement in COVID-19 pneumonia by means of a cascade of two U-nets: training and assessment on multiple datasets using different annotation criteria

Francesca Lizzi [1,2], Abramo Agosti [6], Francesca Brero [4,5], Raffaella Fiamma Cabini [4,6], Maria Evelina Fantacci [2,3], Silvia Figini [4,11], Alessandro Lascialfari [4,5], Francesco Laruina [1,2], Piernicola Oliva [8,9], Stefano Piffer [7,10], Ian Postuma [4], Lisa Rinaldi [4,5], Cinzia Talamonti [7,10], Alessandra Retico [2]

Published in *International Journal of Computer Assisted Radiology and Surgery*, October 2021, DOI: https://doi.org/10.1007/s11548-021-02501-2

Abstract

Purpose This study aims at exploiting Artificial intelligence (AI) for the identification, segmentation and quantification of COVID-19 pulmonary lesions. The limited data availability and the annotation quality are relevant factors in training AI-methods. We investigated the effects of using multiple datasets, heterogeneously populated and annotated according to different criteria.

[1] Scuola Normale Superiore, Pisa
[2] National Institute of Nuclear Physics (INFN), Pisa division, Pisa, IT
[3] Department of Physics, University of Pisa, Pisa, IT
[4] INFN, Pavia division, Pavia, IT
[5] Department of Physics, University of Pavia, Pavia, IT
[6] Department of Mathematics, University of Pavia, Pavia, IT
[7] Department of Biomedical Experimental Clinical Science "M. Serio", University of Florence, Florence, IT
[8] Department of Chemistry and Pharmacy, University of Sassari, Sassari, IT
[9] INFN, Cagliari division, Cagliari, IT
[10] INFN, Florence division, Florence, IT
[11] Department of Social and Political Science, University of Pavia, Pavia, IT

Methods We developed an automated analysis pipeline, the *LungQuant* system, based on a cascade of two U-nets. The first one (U-net$_1$) is devoted to the identification of the lung parenchyma, the second one (U-net$_2$) acts on a bounding box enclosing the segmented lungs to identify the areas affected by COVID-19 lesions. Different public datasets were used to train the U-nets and to evaluate their segmentation performances, which have been quantified in terms of the Dice Similarity Coefficients. The accuracy in predicting the CT-Severity Score (CT-SS) of the *LungQuant* system has been also evaluated.

Results Both the volumetric DSC (vDSC) and the accuracy showed a dependency on the annotation quality of the released data samples. On an independent dataset (COVID-19-CT-Seg), both the vDSC and the surface DSC (sDSC) were measured between the masks predicted by *LungQuant* system and the reference ones. The vDSC (sDSC) values of 0.95±0.01 and 0.66±0.13 (0.95±0.02 and 0.76±0.18, with 5 mm tolerance) were obtained for the segmentation of lungs and COVID-19 lesions, respectively. The system achieved an accuracy of 90% in CT-SS identification on this benchmark dataset.

Conclusion We analysed the impact of using data samples with different annotation criteria in training an AI-based quantification system for pulmonary involvement in COVID-19 pneumonia. In terms of vDSC measures, the U-net segmentation strongly depends on the quality of the lesion annotations. Nevertheless, the CT-SS can be accurately predicted on independent test sets, demonstrating the satisfactory generalization ability of the *LungQuant*.

Keywords: COVID-19, Chest Computed Tomography, Ground-glass opacities, Segmentation, Machine Learning, U-net.

II.1 Introduction

The task of segmenting the abnormalities of the lung parenchyma related to COVID-19 infection is a typical segmentation problem that can be addressed with methods based on Deep Learning (DL). CT findings of patients with COVID-19 infection may include bilateral distribution of ground-glass opacifications (GGO), consolidations, crazy-paving patterns, reversed halo sign and vascular enlargement [1]. Due to the extremely heterogeneous appearance of COVID-19 lesions in density, textural pattern, global shape and location in the lung, an analytical approach is definitely hard to code. The potential of DL-based segmentation approaches is particularly suited in this case, provided that a sufficient number of annotated examples are available for training the models. Few fully automated software tools devoted to this task have been recently proposed [2]–[4]. Lessmann *et al.* [2] developed a U-net model for lesion segmentation trained on semi-automatically annotated COVID-19 cases. The output of this system was then combined with the lung lobe segmentation algorithm reported in Xie *et al.* [5]. The approach proposed in Fang *et al.* [3] implements the automated lung segmentation method provided in the work of Hofmanninger *et al.* [6], together with a lesion segmentation strategy based on

Dataset name	Lung mask	GGO mask	CT-SS	N. of cases
Plethora [9]	Yes	No	No	402
Lung CT Segmentation Challenge [10]	Yes	No	No	60
COVID-19 Challenge [11]	No	Yes	No	199
MosMed [12]	No	No	No	1110
MosMed (annotated subsample)	No	Yes	Inferable	50
MosMed (in-house annotated subsample)	Yes	No	No	91
COVID-19-CT-Seg [4]	Yes	Yes	Inferable	10

Table II.1: A summary of the datasets used in this study. The CT Severity Score (CT-SS) information is not available for all datasets, but it can be computed for data which has both lung masks and ground-glass opacification (GGO) masks.

multiscale feature extraction [7]. The specific problem related to the development of fully automated DL-based segmentation strategies with limited annotated data samples has been explicitly tackled by Ma *et al.* [4]. The authors studied how to train and evaluate a DL-based system for lung and COVID-19 lesion segmentation on poorly populated samples of CT scans. They also made the data publicly available, allowing for a fair comparison with their system.

In this work we present a DL-based fully automated system to segment both lungs and lesions associated with COVID-19 pneumonia, the *LungQuant* system, which provides the part of lung volume compromised by the infection. We extended the study proposed by Ma *et al.* [4] focusing our efforts in investigating and discussing the impact of using different datasets and different labeling styles. Data can be highly variable in terms of acquisition protocols and machines when they are gathered from different sources. This poses a serious problem of dependence of the segmentation performances on the training sample characteristics. Despite advanced data harmonisation strategies could mitigate this problem [8], this approach is not applicable in absence of data acquisition information, as it is in this study for the available CT data. Nevertheless, DL methods, when trained with sufficiently large samples of heterogeneous data, can acquire the desired generalisation ability by themselves. In our analysis, we implemented an inter-sample cross-validation method to train, test and evaluate the generalisation ability of the *LungQuant* DL-based segmentation pipeline across the different available datasets. Finally, we also quantified the effect of using larger datasets to train, validate and test this kind of algorithm.

II.2 Material and Methods

II.2.1 Datasets

We used only publicly available datasets in order to make our results easily verifiable and reproducible. Five different datasets have been used to train and evaluate our segmentation pipeline. Most of them include image annotations, but each annotation has been associated with patients using different criteria. In Table II.1, a summary of available labels for each dataset is reported.

The lung segmentation problem has been tackled using a wide representation of the population and three different datasets: the Plethora, the Lung CT Segmentation Challenge and a subset of the MosMed dataset. On the other hand, the number of samples that are publicly available for COVID-19 infection segmentation may not be sufficient to obtain good performances on this task. The currently available data, provided along with infection annotations, have been labelled following different guidelines and released in NifTI format. They do not contain complete acquisition and population information and they have been stored according to different criteria (see the Appendix II.A for further details). Some of the choices made during the DICOM to NifTI conversion may strongly affect the quality of data. For example, the MosMed dataset as described by Morozov *et al.* [12] preserves only one slice out of ten during this conversion. This operation results in a significantly loss of resolution with respect to the COVID-19 Challenge dataset. Questioning how much such conversion influences the quantitative analysis is important to improve not only the performance but also the possibility of comparing DL algorithm in a fair modality.

II.2.2 *LungQuant*: a DL based quantification analysis pipeline

The analysis pipeline, which is hereafter referred to as the *LungQuant* system, provides in output the lung and COVID-19 infection segmentation masks, the percentage P of lung volume affected by COVID-19 lesions and the corresponding CT-SS (CT-SS=1 for P<5%, CT-SS=2 for 5% $\leq$ P<25%, CT-SS=3 for 25% $\leq$ P<50%, CT-SS=4 for 50% $\leq$ P<75%, CT-SS=5 for P $\geq$ 75%).

A summary of our image analysis pipeline is reported in Fig. II.1. The central analysis module is a U-net for image segmentation [13] (see sec. II.2.2.1), which is implemented in a cascade of two different U-nets: the first network, U-net$_1$, is trained to segment the lung and the second one, U-net$_2$, is trained to segment the COVID lesions in the CT scans.

II.2.2.1 U-net

For both lung and COVID-19 lesion segmentation, we implemented a U-net using Keras [14], a Python DL API that uses Tensorflow as backend. In Figure II.2 a simplified scheme of our U-net is reported.

Each block of layers in the compression path (left) is made by 3 convolutional layers, ReLu activation functions and instance normalization layers. The input of each block is added to the block output in order to implement a residual connection. In the decompression path (right), one convolutional layer has been replaced by a de-convolutional layer to upsample the images to the input size. In the last layer of the U-nets, a softmax is applied to the final feature map and then the loss is computed.

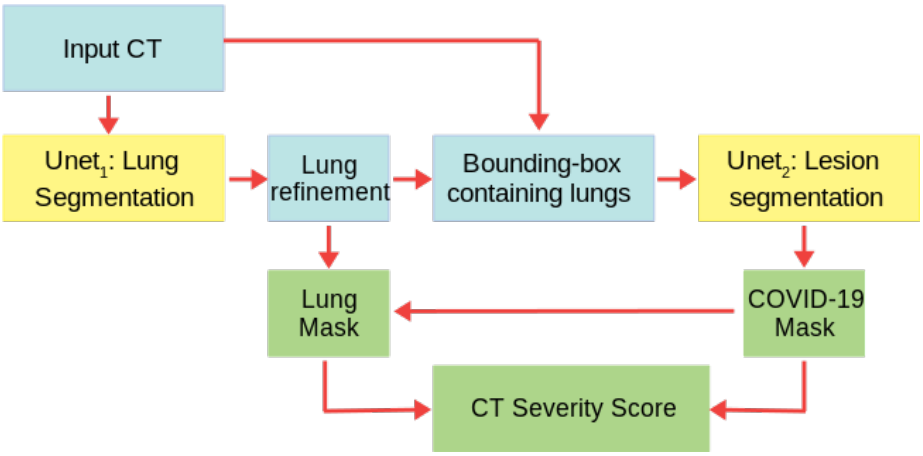


Figure II.1: A summary of the whole analysis pipeline: the input CT scans are used to train U-net$_1$, which is devoted to lung segmentation; its output is refined by a morphology-based method. A bounding box containing the segmented lungs is made and applied to all CT scans for training U-net$_2$, which is devoted to COVID-19 lesion segmentation. Finally, the output of U-net$_2$ is the definitive COVID-19 lesion mask, whereas the definitive lung mask is obtained as the union between the outputs of U-net$_1$ and U-net$_2$. The ratio between the COVID-19 lesion mask and the lung mask provides the CT-SS for each patient.

II.2.2.2 The U-net cascade for lesion quantification and severity score assignment

The input CT scans, whose number of slices is highly variable, have been resampled to matrices of 200x150x100 voxels and then used to train U-net$_1$, which is devoted to lung segmentation, using the three datasets containing original CT scans and lung masks (see Table II.1). The output of U-net$_1$ was refined using a connected-component labeling strategy to remove small regions of the segmented mask not connected with the main objects identified as the lungs. We identified the connected components in the lung masks generated by U-net$_1$ and we excluded those components whose number of voxels was below an empirically-fixed threshold (see Supplementary Materials for further details). We then built for each CT a bounding box enclosing the refined segmented lungs, adding a conservative padding of 2.5 cm. The bounding boxes were used to crop the training images for U-net$_2$, which has the same architecture as U-net$_1$. Training U-net$_2$ to recognize the COVID-19 lesions on a conservative bounding box has two main advantages: it allows to restrict the action volume of the U-net to the region where the lung parenchyma is supposed to be, thus avoiding false-positive findings outside the chest; it facilitates the U-net training phase, as the dimensions of the lungs of different patients are standardized to focus the U-net learning process on the textural patterns characterizing the COVID-19 lesions. The cropped images were resized to a matrix of 200x150x100

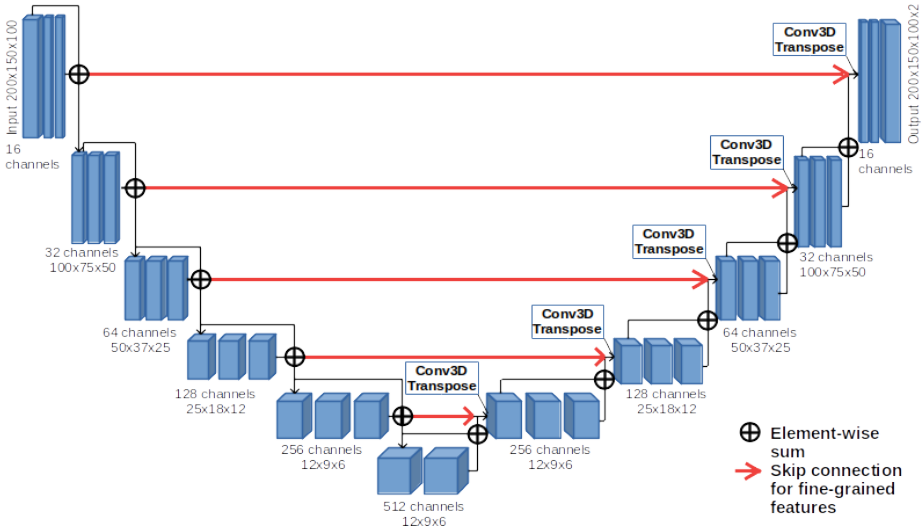


Figure II.2: U-net scheme: the neural network is made of 6 levels of depth. In the compression path (left), the input is processed through convolutions, activation layers (ReLu) and instance normalization layers, while in the decompression one (right), in addition to those already mentioned, 3D Transpose Convolution (de-convolution) layers are also introduced.

voxels. We applied a windowing on the grey-level values of the CT scans to optimize the image contrast for the two segmentation problems: the [-1000, 1000] HU window range for the U-net$_1$ and the [-1000, 300] HU range for U-net$_2$. The first window highlights the contrast between the lung parenchyma and the surrounding tissues, whereas the second one enhances the heterogeneous structure of the lung abnormalities related to the COVID-19 infection. We implemented a data augmentation strategy, relying on the most commonly used data augmentation techniques for DL (see Supplementary Materials for further details) to overcome the problem of having a limited amount of labelled data. We transformed the images with rotations, zooming, elastic transformations and adding gaussian noise.

The *LungQuant* system returns the infection mask as the output of U-net$_2$ and the lung mask as the union between the output of U-net$_1$ and U-net$_2$. This choice has been made *a priori* by design, as U-net$_1$ has been trained to segment the lungs relying on the available annotated data, which are almost totally of patients not affected by COVID-19 pneumonia. Thus, U-net$_1$ is expected to be unable to accurately segment the areas affected by GGO or consolidations; as also these areas are part of the lungs, they should be instead included in the mask.

Lastly, once lung and lesion masks have been identified, the *LungQuant* system computes the percentage of lung volume affected by COVID-19 lesions as the ratio between the volume of the infection mask and the volume of the

lung mask, and converts it into the corresponding CT severity score.

II.2.3 Training details and evaluation strategy for the U-nets

Both U-net$_1$ and U-net$_2$ have been evaluated using the volumetric Dice Similarity Coefficients (vDSC). U-net$_1$ has been trained with the vDSC as loss function, while U-net$_2$ has been trained using the sum of the vDSC and a weighted cross-entropy as error function in order to balance the number of voxels representing lesions and the background (see Supplementary Materials for further details). The performances of the whole system have been evaluated also with the surface Dice Similarity Coefficient (sDSC) for different values of tolerance [15].

II.2.3.1 Cross-validation strategy

To train, validate and test the performances of the two U-nets, we partitioned the datasets into training, validation and test sets. We then evaluated the network performance separately and globally. U-net$_2$ has been trained twice, i.e. on the 60% and 90% of the CT scans of COVID-19-Challenge and Mosmed datasets to investigate the effect of maximizing the training set size on the lesion segmentation. The amount of CT scan used for train, validation and test sets for each U-net is reported in Table II.2. To evaluate the ability of the trained networks to predict the percentage of the affected lung parenchyma and thus the CT-SS classification, we used a completely independent set consisting of 10 CT scans from the COVID-19-CT-Seg dataset, which is the only publicly available dataset containing both lung and infection mask annotations.

U-net$_1$	train	val	test
Plethora	319	40	40
MosMed (91 CT-0)	55	18	18
LCTSC	36	12	12
COVID-19-CT-Seg	/	/	10
U-net$_2^{60\%}$	train (60%)	val (20%)	test
COVID-19 Challenge	119	40	40
MosMed (50 CT-1)	30	10	10
COVID-19-CT-Seg	/	/	10
U-net$_2^{90\%}$	train (90%)	val (10%)	test
COVID-19 Challenge	179	20	/
MosMed (50 CT-1)	45	5	/
COVID-19-CT-Seg	/	/	10

Table II.2: Number of CT scans assigned to the train, validation (val) and test sets used during the training and performance assessment of the U-net$_1$ and the U-net$_2$ networks.

II.3 Results

In this section we report, first, the performance achieved by U-net$_1$ and U-net$_2$, then, the quantification performance of the integrated *LungQuant* system, evaluated on a completely independent test set. We trained both the U-nets for 300 epochs on a NVIDIA V100 GPU using ADAM as optimizer and we kept the models trained at the epoch where the best evaluation metric on the validation set was obtained.

II.3.1 U-net$_1$: Lung segmentation performance

U-net$_1$ for lung segmentation was trained and validated using three different datasets, as specified in Table II.2. Then, we tested U-net$_1$ on each of the three independent test sets and we reported in Table II.3 the performance achieved in terms of vDSC, computed between the segmented masks and the reference ones, both separately for each dataset and globally.

The evaluation of the lung segmentation performances was made in three cases: 1) on CT scans and masks resized to the 200x150x100 voxel array size; 2) on CT scans and masks in the original size before undergoing the morphological refinement; 3) on CT scans and masks in the original size and after the morphological refinement. Even if segmentation refinement has a small effect on vDSC, since it is a volume-based metric, as shown in Table II.3, it is a fundamental step to allow the definition of precise bounding boxes enclosing the lungs, and thus to facilitate the U-net$_2$ learning process.

Test set	Masks of U-net size vDSC	Masks before refinement vDSC	Masks after refinement vDSC
Plethora	0.96 ± 0.02	0.95 ± 0.02	0.95 ± 0.04
MosMed	0.97 ± 0.02	0.97 ± 0.02	0.97 ± 0.02
LCTSC	0.96 ± 0.03	0.95 ± 0.03	0.96 ± 0.01
COVID-19-CT-Seg	0.96 ± 0.01	0.95 ± 0.01	0.95 ± 0.01

Table II.3: Performances achieved by U-net$_1$ in lung segmentation on different test sets, evaluated in terms of the vDSC at three successive stages of the segmentation procedure.

II.3.2 U-net$_2$: COVID-19 lesion segmentation performance

U-net$_2$ for COVID-19 lesion segmentation has been trained and evaluated separately on the COVID-19-Challenge dataset and on the annotated subset of the MosMed dataset, following the train/validation/test partitioning reported in Table II.2. The segmentation performances achieved on the test sets are reported in terms of the vDSC in Table II.4, according to the cross-sample validation scheme.

As expected, the U-net$_2$ performances are higher when both the training set and independent test sets belong to the same data cohort. By contrast, when a U-net$_2$ is trained on COVID-19-Challenge data and tested on Mosmed (and the

U-net	Trained on	Test set	U-net size (vDSC)	Original CT size (vDSC)
$U\text{-}net_2^{60\%}$	COVID-19 Challenge	COVID-19 challenge	0.51 ± 0.24	0.51 ± 0.25
	COVID-19 Challenge	MosMed	0.39 ± 0.19	0.40 ± 0.19
	MosMed	MosMed	0.54 ± 0.22	0.55 ± 0.22
	MosMed	COVID-19 challenge	0.25 ± 0.23	0.25 ± 0.23
	COVID-19 challenge + MosMed	COVID-19 challenge + MosMed	0.49 ± 0.21	0.50 ± 0.21
$U\text{-}net_2^{90\%}$	COVID-19 Challenge + MosMed	COVID-19 Challenge + MosMed	0.64 ± 0.23	0.65 ± 0.23

Table II.4: Performances achieved by $U\text{-}net_2$ in COVID-19 lesion segmentation, evaluated in terms of the vDSC. The composition of the train and test sets is reported in Table II.2.

other way around) performances significantly decrease. This effect is related to the different criteria used to both collect and annotate the data. We obtained a better result with the $U\text{-}net_2$ trained on the COVID-19 Challenge dataset and tested on the MosMed test set, since the network has been trained on a larger data sample and hence it has a higher generalization capability. The best segmentation performances have been obtained by the $U\text{-}net_2$ trained using the 90% of the available data, $U\text{-}net_2^{90\%}$, which reaches a vDSC of 0.65 ± 0.23 on the test set. This result suggests the need to train U-net models on the largest possible data samples in order to achieve higher segmentation performance.

II.3.3 Evaluation of the quantification performance of the *LungQuant* system on a completely independent set

II.3.3.1 Evaluation of lung and COVID-19 lesion segmentations

Once the two U-nets have been trained and the whole analysis pipeline has been integrated into the *LungQuant* system, we tested it on a completely independent set (COVID-19-CT-Seg dataset) of CT scans. The performances of the whole process were quantified both in terms of vDSC and sDSC with tolerance values of 1, 5 and 10 mm (Table II.5). A very good overlap between the predicted and reference lung masks is observable in terms of vDSC, whereas the sDSC values are highly dependent on tolerance values, ranging from moderate to very good agreement measures. Regarding the lesion masks a moderate overlap is measured between the predicted and reference lesion masks in terms of vDSC, whereas the sDSC returns measures extremely dependent on tolerance values, that span from limited to moderately good and ultimately satisfactory performances for tolerance values of 1 mm, 5 mm and 10 mm, respectively. Figure II.3 allows for a visual comparison between the lung and lesion masks provided by the *LungQuant* system integrating $U\text{-}net_2^{90\%}$ and the reference ones.

	Lung Segmentation			
Metrics	vDSC	sDSC (1 mm)	sDSC (5 mm)	sDSC (10 mm)
LungQuant (U-net$_2^{60\%}$)	0.96 ± 0.01	0.66 ± 0.09	0.95 ± 0.02	0.98 ± 0.01
LungQuant (U-net$_2^{90\%}$)	0.95 ± 0.01	0.65 ± 0.09	0.95 ± 0.02	0.98 ± 0.01
	Infection Segmentation			
Metrics	vDSC	sDSC (1 mm)	sDSC (5 mm)	sDSC (10 mm)
LungQuant (U-net$_2^{60\%}$)	0.62 ± 0.09	0.29 ± 0.06	0.75 ± 0.11	0.90 ± 0.09
LungQuant (U-net$_2^{90\%}$)	0.66 ± 0.13	0.36 ± 0.13	0.76 ± 0.18	0.87 ± 0.13

Table II.5: Performances of the *LungQuant* system on the independent COVID-19-CT-Seg test dataset. The vDSC and sDSC computed between the lung and lesion reference masks and those predicted by the *LunQuant* system are reported.

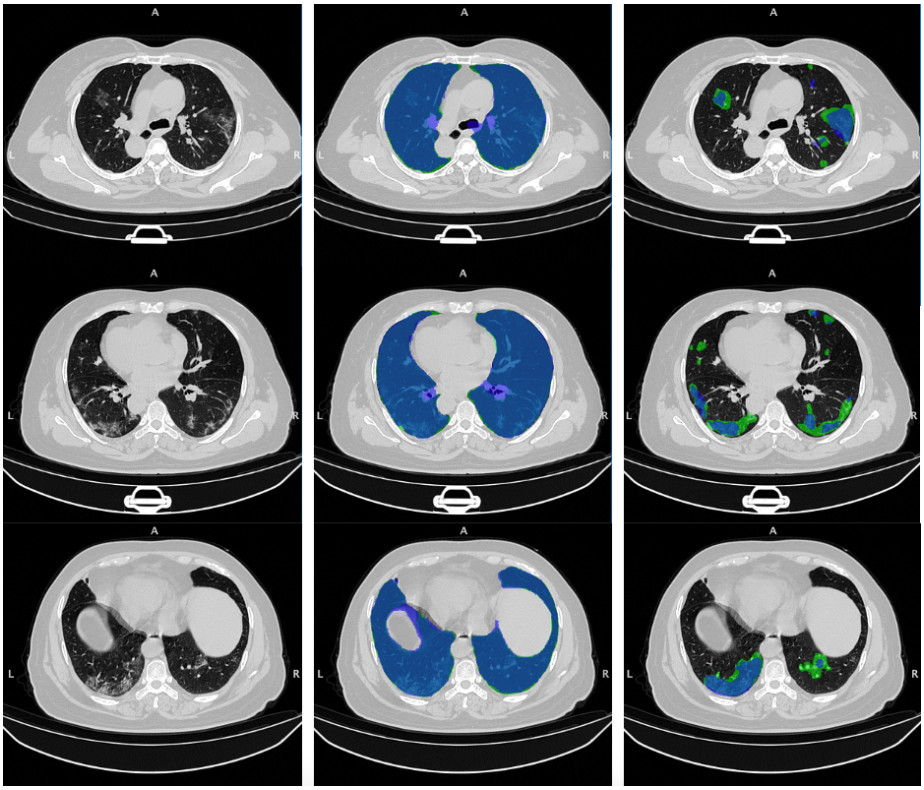


Figure II.3: On the rows: three axial slices of the first CT scan on the COVID-19-CT-Seg test dataset (*coronacases*001.*nii*) are shown. On the columns: original images (left); overlays between the predicted and the reference lung (center) and COVID-19 lesion (right) masks. The reference masks are in green, while the predicted ones, obtained by the *LungQuant* system integrating U-net$_2^{90\%}$,are in blue.

II.3.3.2 Percentage of affected lung volume and CT-SS estimation

The lung and lesion masks provided by the *LungQuant* system can be further processed to derive the physical volumes of each mask and the ratios between the lesion and lung volumes. We show in Fig. II.4 the relationship between the percentage of lung involvement as predicted by the *LungQuant* system vs. the corresponding values for the reference masks of the fully independent test set COVID-19-CT-Seg, for both the *LungQuant* systems with the U-net$_2^{60\%}$ and the U-net$_2^{90\%}$. Despite the limited range of samples to carry out this test, an agreement between the *LungQuant* system output and the reference values is observed for both U-net$_2^{60\%}$ and U-net$_2^{90\%}$. In terms of the mean absolute error (MAE) among the estimated and the reference percentages of affected lung volume (P), we obtained a Mean Absolute Error equal to MAE=4.6% for the LungQuant system with U-net$_2^{60\%}$ and MAE=4.2% for the system with U-net$_2^{90\%}$.

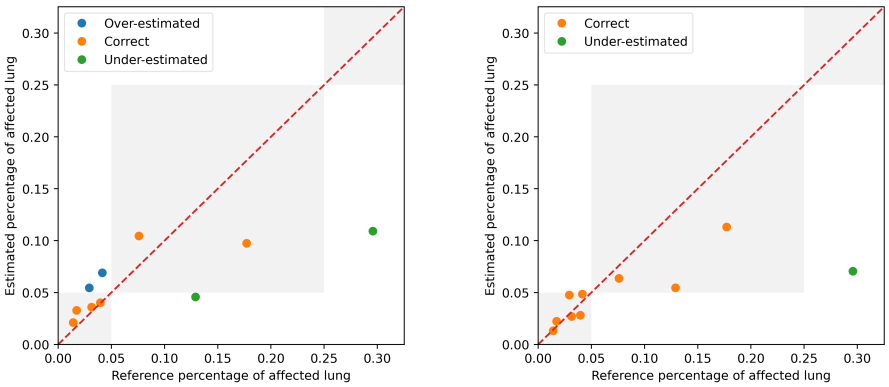


Figure II.4: Estimated percentages P of affected lung volume versus the ground truth percentages, as obtained by the *LungQuant* system integrating U-net$_2^{60\%}$ (left) and U-net$_2^{90\%}$ (right). The gray areas in the plot backgrounds guide the eye to recognize the CT-SS values assigned to each value of P (from left to right: CT-SS=1, CT-SS=2, CT-SS=3).

The accuracy in assigning the correct CT-SS class is reported in Table II.6, together with the number of misclassified cases, for the 10 cases of the COVID-19-CT-Seg dataset. The best accuracy achieved by *LungQuant* is of 90% with U-net$_2^{90\%}$. In all cases, the system misclassifies the examples by 1 class at most.

II.4 Discussion and Conclusion

We developed a fully automated quantification pipeline, the *LungQuant* system, for the identification and segmentation of lungs and pulmonary lesions related to COVID-19 pneumonia in CT scans. The system returns the COVID-19 related lesions, the lung mask and the ratio between their volumes, which is

U-net	Dataset	Accuracy	Misclassified by 1 class	Misclassified by 2 classes
U-net$_2^{60\%}$	COVID-19-CT-Seg	6/10	4/10	0
U-net$_2^{90\%}$	COVID-19-CT-Seg	9/10	1/10	0

Table II.6: Classification performances of the whole system in predicting CT Severity Score on MosMed and COVID-19-CT-Seg datasets. The number of misclassified cases is reported.

converted into a CT Severity Score. The performance obtained against a voxel-wise segmentation ground truth was evaluated in terms of the vDSC, which provides a measure of the overlap between the predicted and the reference masks. The *LungQuant* system achieved a vDSC of 0.95 ± 0.01 in the lung segmentation task and of 0.66 ± 0.13 in segmenting the COVID-19 related lesions on the fully annotated publicly available benchmark COVID-19-CT-Seg dataset of 10 CT scans. The *LungQuant* has been evaluated also in terms of sDSC for different values of tolerance. The results obtained at a tolerance of 5 mm, equal to 0.76 ± 0.18 is satisfactory for our purpose, given the heterogeneity of the labelling process.

Regarding the correct assignment of the CT-SS, the *LungQuant* system showed an accuracy of 90% on the completely independent test set COVID-19-CT-Seg. Despite this result is encouraging, it was obtained on a rather small independent test set, thus, a broader validation on larger data sample with more heterogeneous composition in terms of disease severity is required. Training DL algorithms requires a huge amount of labeled data. The lung segmentation task has been made feasible in this work thanks to the use of lung CT datasets collected for purposes different from the study of COVID-19 pneumonia. Training a segmentation system on these samples had the effect that when we use the trained network to process the CT scan of a patient with COVID-19 lesions, especially in case ground glass opacities and consolidation are very severe, the lung segmentation is not accurate anymore. In order to overcome this problem, the proposed *LungQuant* system returns a lung mask which is the logical union between the output mask of the U-net$_1$ and the infection mask generated by the U-net$_2$. The *LungQuant* system can actually be improved whether lung masks annotation are available on subjects with COVID-19 lesions. A similar problem occurs for the segmentation of ground glass opacities and consolidations. The available data, in fact, are very unbalanced with respect to the severity of COVID-19 disease and, hence, the accuracy in segmenting the most severe case is worse. The current lack of a large dataset, collected by paying attention to adequately represent all categories of disease severity, limits the possibility to carry out accurate training of AI-based models. Finally, we found that the difference in the annotation and collection guidelines among datasets is an issue. Processing aggregated data from different sources can be difficult if labelling has been performed using different guidelines. CT scans should contain the acquisition parameters, usually stored in the DICOM header,

when they are published. The lack of this information is a drawback of our study. If we had that data, we could study more in detail the dependence of the *LungQuant* performances on specific acquisition protocols or scanners. On the contrary, even with this information, it would not be possible to standardize the different annotation styles. The results of *LungQuant* (last 2 rows of Table II.4) demonstrate its robustness across the different datasets even without a dedicated preprocessing step to account for this information.

Appendix II.A
Additional descriptions of Materials and Methods

II.A.1 Characteristics of the public datasets used in the study

II.A.1.1 The Plethora dataset

The PleThora dataset [9] is a chest CT scan collection with thoracic volume and pleural effusion segmentations, delineated on 402 CT studies of the Non-Small Cell Lung Cancer (NSCLC) radiomics dataset, available through the The Cancer Imaging Archive (TCIA) repository [16]. This dataset has been made publicly available to facilitate improvement of the automatic segmentation of lung cavities, which is typically the initial step in the development of automated or semi-automated algorithms for chest CT analysis. In fact, automatic lung identification struggles to perform consistently in subjects with lung diseases. The PleThora lung annotations have been produced with a U-net based algorithm trained on chest CT of subjects without cancer, manually corrected by a medical student and revised by a radiation oncologist or a radiologist.

II.A.1.2 The 2017 Lung CT Segmentation Challenge dataset

The Lung CT Segmentation Challenge (LCTSC) dataset consists of CT scans of 60 patients, acquired from 3 different institutions and made publicly available in the context of the 2017 Lung CT Segmentation Challenge [10]. Since the aim of this challenge was to foster the development of auto-segmentation methods for organs at risk in radiotherapy, the lung annotations followed the RTOG 1106 contouring atlas.

II.A.1.3 The 2020 COVID-19 Lung CT Lesion Segmentation Challenge dataset

The 2020 COVID-19 Lung CT Lesion Segmentation Challenge dataset (COVID-19 Challenge) is a public dataset consisting of unenhanced chest CT scans of 199 patients with positive RT-PCR for SARS-CoV-2 [11]. Each CT is accompanied with the ground truth annotations for COVID-19 lesions. Data has been provided in NIfTI format by The Multi-national NIH Consortium for CT AI in COVID-19 via the NCI TCIA public website [16]. Annotations have been made using a COVID-19 lesion segmentation model provided by NVIDIA, which takes a full CT

chest volume and produces pixel-wise segmentations of COVID-19 lesions. These segmentations have been adjusted manually by a certified radiologists board, in order to give 3D consistency to lesion masks. The dataset annotation was made possible through the joint work of Children's National Hospital, NVIDIA and National Institutes of Health for the COVID-19-20 Lung CT Lesion Segmentation Grand Challenge.

The dataset and the annotations have been made available in the context of a MICCAI-endorsed international challenge (https://covid-segmentation.grand-challenge.org/) which had the aim to compare AI-based approaches to automated segmentation of COVID-19 lung lesions.

II.A.1.4 The MosMed dataset

MosMed [12] is a COVID-19 chest CT dataset collected by the Research and Practical Clinical Center for Diagnostics and Telemedicine Technologies of the Moscow Health Care Department. It includes CT studies taken from 1110 patients. Each study is represented by one series of images reconstructed into soft tissue mediastinal window. MosMed provides 5 labeled categories, based on the percentage of lung parenchyma affected by COVID-19 lesions. The 5 categories of lung involvement and their correspondence to the CT-SS scale are described in Table II.7. The first category (CT-0) contains cases with normal lung tissue and no CT-signs of viral pneumonia, whereas the other categories contain GGO (CT-1 and CT-2) and both GGO and regions of consolidation in the higher classes (CT-3 and CT-4).

MosMed CT category	N. of cases	Percentage P of involved lung parenchyma	Corresponding CT-SS
0	254	$P = 0$	0
1	684	$0 < P \leq 25$	1, 2
2	125	$25 < P \leq 50$	3
3	45	$50 < P \leq 75$	4
4	2	$75 < P \leq 100$	5

Table II.7: MosMed severity categories defined on the basis of the percentage P of lung volume affected by COVID-19 lesions. The correspondence to the CT-SS scale is reported.

A small subset of class CT-1 cases (50 patients) had been annotated by expert radiologists with the support of MedSeg software (2020 Artificial Intelligence AS). The annotations consist of binary masks in which white voxels represent both ground-glass opacifications and consolidations. Both CT scans and annotations were provided in NIfTI format. During the DICOM-to-NIfTI conversion process, only one slice out of ten was preserved and, as a result, MosMed CT scans have a reduced total number of slices with respect to the other datasets.

II.A.1.5 The COVID-19-CT-Seg dataset

The COVID-19-CT-Seg dataset is a collection of CT scans taken from the Coronacases Initiative and Radiopaedia [4]. It contains 20 CT scans tested positive for COVID-19 infection. This public dataset contains both lung and infection annotations. The ground truth has been made in three steps: first, junior radiologists (1-5 years of experience) delineated lungs and infections annotations, then two radiologists (5-10 years of experience) refined the labels and finally the annotations have been verified and optimized by a senior radiologist (more than 10 years of experience in chest radiology). The annotations have been produced with the ITK-SNAP software. Ten CT images of this dataset were provided in 8-bit depth, therefore, we decided to not use them.

II.A.2 Additional training details and evaluation strategy for the U-nets

II.A.2.1 Evaluation metrics

The segmentation performances for both U-nets have been evaluated with the volumetric Dice Similarity Coefficient (vDSC), computed between the true mask volume (V_{true}) and the predicted mask volume ($V_{predict}$), and with the surface Dice Similarity Coefficient (sDSC), computed between the true surface (S_{true}), and the predicted one defined, ($S_{predict}$) [15], as follows;

$$\text{vDSC}_{metric} = \frac{2 \cdot |V_{true} \cap V_{predict}|}{|V_{true}| + |V_{pred}|} \tag{II.1}$$

$$\text{sDSC}_{metric} = \frac{2 \cdot |S_{true} \cap S_{predict}|}{|S_{true}| + |S_{pred}|} \tag{II.2}$$

The loss function used to train the U-net for lung segmentation is the vDSC loss, defined as follows

$$\text{vDSC}_{loss} = 1 - \frac{2 \cdot |M_{true} \cap M_{pred}|}{|M_{true}| + |M_{pred}|} \tag{II.3}$$

and computed only on the foreground (white voxels). We used this strategy in order to avoid giving excessive weight to the background (black voxels), since the number of black and white voxels is quite unbalanced in favor of the former.

For U-net$_2$, we used a loss function (L) consisting of the sum of the vDSC loss and a weighted cross-entropy (CE), defined as follows:

$$L = vDSC_{loss} + CE_{weighted} \tag{II.4}$$

$$CE_{weighted} = w(x) \sum_{x \in \Omega} log(M_{true}(x) \cdot M_{pred}(x)) \tag{II.5}$$

where $w(x)$ is the weight map which takes into account the frequency of white voxels, x is the current sample and Ω is the training set.

Since the background class is larger than the foreground class on the order 10^3, we computed the weight map $w(x)$ for each ground-truth segmentation to increase the relevance of the underrepresented class, following the approach described in [17]. The weight map was defined as $w(x) = w_0/f_j$ where f_j is the average number of voxels of the j^{th} class over the entire training data set $(j = 0, 1)$ and w_0 is the the average between the frequencies f_j.

II.A.2.2 Data augmentation

Data augmentation is a strategy to increase the size of the training set by synthetically generating additional training images through geometric transformations. This technique is particularly important to improve the generalization capability of the model, especially in the case of a limited number of training samples. In our work, we applied data augmentation during the data pre-processing phase (after defining the bounding boxes enclosing the segmented lungs) in order to generate a fixed number of augmented images for each original data. We chose an augmentation factor equal to 2 which means that the number of artificially generated images is twice the number of the original training set. For each image in the training set, two of the following geometric transformations were randomly chosen:

- *Zooming.* The CT image and the ground truth masks were zoomed in the axial plane, using a third-order spline interpolation and the k-nearest neighbor method, respectively. The zooming factor was randomly chosen among the following values: 1.05, 1.1, 1.15, 1.2.

- *Rotation.* The CT image and the ground truth mask were rotated in the axial plane, using a third-order spline interpolation and the k-nearest neighbor method, respectively. The rotation angle was randomly sampled among the following values: -15°, -10°, -5°, 5°, 10°, 15°.

- *Gaussian noise.* An array of noise terms randomly drawn from a normal distribution was added to the original CT image. For each image, the mean of the Gaussian distribution was randomly sampled in the [-400, 200] HU range and the standard deviation randomly chosen among 3 values: 25, 50, 75 HU.

- *Elastic deformation.* An elastic distortion was applied to the original 3D CT and mask arrays following the approach of Simard *et al.* [18]. This transformation has two parameters: the elasticity coefficient which we fixed to 12 and the scaling factor, fixed to 1000.

- *Motion blurring.* Slice by slice, we convolved the CT image with a linear kernel (i.e. ones along the central row and zero elsewhere for a matrix of size $k \times k$) through the function filter2D, defined in the OpenCV Python library [19], keeping the output image size the same as the input image. The filter is applied with a kernel size of 4, 3, and 3, in the anterior-posterior, latero-lateral and cranio-caudal direction, respectively.

An example of the application of these augmentation techniques to one CT scan of the dataset is provided in Fig. II.5.

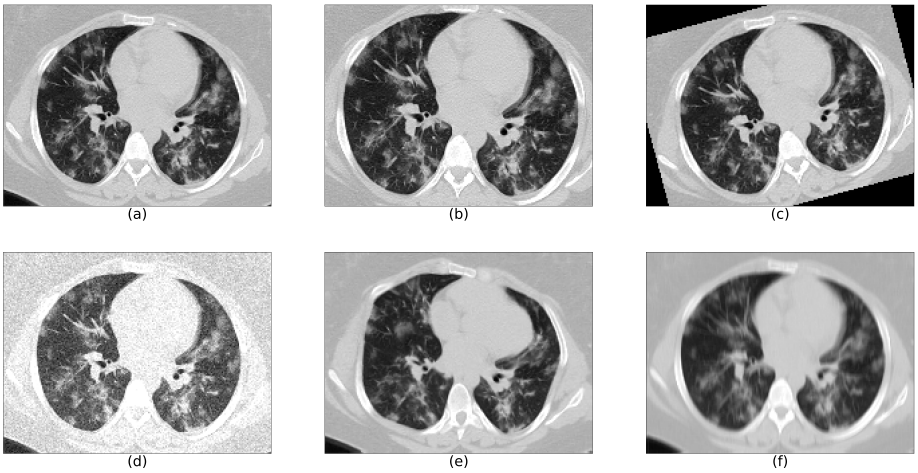


Figure II.5: Data augmentation to increase the diversity of dataset: a) Image without data augmentation; b) Zooming; c) Rotation; d) Gaussian noise; e) Elastic deformations; f) Motion blurring.

II.A.3 Morphological refinement of U-net$_1$ lung segmentation

In order to remove false-positive regions (*i.e.* voxels misclassified as lung parts), at first, we identified the connected components in the lung masks generated by U-net$_1$, then, we excluded those components whose number of voxels was below an empirically-fixed threshold. This threshold was set to the 40% of the foreground mask, and it was reduced to 30% whether the resulting number of voxels was found to be lower than the 65% of the initial mask provided by U-net$_1$. Figure II.6 shows some examples of how this procedure works on real CT scans.

II.A.4 Generation of a set of reference lung segmentation for model training

As reported in Table 1 (main paper), the available datasets with lung mask annotations, which were necessary to train the U-net for lung segmentation, are mainly of subjects affected by lung cancer (Plethora and LCTSC datasets). To complement this sample with subjects without lesions, and, at the same time, to expose to U-net to the acquisition characteristics of the MosMed CT scans, we generated the lung mask annotations for a subset of subjects of the CT-0 MosMed category, i.e. that of subjects without COVID-19 lesions.

An in-house lung segmentation algorithm was developed for this purpose and implemented in *matlab* (The MathWorks, Inc.). It is based on the following steps: 1) CT windowing in the [-1000,1000] HU range; 2) rough segmentation

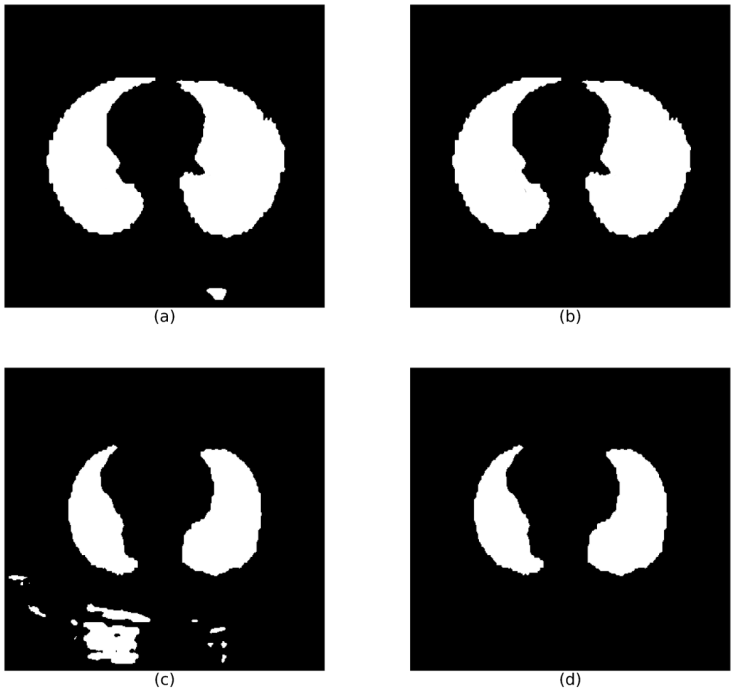


Figure II.6: Morphological refinement of the U-net$_1$ output: a) and c) lung masks as generated by U-net$_1$; b) and d) refined masks after the connected component selection.

of the lungs on a central coronal slice (Otsu binary thresholding and removal of components connected with the image border) to define the minimum and maximum axial coordinates of the lung region; 3) 2D rough segmentation of the lungs on each axial slice (same procedure as the previous step) to generate a 3D seed mask for the following step; 4) segmentation of the lung parenchyma by an active contour model (*activecontour* matlab function); 5) filling holes (e.g. vessels and airway walls) with 3D morphological operators (*imclose* matlab function).

This algorithm, which accurately segments the lung parenchyma in absence of lesions, has very limited performance on CT scans of subjects with COVID-19 lesions.

Acknowledgements. This work has been carried out within the Artificial Intelligence in Medicine (AIM) project funded by INFN (CSN5, 2019-2021), https://www.pi.infn.it/aim. We are grateful to the staff of the Data Center of the INFN Division of Pisa. We thank the CINECA Italian computing center for making available part of the computing resources used in this paper; in particular, Dr. Tommaso Boccali (INFN, Pisa) as PI of PRACE Project Access #2018194658 and a 2021 ISCRA-C grant. Moreover, we thank the EOS cluster

of Department of Mathematics “F. Casorati” (Pavia) for computing resources.

Conflict of interest

The authors declare that they have no conflict of interest.

Ethical approval and informed consent

All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki Declaration and its later amendments or comparable ethical standards.

Informed consent was obtained from all individual participants included in the study.

References

[1] Carotti, M., Salaffi, F., Sarzi-Puttini, P., *et al.*, “Chest CT features of coronavirus disease 2019 (COVID-19) pneumonia: Key points for radiologists,” *La Radiologia Medica*, vol. 125, no. 7, pp. 636–646, 2020.

[2] Lessmann, N., Sánchez, C. I., Beenen, L., *et al.*, “Automated assessment of COVID-19 reporting and data system and chest CT severity scores in patients suspected of having COVID-19 using artificial intelligence,” *Radiology*, vol. 298, no. 1, E18–E28, 2021.

[3] Fang, X., Kruger, U., Homayounieh, F., *et al.*, “Association of ai quantified COVID-19 chest CT and patient outcome,” *International Journal of Computer Assisted Radiology and Surgery*, vol. 16, pp. 435–445, 2021.

[4] Ma, J., Wang, Y., An, X., *et al.*, “Toward data-efficient learning: A benchmark for COVID-19 CT lung and infection segmentation,” *Medical Physics*, vol. 48, no. 3, pp. 1197–1210, 2021.

[5] Xie, W., Jacobs, C., Charbonnier, J.-P., and Van Ginneken, B., “Relational modeling for robust and efficient pulmonary lobe segmentation in CT scans,” *IEEE Transactions on Medical Imaging*, vol. 39, no. 8, pp. 2664–2675, 2020.

[6] Hofmanninger, J., Prayer, F., Pan, J., Röhrich, S., Prosch, H., and Langs, G., “Automatic lung segmentation in routine imaging is primarily a data diversity problem, not a methodology problem,” *European Radiology Experimental*, vol. 4, no. 1, pp. 1–13, 2020.

[7] Fang, X. and Yan, P., “Multi-organ segmentation over partially labeled datasets with multi-scale feature abstraction,” *IEEE Transactions on Medical Imaging*, vol. 39, no. 11, pp. 3619–3629, 2020.

[8] Fortin, J.-P., Cullen, N., Sheline, Y. I., *et al.*, “Harmonization of cortical thickness measurements across scanners and sites,” *Neuroimage*, vol. 167, pp. 104–120, 2018.

[9] Kiser, K. J., Ahmed, S., Stieb, S., *et al.*, "PleThora: Pleural effusion and thoracic cavity segmentations in diseased lungs for benchmarking chest CT processing pipelines," *Medical Physics*, vol. 47, no. 11, pp. 5941–5952, 2020.

[10] Yang, J., Sharp, G., Veeraraghavan, H., *et al.*, *Data from Lung CT Segmentation Challenge. The Cancer Imaging Archive.* 2017.

[11] An, P., Xu, S., Harmon, S. A., *et al.*, *CT Images in COVID-19*, 2020.

[12] Morozov, S. P., Andreychenko, A., Pavlov, N., *et al.*, "Mosmeddata: Chest CT scans with COVID-19 related findings dataset," *arXiv preprint arXiv:2005.06465*, 2020.

[13] Ronneberger, O., Fischer, P., and Brox, T., "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, Springer, 2015, pp. 234–241.

[14] Chollet, F., *Keras*, https://keras.io, 2015.

[15] Kiser, K. J., Barman, A., Stieb, S., Fuller, C. D., and Giancardo, L., "Novel autosegmentation spatial similarity metrics capture the time required to correct segmentations better than traditional metrics in a thoracic cavity segmentation workflow," *Journal of Digital Imaging*, vol. 34, no. 3, pp. 541–553, 2021.

[16] Clark, K., Vendt, B., Smith, K., *et al.*, "The cancer imaging archive (tcia): Maintaining and operating a public information repository," *Journal of Digital Imaging*, vol. 26, no. 6, pp. 1045–1057, 2013.

[17] Phan, T. H. and Yamamoto, K., "Resolving class imbalance in object detection with weighted cross entropy losses," *arXiv preprint arXiv:2006.01413*, 2020.

[18] Simard, P. Y., Steinkraus, D., Platt, J. C., *et al.*, "Best practices for convolutional neural networks applied to visual document analysis.," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, Edinburgh, vol. 3, 2003, pp. 958–963.

[19] Bradski, G., "The OpenCV library," *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, vol. 25, no. 11, pp. 120–123, 2000.

Part B

Fast DL reconstruction techniques for preclinical MRF

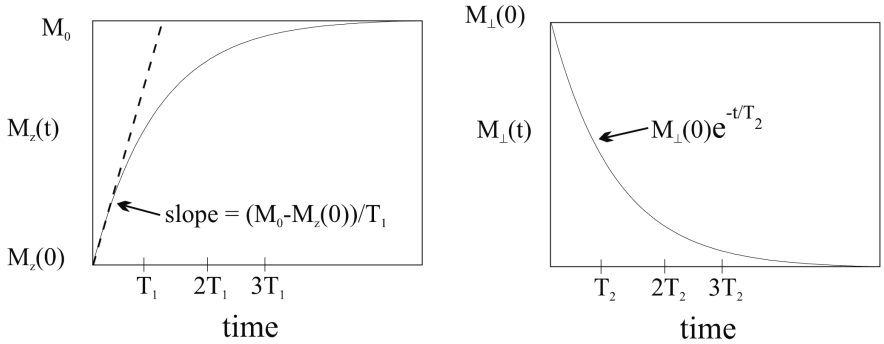
Introduction

In the second part of this thesis, we propose a DL model and a hyperparameter optimization strategy to generate T_1 and T_2 maps acquired using the Magnetic Resonance Fingerprinting (MRF) methodology. MRF is a quantitative magnetic resonance imaging (QMRI) [1] technique, a powerful approach that allows the generation of *quantitative maps* that represent distinct tissue properties.

Although traditional *qualitative MRI* is one of the most important tomographic tools adopted in clinical practice, it primarily provides information about morphology and anatomy, which are visually assessed by clinical experts. The main limitation of qualitative MRI images lies in their relative scale of intensities, making them highly susceptible to variations caused by different acquisition sequences and parameters. On the other hand, QMRI offers pixel-level measurements of specific biological or physical properties, providing absolute quantitative measurements. This characteristic makes QMRI more suitable for accurate comparisons and analysis. In this chapter, we will focus on two specific quantitative maps: T_1 and T_2 relaxation maps.

To comprehend the concepts of T_1 and T_2 relaxation times along with their physical interpretations, we will provide a concise overview of the fundamentals of MRI. A complete description of MRI technique can be found in reference [2]. MRI is based on the principle of Nuclear Magnetic Resonance (NMR), a spectroscopic technique that allows the study of different physical properties of materials. The NMR phenomenon arises from the interaction of ^1H nuclear spins with an external magnetic field $\boldsymbol{B}_0$. By applying a static external magnetic field to the material of interest, the nuclear magnetic moments begin to move along the field's axis through a circular motion called precession. The combination of these individual magnetic moments gives rise to a net macroscopic magnetization vector $\boldsymbol{M}$ of the system. At the equilibrium, assuming that $\boldsymbol{B}_0$ is aligned along the z axis, the projection on the xy plane of the magnetization vector $\boldsymbol{M}_\perp$ has an expectation value equal to zero, while the component of the magnetization along z, M_z, assumes a constant value, M_0. This equilibrium state is perturbed by the application of a second Radio Frequency (RF) oscillating magnetic field $\boldsymbol{B}_1$, perpendicular to the direction of the static field and whose frequency matches the intrinsic precession frequency of the nuclear spin. Once the RF pulse is turned off, the magnetization will return to the equilibrium state, through a process called relaxation. By analyzing the radiated signal emitted by the magnetization during the relaxation, it is possible to obtain microscopic information of the investigated system. Specifically, during the relaxation, the magnetization vector $\boldsymbol{M}$ can be described by the empirical equation known as the *Bloch equation* [3]:

$$\frac{d\boldsymbol{M}}{dt} = \gamma \boldsymbol{M} \times \boldsymbol{B}_0 + \frac{1}{T_1}(M_0 - M_z)\boldsymbol{z} - \frac{1}{T_2}\boldsymbol{M}_\perp \tag{II.6}$$



(a) T_1 (b) T_2

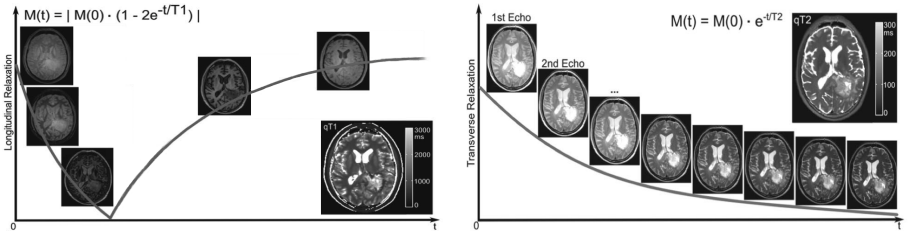
Figure II.7: In Figure II.7a illustrates the recovery of the longitudinal component of magnetization from its initial value $\boldsymbol{M_z}(0)$ to reach the equilibrium value M_0. In Figure II.7b, it is represented the decay of the magnitude of the transverse magnetization in the co-rotating reference frame (i.e. a reference frame rotating around the z axis with the same frequency as $\boldsymbol{B}_1$), starting from an initial value $\boldsymbol{M}_\perp(0)$. Source [2].

where γ is the nuclear gyromagnetic ratio of ^1H. From the equation we can observe that, the relaxation of the magnetization is characterized by two separate processes, each with their own time constants:

- T_1 or *longitudinal relaxation time* is the relaxation time of M_z. It arises from nuclear spin interactions with the surrounding microenvironment (*spin-lattice* interactions), Figure II.7a;
- T_2 or *transverse relaxation time* is the relaxation time of $\boldsymbol{M}_\perp = M_x\boldsymbol{x}+M_y\boldsymbol{y}$. This relaxation is a result of interactions between nuclear spins (*spin-spin* interactions), Figure II.7b.

Considering these aspects, it becomes evident that measurements of T_1 and T_2 relaxation times provide insight into the physical properties of the studied materials, as they are influenced by interactions among ^1H nuclear spins within their surrounding environment. If we consider NMR relaxation of water protons of the human body, T_1 and T_2 relaxation times have the potential to quantify specific physical properties of tissues, which may be used to monitor medical conditions or disease progression. In this context, QMRI aims to generate images known as quantitative maps, in which each pixel's intensity represents the corresponding physical property's value.

Most QMRI approaches involve a series of measurements that vary one MRI acquisition parameter at a time (e.g., flip angle (FA), echo time (TE), inversion time (TI), or repetition time (TR)) to calculate the QMRI parameter value from the resulting changes in pixel-wise intensities. Acquisition imperfections (e.g., magnetic field inhomogeneities) are either averaged out by the series of



(a) T_1 (b) T_2

Figure II.8: Conventional T_1 and T_2 mapping approach, reprinted from [4]. In Figure II.8a, a series of IR-SE or IR-TSE images with different inversion times are acquired, enabling the calculation of the T_1 value for each pixel. In Figure II.8b, a series of SE or TSE images are acquired with different echo times. An exponential fitting is performed for each pixel to determine the T_2 value.

measurements or are directly measured and subsequently corrected. Compared to conventional MRI, QMRI parametric maps are more reliable since they are ideally independent of scanner hardware, software, and acquisition site. The typical MRI sequences employed for the acquisition of T_1 and T_2 maps are:

- *Spin Echo (SE)* or *Turbo Spin Echo (TSE)*: images are acquired by keeping all the parameters fixed and varying TE. A pixel-wise exponential fitting is performed on the series of images, according to the following equation:

$$y(TE) = a_1 \cdot e^{-TE/T_2} + b_1 \tag{II.7}$$

  The T_2 value is calculated for each pixel and the T_2 map is generated, Figure II.8b.

- *Inversion Recovery-Spin Echo (IR-SE)* or *Inversion Recovery-Turbo Spin Echo (IR-TSE)* (i.e., SE or TSE with an initial π inversion RF-pulse): images are acquired by keeping all the parameters fixed and varying TI. A pixel-wise fitting is performed on the data, according to:

$$y(TI) = a_2 \cdot |1 - b_2 \cdot e^{-TI/T_1}| \tag{II.8}$$

  by the fitted T_1 parameter the corresponding T_1 map is generated, Figure II.8a.

While QMRI offers several advantages, it is important to acknowledge its two primary limitations. Firstly, it permits the measurement of only one map at a time, which results in increased acquisition time when multiple maps are required. Furthermore, the acquisition times for these sequences are notably lengthy, making QMRI impractical for routine clinical use.

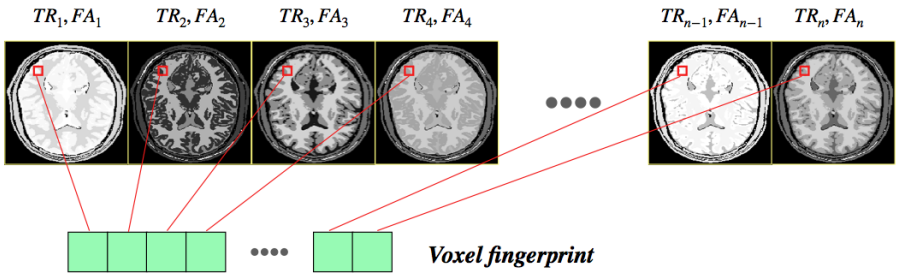


Figure II.9: A sequence of images is acquired by varying some instrumental parameters (TR and FA) in a pseudo-random way. Every voxel generates a unique signal trajectory, called voxel fingerprint.

Magnetic Resonance Fingerprinting

MRF has recently emerged as a technique able to overcome the problems described above [5], [6]. It is a fast technique able to obtain multi-parametric maps in one-shot measurement. The MRF methodology consists of three parts: signal acquisition, dictionary creation, and pattern matching for maps generation. These three phases will be briefly described here below.

Signal Acquisition The key idea of MRF is to apply a train of RF pulses varying in a pseudo-random way some MR instrumental parameters (e.g. FA, TR etc.) that provides fingerprint-like signal evolutions for combinations of desired tissue properties, such as T_1 and T_2, Figure II.9. In this way, tissues generate unique signal trajectories, or "fingerprints", which reflect specific properties of the scanned material. Many different acquisition sequences can be used (such as steady-state free precession (FISP), echo planar imaging (EPI), etc.), as long as they offer sufficient signal-to-noise ratio and sensitivity to the properties of interest. Additionally, it is necessary that the signal evolution trajectories produced by these sequences should be well-understood and easily described by the Bloch equations II.6.

Dictionary Generation All the acquired voxel fingerprints are compared with a precomputed dictionary of possible signal trajectories realized using an appropriate signal simulation model. The dictionary is generated by modeling the spin behavior during the acquisition. Usually, the trajectories are simulated using the Bloch equations II.6, which model the effects of the sequence on a single voxel.

Pattern Matching The final step after acquisition and dictionary generation involves pattern matching to identify the dictionary entry that best represents the tissue properties of each voxel. This is achieved by computing the inner

product between the acquired voxel fingerprint and each dictionary entry to find the entry with the highest similarity value. This process is repeated for every voxel in the image, generating multiple property maps from a single acquisition scan.

The important advantages of MRF compared to other QMRI approaches are: i) the shorter acquisition time compared to the conventional T_1 and T_2 mapping, ii) the simultaneous estimation of different parametric maps and iii) less sensitivity to instrumental constraints, including temporal delays and motion artifacts. All these aspects could improve the patient comfort during the exam and bring economic benefits to the hospital, related to the energy-saving needed to operate the machines and to the possibility to analyze more patients.

Limitations of the reconstruction process

One of the primary limitations of the MRF methodology is the computationally expensive reconstruction process. This is because the dictionary must be large and dense enough to cover all the possible tissue values. Additionally, the size of the dictionary increases exponentially with the number of tissue parameters encoded in the simulated trajectories. This rapid growth can lead to the creation of exceedingly large dictionaries, which require significant computational resources for processing. The resulting high memory, storage, and computational requirements are limiting factors for the clinical adoption of MRF.

Some dictionary compression methods have been recently developed to reduce the dictionary size and to speed up the matching process. For example, in [7] the singular value decomposition (SVD) is used as a compression tool to reduce the time-dimension of the dictionary, resulting in fewer points to compare. However, the dictionary compression by the SVD method is itself a computationally expensive operation. Another approach to face the problem is proposed by [8], which uses a polynomial fitting of the fingerprinting signals in order to reduce the memory requirements for storing the dictionary. However, this technique introduces additional approximations which can affect the accuracy of the reconstructed maps.

Fast DL reconstruction techniques for preclinical MRF

In the following work III, we present an optimized MRF framework to provide quantitative multiparametric maps for preclinical studies. We implement the MRF acquisition by comparing two different acquisition parameter profiles. To estimate T_1 and T_2 maps from the MRF trajectories, we proposed a DL method and a hyperparameters optimization algorithm to automatically select the best reconstruction methodology. A summary of our MRF analysis framework is depicted in Figure II.10. This new DL approach improves the reconstruction performance in the estimation of both T_1 and T_2 maps as well as the computational time necessary for the calculation. Furthermore, we demonstrated that the DL system allows us to use a lower number of MRF

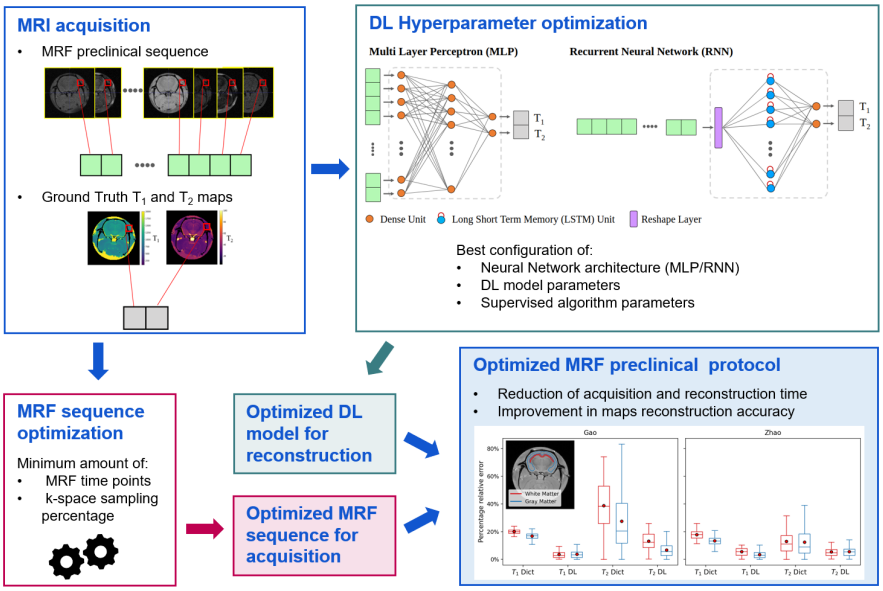


Figure II.10: Overview of the MRF analysis framework. The system consists of three phases: i) MRF acquisition, ii) DL hyperparameter optimization and iii) MRF sequence optimization.

images and a lower k-space sampling percentage, significantly reducing the acquisition time of MRI examinations.

References

[1] Keenan, K. E., Biller, J. R., Delfino, J. G., *et al.*, "Recommendations towards standards for quantitative MRI (qMRI) and outstanding needs," *Journal of Magnetic Resonance Imaging: JMRI*, vol. 49, no. 7, e26, 2019.

[2] Brown, R. W., Cheng, Y.-C. N., Haacke, E. M., Thompson, M. R., and Venkatesan, R., *Magnetic resonance imaging: physical principles and sequence design*. John Wiley & Sons, 2014.

[3] Bloch, F., "Nuclear induction," *Physical Review*, vol. 70, no. 7-8, p. 460, 1946.

[4] Hattingen, E., Jurcoane, A., Nelles, M., *et al.*, "Quantitative MR imaging of brain tissue and brain pathologies," *Clinical Neuroradiology*, vol. 25, pp. 219–224, 2015.

[5] Ma, D., Gulani, V., Seiberlich, N., *et al.*, "Magnetic resonance fingerprinting," *Nature*, vol. 495, no. 7440, pp. 187–192, 2013.

[6] Bipin Mehta, B., Coppo, S., Frances McGivney, D., *et al.*, "Magnetic resonance fingerprinting: A technical review," *Magnetic Resonance in Medicine*, vol. 81, no. 1, pp. 25–46, 2019.

[7] McGivney, D. F., Pierre, E., Ma, D., *et al.*, "SVD compression for magnetic resonance fingerprinting in the time domain," *IEEE Transactions on Medical Imaging*, vol. 33, no. 12, pp. 2311–2322, 2014.

[8] Yang, M., Ma, D., Jiang, Y., *et al.*, "Low rank approximation methods for MR fingerprinting with large scale dictionaries," *Magnetic Resonance in Medicine*, vol. 79, no. 4, pp. 2392–2400, 2018.

Paper III

Fast Deep Learning reconstruction techniques for preclinical Magnetic Resonance Fingerprinting

Raffaella Fiamma Cabini[1,2], Leonardo Barzaghi[1,2,3], Davide Cicolari[4,5,6,7], Paolo Arosio[5,6], Stefano Carrazza[5,6], Silvia Figini[8,2], Marta Filibian[9,2], Andrea Gazzano[11], Rolf Krause[10], Manuel Mariani[4], Marco Peviani[11], Anna Pichiecchio[12,3], Diego Ulisse Pizzagalli[10], Alessandro Lascialfari[4,2]

Published in *NMR in Biomedicine*, August 2023, DOI: https://doi.org/10.1002/nbm.5028

Abstract

We propose a deep learning (DL) model and a hyperparameter optimization strategy to reconstruct T_1 and T_2 maps acquired with the magnetic resonance fingerprinting (MRF) methodology. We applied two different MRF sequence routines to acquire images of ex vivo rat brain phantoms using a 7-T preclinical scanner. Subsequently, the DL model was trained using experimental data, completely excluding the use of any theoretical MRI signal simulator. The best combination of the DL parameters was implemented by an automatic hyperparameter optimization strategy,

[1] Department of Mathematics, University of Pavia, Via Ferrata 5, 27100 Pavia, Italy
[2] INFN, Istituto Nazionale di Fisica Nucleare, Pavia Unit, Via Bassi 6, 27100, Pavia, Italy
[3] Advanced Imaging and Radiomics Center, Department of Neuroradiology, IRCCS Mondino Foundation, Via Mondino 2, 27100 Pavia, Italy
[4] Department of Physics, University of Pavia, Via Bassi 6, 27100, Pavia, Italy
[5] Department of Physics, University of Milan, Via Celoria 16, 20133, Milan, Italy
[6] INFN, Istituto Nazionale di Fisica Nucleare, Milano Unit, Via Celoria 16, 20133, Milan, Italy
[7] Department of Medical Physics, ASST GOM Niguarda, P.zza Ospedale Maggiore 3, 20162, Milan, Italy
[8] Department of Social and Political Science, University of Pavia, Corso Carlo Alberto 3, 27100 Pavia, Italy
[9] Centro Grandi Strumenti, University of Pavia, Via Bassi 21, 27100 Pavia, Italy
[10] Euler institute, USI, Via La Santa 1, 6962 Lugano, Switzerland
[11] Laboratory of Cellular and Molecular Neuropharmacology, Department of Biology and Biotechnology "L. Spallanzani", University of Pavia, Via Ferrata 9, 27100 Pavia, Italy
[12] Department of Brain and Behavioural Sciences, University of Pavia, Via Mondino 2, 27100 Pavia, Italy

whose key aspect is to include all the parameters to the fit, allowing the simultaneous optimization of the neural network architecture, the structure of the DL model, and the supervised learning algorithm. By comparing the reconstruction performances of the DL technique with those achieved from the traditional dictionary-based method on an independent dataset, the DL approach was shown to reduce the mean percentage relative error by a factor of 3 for T_1 and by a factor of 2 for T_2, and to improve the computational time by at least a factor of 37. Furthermore, the proposed DL method enables maintaining comparable reconstruction performance, even with a lower number of MRF images and a reduced k-space sampling percentage, with respect to the dictionary-based method. Our results suggest that the proposed DL methodology may offer an improvement in reconstruction accuracy, as well as speeding up MRF for preclinical, and in prospective clinical, investigations.

Keywords: MR Fingerprinting, Deep Learning, Neural Networks, Quantitative MRI

III.1 Introduction

In recent years, there has been growing interest in quantitative multiparametric mapping (MPM) by means of magnetic resonance imaging (MRI) as a valuable tool for measuring, until the millimeter scale of resolution, physical properties such as nuclear spin–lattice relaxation time (T_1), nuclear spin–spin relaxation time (T_2), and proton density (PD) [1], [2]. The quantification of multiple MR parameters improved the applicability of noninvasive MRI methods in the diagnostic assessment and follow-up of pathological conditions in different clinical and preclinical areas [3]–[8]. Quantitative MPM, however, requires multiple sequences to measure quantitative parameters, and so it may result in a very time-consuming approach [9], [10].

Recently, MR fingerprinting (MRF) has emerged as an accurate and time-efficient MPM technique delivering multiparametric maps in one-shot measurement [11]. The key idea of MRF is to apply a train of radiofrequency (RF) pulses varying in a pseudo-random way some MR acquisition parameters, for example, flip angle (FA), repetition time (TR), echo time (TE), or readout trajectory, to eventually provide unique signal evolutions, called "fingerprints", for combinations of desired tissue physical properties, such as T_1, T_2, and PD [12]. In this way, the sample properties and composition can be traced back from the analysis of the fingerprints collected in each voxel. The important advantages of MRF compared with other MPM approaches are (i) the shorter acquisition time, (ii) the simultaneous estimation of different parametric maps, and (iii) less sensitivity to instrumental constraints, including temporal delays and motion artifacts [11]–[13].

The traditional MRF maps reconstruction method is known as pattern matching [14]. Such a method compares the acquired voxel-wise fingerprints with a precomputed dictionary of synthetic MR signal evolutions generated using the Bloch equations or the extended phase graph (EPG) signal simulation model [15].

For each voxel, the dictionary entry with the best matching correlation with the fingerprint is selected and the associated parameters (e.g., T_1, T_2, and PD) are assigned to the voxel to generate the quantitative maps associated with tissue properties. One of the main drawbacks of this methodology is that the reconstruction dictionary must be sufficiently large to cover the entire range of possible tissue values and to avoid errors in reconstructed tissue maps. Furthermore, the dictionary size grows exponentially with the number of tissue parameters encoded in the simulated fingerprints [16]. This requires significant storage and computational resources, which are limiting factors for the clinical adoption of MRF techniques. Some dictionary-compression methods have been developed to reduce the dictionary size and to speed up the matching process [17]. However, these methods make additional approximations that can affect the accuracy of the reconstructed maps.

The use of machine learning (ML) and deep learning (DL) algorithms has been suggested as a strategy to overcome these reconstruction issues [18]–[20]. Among these methods, neural networks (NNs) are particularly able to approximate a nonlinear function between the inputs and the outputs that may be difficult to represent via analytical functions. Using these techniques, it is possible to define a regressive DL model that is capable of predicting, after a supervised training procedure, the parameters associated with the experimental fingerprint as input. The three main advantages of this approach are (i) the fast computation of the quantitative parameter maps, (ii) the possibility of predicting MR parameters of unknown signal evolutions (while the matching process restricts the parameters prediction to those present in the dictionary), and (iii) the capability of addressing the scalability problem of the dictionary without increasing the calculation time. Recent studies [18], [19], [21] demonstrated the feasibility of a DL-based approach for MRF. This was achieved by training the DL model using ground truth data generated via pattern matching, then applying the results to experimental data. The NN is trained to learn the tissue properties from the generated dataset, eliminating the need to memorize the whole set of fingerprints after the training procedure. These works show that, compared with the traditional matching procedure, the application of the DL model to MRF data is 300–5000 times faster [21].

Because of these advantages, many applications of the MRF framework have been reported in the clinical field. For example, in brain studies, the MRF-generated relaxometry maps were shown as a promising tool to help the diagnosis of suspicious hippocampal sclerosis in patients with mesial temporal lobe epilepsy [22], to identify inconspicuous epileptogenic lesions from patients with negative conventional MRI diagnosis [23], to study focal cerebral alterations and identify patients with frontotemporal lobe degeneration [24], and to classify Parkinson's disease subjects and their disease severity [25]. Moreover, the fast MRF map quantification makes it highly applicable to abdominal and cardiac imaging [26], [27].

However, as far as we know, the applications of MRF methodology in the preclinical field are still limited and an optimized MRF protocol for both acquisition and reconstruction phases that exploits DL techniques has not been

proposed yet. In this context, the aim of this preclinical study is to provide an optimized, DL-based MRF framework to measure T_1 and T_2 maps of the ex vivo rat brain. To pursue this goal: (i) we implemented a preclinical MRF pulse sequence with two different FA and TR pseudo-random profiles [28], [29] to generate unique T_1 and T_2 MRF signal timecourses and to investigate which pseudo-random profile delivers the most accurate T_1 and T_2 maps reconstruction via the pattern-matching method; (ii) we trained, validated, and tested two supervised DL models with all the acquired MRF images to assess which network predicts the T_1 and T_2 maps with the most accuracy; and (iii) we optimized both the MRF acquisition sequence and the DL models to deliver the least time-consuming framework to acquire MRF images and to predict accurate and reliable T_1 and T_2 maps.

In particular, this study aims to extend current DL-based methods for the analysis of MRF data in three main directions. First, we will perform a supervised training procedure on experimental data, excluding the use of MRI signal simulators. This approach ensures that the DL models are trained on real experimental inputs, which should enhance their reliability and generalizability. Second, we will present an automatic procedure to optimize simultaneously the NN architecture, the structure of the DL model, and the supervised learning algorithm. Lastly, we will compare different MRF acquisition settings and DL architectures to optimize both the acquisition and reconstruction phases. By combining these approaches, we aim to develop an advanced DL-based MRF framework that improves both the accuracy and efficiency of T_1 and T_2 map measurements in preclinical research settings.

III.1.1 Magnetic resonance imaging acquisition

MRI acquisitions were performed with a 7-T Pharmascan scanner (Bruker, Billerica, MA, USA) equipped with a circularly polarized ^1H transmit/receive volume coil. Before performing gold standard mapping and running the MRF protocols, a B_0 map was acquired and automatic shimming by means of the MAPSHIM utility was executed using all the available shims up to the Z3 order. All the acquisition protocols were based on Cartesian sampling with a linear phase-encoding order, starting from the lower edge of the k-space. No phase-encoding acceleration was applied.

The gold standard T_1 and T_2 mapping protocol was applied to one axial slice (slice thickness = 1.0 mm, FOV = 33×33 mm^2, matrix size = 128×128, with read, phase and slice offset = 0) and included an inversion recovery-spin echo (IR-SE) sequence with 10 inversion time points (TR/TE = 5000/4.59 ms, TI = 100, 200, 400, 700, 1100, 1600, 2200, 2900, 3700, 4600 ms, scan time = 106 min), and a SE sequence with 10 echo time points (TR = 5000 ms; TE = 6, 10, 20, 40, 60, 80, 100, 150, 200, and 300 ms; scan time = 106 min). The IR-SE and SE data were analyzed with an in-house fitting software developed in Matlab (MathWorks, MA, USA). In particular, T_1 maps were computed from

the following three-parameter model:

$$y(TI) = a_1 \cdot |1 - b_1 \cdot e^{-\frac{TI}{T_1}}| \tag{III.1}$$

while T_2 maps from the following exponential decay function:

$$y(TE) = a_2 \cdot e^{-\frac{TE}{T_2}} + b_2 \tag{III.2}$$

The MRF framework was implemented according to the MRF-steady-state free precession (FISP) sequence proposed by Gao et al. [28], keeping the same slice geometry set for the gold standard mapping protocol [slice thickness = 1.0 mm, FOV = 33×33 mm^2, matrix size = 128×128, with read, phase and slice offset = 0] with TE = 3.2 ms and two different FA/TR profiles. Specifically, we implemented both the 600 FA/TR points profile proposed by Gao et al. [28] and the optimized 400 FA/TR points profile proposed by Zhao et al. [29] with scan times of 30 min and 20 min respectively. For detailed visualization of these profiles, please refer to Figure III.1. As regards the MRF dictionary-based reconstruction method, T_1 and T_2 maps were computed by using the traditional dictionary simulation and the pattern matching method. In particular, the two dictionaries (Gao et al. and Zhao et al.) were generated exploiting the Extended Phase Graph (EPG) formalism15 to simulate the fingerprints for 26710 combinations of T_1 (range: 100-2500 ms) and T_2 (range: 1-200 ms).

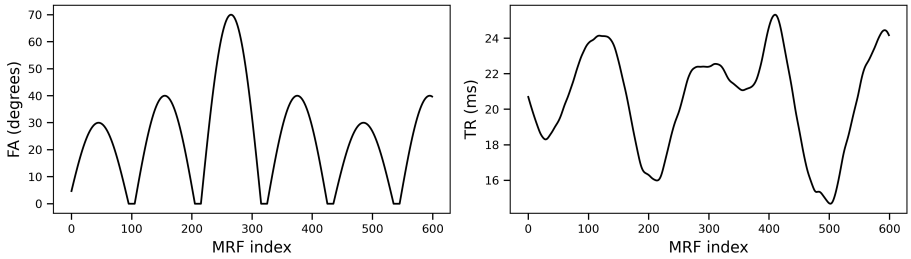
III.1.2 Input data

The dataset consisted of six slices of two ex vivo rat brain phantoms, five slices from the first phantom and one slice from the second one. For each slice of the dataset, ground truth T_1 and T_2 maps (through the IR-SE and SE sequences, respectively) and the two MRF sequences were acquired.

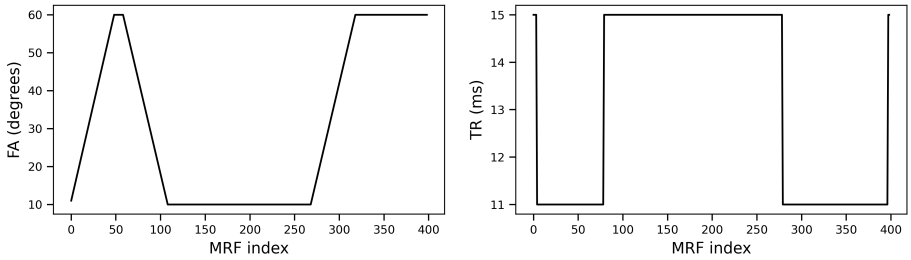
Before passing the MRF images to the DL model, the following preliminary steps were performed. First, we normalized the pixel intensities of the MRF images and ground truth T_1 and T_2 maps to the [0,1] range. Then the MRF images and ground truth T_1 and T_2 maps were transformed into collections of mono dimensional trajectories that compose the input and the expected output datasets, respectively. We excluded from the training dataset MRF signals that belong to the background and to the skull regions.

III.1.3 Deep Learning reconstruction

We defined two different architectures of the DL model: a multilayer perceptron (MLP) and a recurrent neural network (RNN). Both the models were used to perform a voxel-wise regression of the T_1 and T_2 parameters, given a MRF signal evolution trajectory as input. This means that the DL model receives, as input, the MRF signal evolution trajectory from an individual voxel and produces, as output, the estimated pair of T_1 and T_2 values corresponding to that specific voxel. A schematic representation of both the architectures is provided in Figure III.2. While MLPs are general purpose models for handling



(a) Gao et al. profile



(b) Zhao et al. profile

Figure III.1: Flip angle (FA)/repetition time (TR) profiles used for the magnetic resonance fingerprinting (MRF) acquisitions and dictionary generation. Specifically, III.1a Profile used for acquiring the Gao et al. dataset of 600 time points; III.1b Profile used for acquiring the Zhao et al. dataset of 400 time points.

nonspecific input data, RNNs are intended to deal with time-series signals to solve temporal problems. Specifically, the MLP was composed of a sequence of fully connected layers. The input layer consists of a number of nodes equal to the MRF sequence length with a sigmoid activation function, while the output layer consists of two nodes for the prediction of T_1 and T_2 values. The RNN was composed of a long-short term memory (LSTM) block with a hyperbolic tangent activation function and a sigmoid activation for the recurrent step, followed by a fully connected layer with two nodes for the output predictions. Because the LSTM block works better with short input sequences [30], we reshape the one-dimensional MRF sequence into multiple parallel time series, each 20 time points long. The appropriate activation function of the output layer of each of the models was found using the hyperparameter optimization algorithm.

The supervised training procedure of both models was performed by minimizing the mean squared error (MSE) loss function and using the mean absolute error (MAE) as an additional metric to monitor the process. Early stopping was performed by monitoring the validation error: the training procedure was stopped when the validation error did not improve within a maximum number of epochs. This maximum number of steps is called early

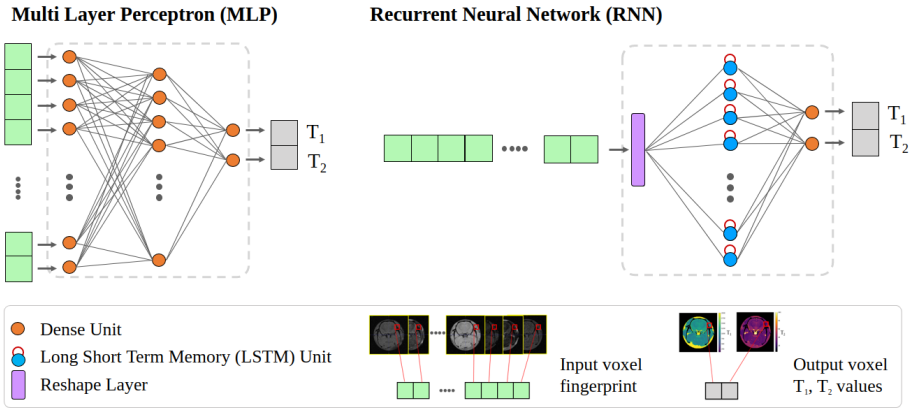


Figure III.2: Schematic representation of the two architectures used: a multilayer perceptron (MLP) on the left and a recurrent neural network (RNN) composed of a long-short term memory (LSTM) block on the right. Both models receive the magnetic resonance fingerprinting trajectory of a single voxel as input and output the corresponding T_1 and T_2 values for that voxel. The activation functions, number of hidden dense layers, and number of dense units for the MLP, as well as the number of LSTM units for the RNN, were optimized through the automatic hyperparameter tuning procedure.

stopping patience, and it was set equal to a percentage of the total number of training epochs. To train, test, and evaluate the generalization ability of the proposed method, we performed a cross-validation procedure among different slices of the two phantoms. Specifically, we trained the model on five slices of the first phantom, randomly partitioning the input dataset into two separate subsets (80%/20%) for the training and validation phases, respectively. We used the slice of the second phantom excluded from the training process and from the hyperparameter tuning as an independent test set.

Code has been written in Python 3.8 leveraging on the tensorflow 2.8 machine learning platform [31].

III.1.4 Hyperparameter optimization

To find the best configuration of the model for the MRF problem and for the available data, we performed the hyperparameter optimization through the Hyperopt library [32]. Using the Hyperopt framework, many combinations among a selection of parameters are scanned in a semiautomatic way using the tree-structured Parzen estimator (TPE) algorithm to find the best configuration of the model. In particular, the TPE algorithm performs a Bayesian scan of the search space by sampling more densely the most promising regions of the space and, as a consequence, learning from the tuning history. To optimize the

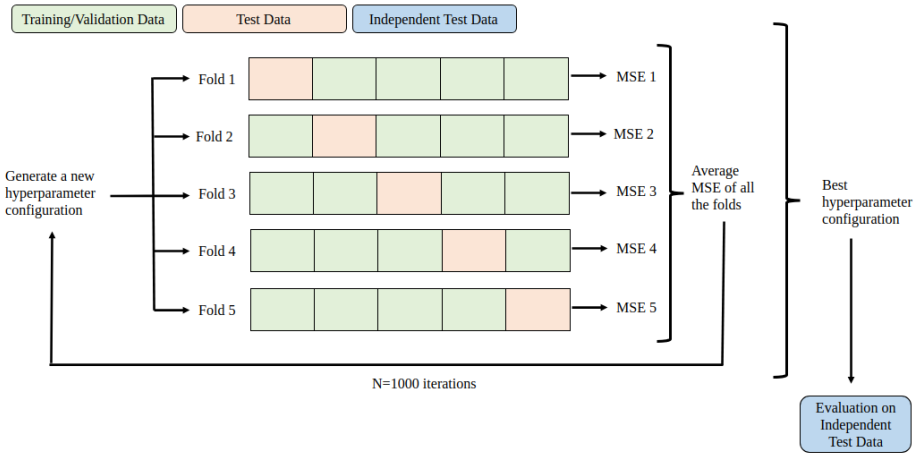


Figure III.3: Schematic representation of k-fold cross-validation for hyperparameter optimization. The cross-validation was performed on the five slices of the first phantom. The sixth slice of the second phantom was not included in the optimization process and was used as an independent test set. MSE, mean squared error.

hyperparameters configuration we applied the k-folding method (k = 5) to five slices of the first phantom. The model was trained and validated on k–1 slices (80% training, 20% validation) and it was tested on the k-th slice. For each hyperparameter combination, the k-folding technique was repeated five times, giving an opportunity to each fold to be the held-out test set, and the results were averaged to obtain a global evaluation of the hyperparameter combination. The whole process was repeated for 1000 hyperparameter combinations and returns the configuration of the search space, which minimizes the tuning loss. A schematic representation of the k-fold cross-validation strategy is provided in Figure III.3. Specifically, we used the average MSE to evaluate the tuning procedure, defined as:

$$L = \frac{1}{5}\sum_{k=1}^{5} MSE_k. \tag{III.3}$$

The hyperparameters that were tuned include the architecture of the model (i.e., MLP/RNN), some NN parameters (i.e., the number of hidden layers, the size of each layers, the initialization function and the activation function of the final layer), as well as some parameters of the supervised learning algorithm (i.e., the optimizer, the initial learning rate, the maximum number of epochs, and the early stopping patience).

III.1.5 Evaluation strategy

oth the dictionary-based reconstruction method and the DL method were evaluated using the percentage relative error (PRE) and the structural similarity index (SSIM).

The PRE quantifies the agreement between the reconstructed map $\mathbf{y}$ and the reference one $\hat{\mathbf{y}}$ by computing the $|\hat{y}_i - y_i|/\hat{y}_i \cdot 100\%$, where y_i and $\hat{y}_i$ are the i-th voxel of $\mathbf{y}$ and of $\hat{\mathbf{y}}$, respectively. To assess the overall agreement of the reconstructed maps, we studied the distribution of the PRE over the entire maps, excluding the areas associated with the background and skull, and over some anatomical districts of interest. Specifically, we selected the cortex and corpus callosum to provide examples of gray and white matter regions, respectively. We used the SSIM to measure the goodness of the reconstruction by comparing the statistical distributions of the two arrays of pixels [33]. Unlike the PRE, the SSIM measures the difference between two images by assessing perceptual characteristics. The SSIM can assume values between 0 and 1: if it is equal to 1 then there is perfect structural similarity between the image and the reference image, while a value of 0 indicates that the images are completely different from each other.

III.1.6 Sequence compression methods

The aim of this phase of the work was to determine the optimal design of the MRF sequence that allows for good estimation of the parametric maps with a shorter acquisition time. We examined the following aspects of the scanning scheme: the minimum number of sequence images necessary to obtain accurate quantitative maps and the minimum sampling percentage of the Cartesian k-space that guarantees good reconstruction performances.

As regards the first point, we evaluated the reconstruction performances of the algorithm with different acquisition sequence lengths, obtained by selecting several subsets of images, starting from the beginning of the MRF acquisition. In particular, we tested different sequence lengths, ranging from the full-length experiments to 20 time points sequences. For the second aspect, to undersample the k-space, we proceeded in the following way. First, we transformed all the MRF images in the frequency domain by using the 2D Fourier transform. Then we undersampled the k-space data by keeping only the central phase-encoding lines of the k-space and applying zero filling to the peripheral lines. We tested 10 different subsampling percentages of the k-space, ranging from 100% to 10% of the total number of phase-encoding lines. By applying the inverse Fourier transform of the undersampled k-space data, we obtained the MRI images. To determine both the minimum number of images of the MRF sequence and the minimum k-space sampling percentage to be used, we applied the Elbow method proposed by Satopaa et al [34]. Specifically, we plotted the SSIM and the mean PRE as a function of the two parameters and we identified the optimal point of the MRF sequence as the point of maximum curvature on the graph using the kneed Python library.

III.2 Results

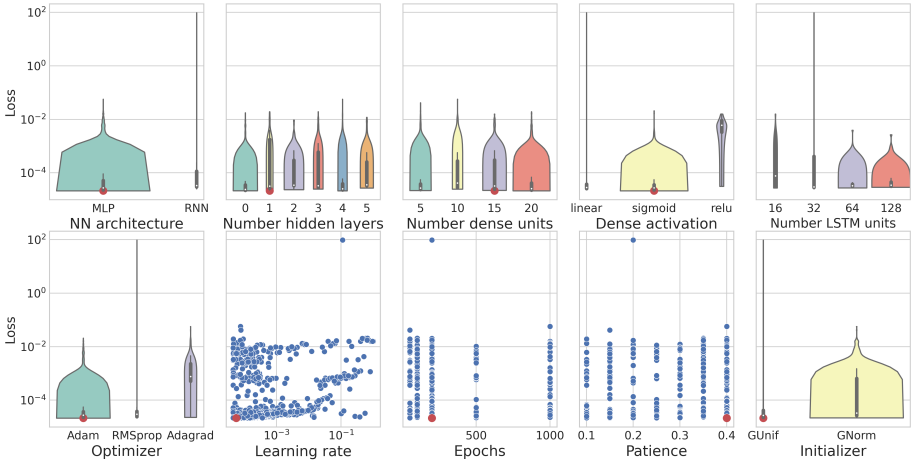
III.2.1 Hyperparameter optimization results

The optimization procedure was performed separately on each of the Gao et al. and Zhao et al. datasets to obtain the best configuration for both acquisition schemes. We performed 1000 iterations of the optimization algorithm and selected the configuration that minimized the tuning loss, L.

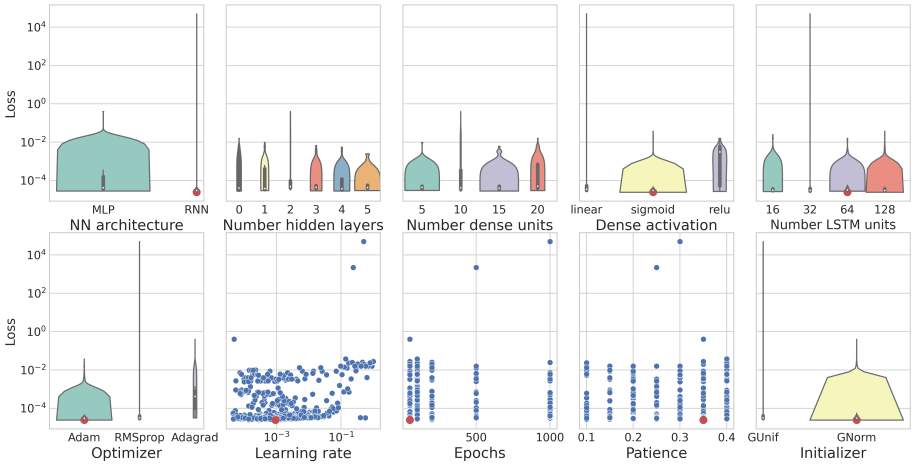
The best configurations for the Gao et al. and Zhao et al. datasets given by optimization scan are listed in Table III.1. The best architecture of the NN for the Gao et al. dataset is defined by MLP with a single hidden layer composed of 15 dense units, while the NN's optimal architecture for the Zhao et al. dataset consists of a RNN composed of a LSTM block with 64 units. The sigmoid activation function is preferred for the output layer, as well as the Glorot [35] initialization function, to set the initial values of the layer's weights (specifically, the Glorot uniform for the Gao et al. dataset and the Glorot normal for the Zhao et al. dataset). Regarding the hyperparameters of the training phase, the Adam optimizer [36] achieved the best result, with an initial learning rate of the order of 10^{-5}. Finally, concerning the maximum number of epochs and the early stopping patience, no large differences were observed in the behavior of the loss function as a function of the scanned values. Figure III.4 presents the results of the hyperparameters tuning process for the III.4a Gao et al. and III.4b Zhao et al. datasets.

Hyperparameters	**Gao et al. value**	**Zhao et al. value**
Architecture	MLP	RNN
Number of hidden layers	1	/
Size dense layers	15	/
Size LSTM block	/	64
Dense activation function	Sigmoid	Sigmoid
Initialization functions	Glorot uniform	Glorot normal
Optimizer	Adam	Adam
Initial learning rate	5.7×10^{-5}	9.3×10^{-5}
Maximum number of epochs	200	50
Early stopping patience	40%	35%

Table III.1: Best configuration of the parameters obtained using the ptimization process. LSTM, long-short term memory; MLP, multilayer perceptron; RNN, recurrent neural network.



(a) Gao et al. dataset



(b) Zhao et al. dataset

Figure III.4: Results of the optimization process for the III.4a Gao et al. dataset and III.4b Zhao et al. dataset as a function of each hyperparameter tuned. We use violin plots and scatter plots to graphically represent the distribution of the loss (mean squared error) values. Red dots highlight the best configuration. LSTM, long-short term memory; MLP, multilayer perceptron; NN, neural network; RNN, recurrent neural network.

III.2.2 Phantom results

We trained the best configuration of the model on both the Gao et al. and Zhao et al. datasets separately, following the cross-validation strategy described in Section III.1.3 and we evaluated the predictive performances of the DL model by applying it to the independent test set. Gold standard T_1 and T_2 maps, the dictionary-based reconstructed maps, and the DL-reconstructed maps, are presented in Figure III.5 for both the Gao et al. and Zhao et al. datasets. First, we can qualitatively see that the estimated parametric maps of both methods have correctly reconstructed the anatomical structures of the brain, preserving the visual contrast between different tissues. We can also observe that the range of T_1 and T_2 maps estimated with all the MRF techniques are comparable with the reference T_1 and T_2 maps, validating the MRF acquisition designs proposed by Gao et al. and Zhao et al. [28], [29]

Figure III.5 shows the error map associated with each parameter computed voxel-wise as the PRE. We observe good agreement between the reference and each of the reconstructed maps for most of the anatomical regions of the phantom. In particular, good results were achieved in the regions of the brain. Larger discrepancies are visible in the volume surrounding the phantom and near to the skull, probably because of the imperfections in the RF excitation field, as already discussed in Gao et al. [28] These errors are more evident in the T_2 maps compared with the T_1 maps. We summarized the distributions of the error maps in the boxplots of Figure III.6. We can observe that the agreement of the reconstructed maps with the ground truth ones is better for the DL method compared with the traditional dictionary-based one, both in terms of mean value and dispersion of the errors. Figure III.7 presents the distributions of the PREs of the two regions of interest (ROIs) marked in the upper left corner. We can see that the PRE distributions for the voxels of the two ROIs reflect the PRE distributions obtained over the entire image (Figure III.6). However, the mean values and the standard deviations of the two ROIs are smaller than those obtained for the complete image. Overall, T_1 and T_2 estimates for white matter are slightly better than those for gray matter. Considering the SSIM of the T_1 map, the DL method reaches a value of 0.91 for the Gao et al. dataset and 0.92for the Zhao et al. dataset; while for the T_2 map, the SSIM is equal to 0.88 for the Gao et al. dataset and 0.89 for the Zhao et al. dataset. Concerning the dictionary-based reconstruction method, the T_1 map achieves a SSIM of 0.91 for both the Gao et al. and the Zhao et al. dataset; considering the T_2 map, the model reaches a SSIM equal to 0.73 for the Gao et al. dataset and 0.81 for the Zhao et al. dataset.

III.2.3 Length of the MRF sequence results

We evaluated the reconstruction performances of the DL algorithm and of the dictionary-based one with different acquisition sequence lengths for both datasets (Figure III.8). As can be observed from Figure III.8, good agreement between the DL-reconstructed and true maps was obtained for a sequence length of at

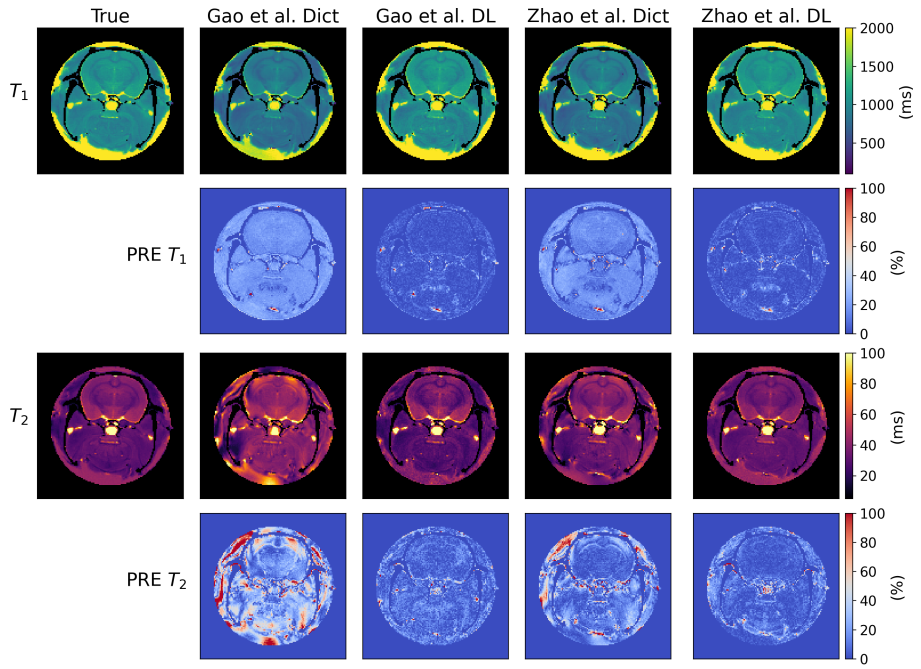


Figure III.5: Comparison between the T_1 and T_2 true maps (first column), those reconstructed with the traditional dictionary-based method (Dict) and those generated with the deep learning (DL) method. The second and third columns represent the results for the Gao et al. dataset, while the fourth and fifth columns represent the results for the Zhao et al. dataset. Regions associated with the background and the skull are set to the minimum value of each map. Error maps are computed voxel-wise as the percentage relative error (PRE).

least 100 time points for both T_1 and T_2. We repeated the same simulation for the dictionary-based method and found that a MRF sequence of at least 480 time points is necessary for a reliable reconstruction of T_1 maps and of 490 time points for T_2 maps. In terms of the acquisition time, reducing the number of MRF time points from 600 of the complete Gao et al. sequence to 100 of the compressed sequence means speeding up the acquisition phase by an acceleration factor of about 2.2.

Regarding the Zhao et al. MRF sequence, the results show that for the DL system, a good reconstruction of T_1 maps is achieved with at least 100 time points, and with at least 60 time points for T_2 maps. For the dictionary-based method, 290 and 330 images are necessary for a good reconstruction of T_1 and T_2 maps, respectively. By using 100 MRF images of the initial 400 means accelerating the acquisition time by a factor of about 1.7.

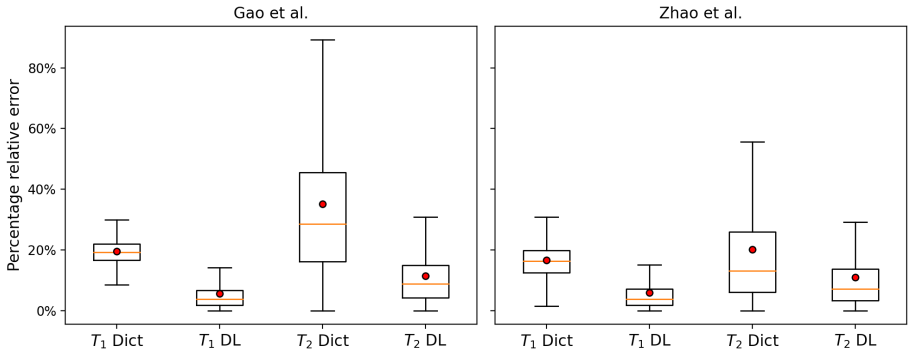


Figure III.6: Boxplots of the percentage relative errors of the T_1 and T_2 maps reconstructed with the deep learning (DL) methodology and with the dictionary-based (Dict) method for the Gao et al. and Zhao et al. datasets. Outlier values were excluded from the plot, as well as background and skull PREs. Red dots represent the mean values of the distributions.

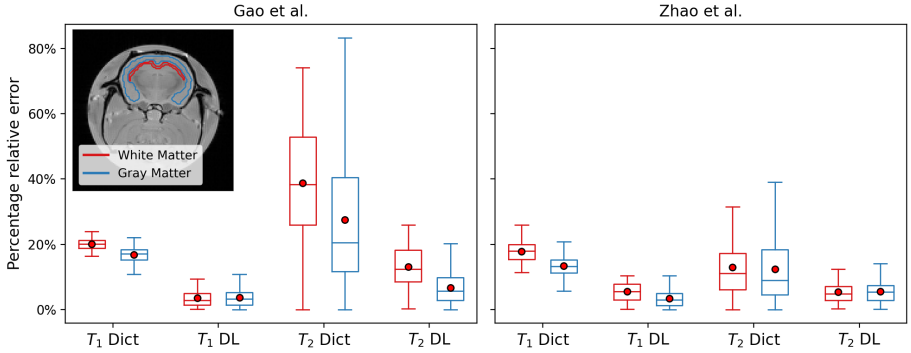


Figure III.7: Boxplots of the percentage relative errors of T_1 and T_2 values computed on the two regions of interest (ROIs) highlighted in the upper left image. We use the red color to represent the ROI denoted with white matter that corresponds to the corpus callosum, while blue represents gray matter, which corresponds to the cortical area. The graph shows the PRE distributions obtained with the deep learning (DL) methodology and with the dictionary-based (Dict) method for the Gao et al. and Zhao et al. datasets. Outlier values were excluded from the plot. Red dots represent the mean values of the distributions.

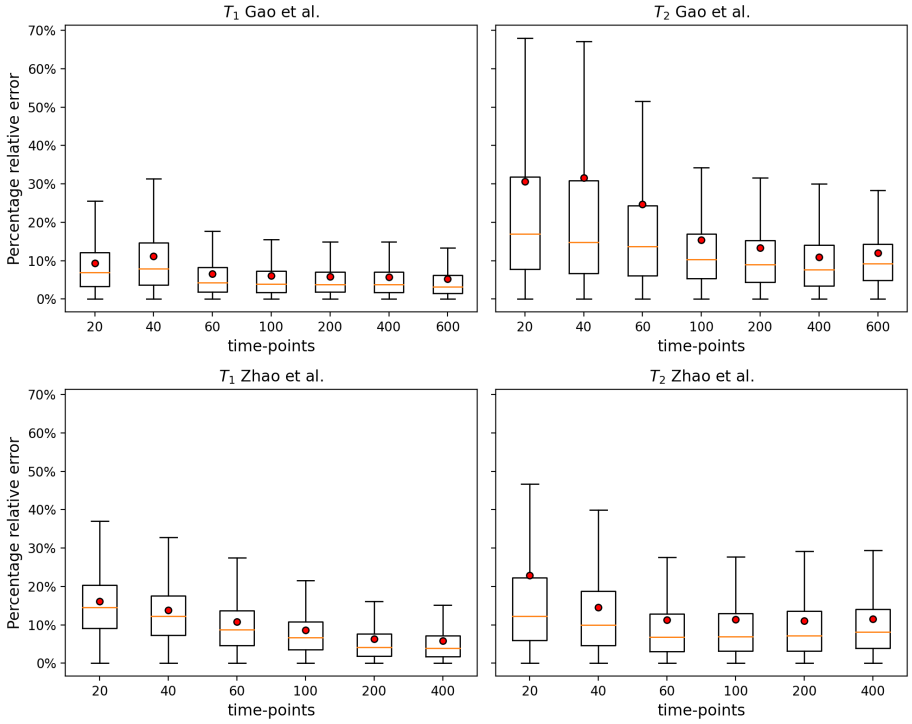


Figure III.8: Boxplots of the percentage relative errors of T_1 and T_2 test maps as a function of different sequence lengths of 20, 40, 60, 100, 200, 400, and 600 time points for the Gao et al. acquisition schedule (first row) and of 20, 40, 60, 100, 200, and 400 time points for the Zhao et al. acquisition schedule (second row). Outlier values were excluded from the plot. Red dots represent the mean values of the distributions.

III.2.4 K-space undersampling results

We tested the reconstruction accuracy of the DL model and of the traditional dictionary-based method for different k-space Cartesian sampling percentages considering the Gao et al. and Zhao et al. full-length experiments (Figure III.9). We performed different scans for the two datasets by varying the k-space sampling percentage from 100% to 10%.

The results for the Gao et al. and Zhao et al. sequences are presented in Figure III.9 for the DL reconstruction method. We can observe that the T_1 and T_2 maps are consistent with a minimum k-space percentage of 40% for both the Gao et al. and Zhao et al. sequences. Considering the traditional reconstruction method, we found that a minimum sampling percentage of 50% is required to estimate accurate maps for both the Gao et al. (40% T_1 and 50% T_2) and the Zhao et al. (40% T_1 and 40% T_2) datasets. For the two acquisition schemes

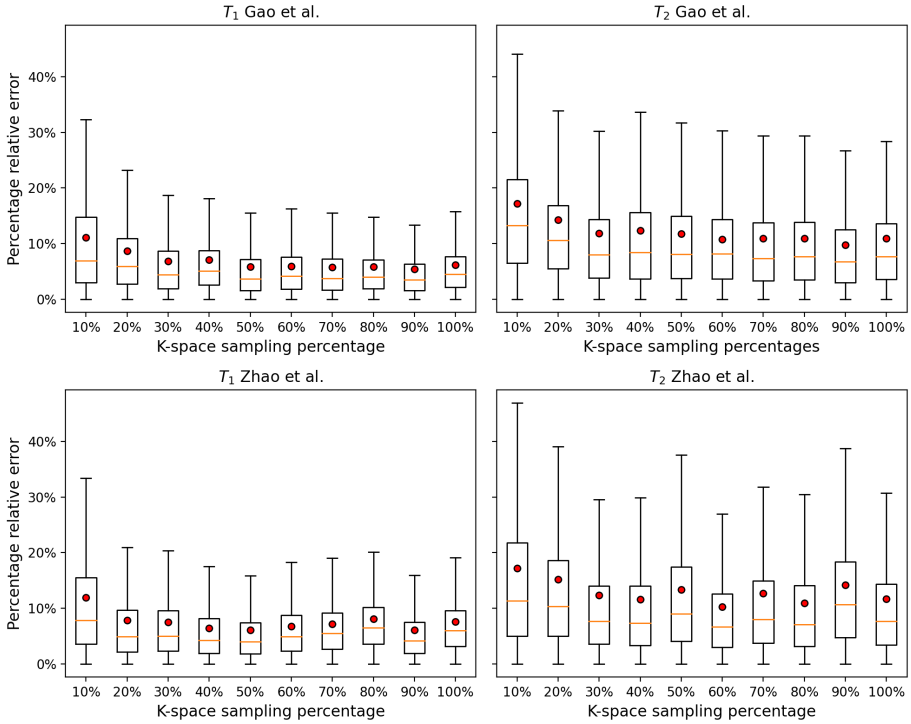


Figure III.9: Boxplots of the percentage relative errors of T_1 and T_2 test maps as a function of the k-space sampling percentage. For both datasets, we tested 10 sampling percentages ranging from 10% to 100% (100% corresponds to full k-space sampling). Outlier values were excluded from the plot. Red dots represent the mean values of the distributions.

and reconstruction methods, we observed that below these thresholds the loss of spatial resolution because of k-space subsampling becomes evident. Because the acquisition time is directly proportional to the number of k-space phase-encoding lines sampled, the scanning time decreases, reducing the k-space coverage. For the Gao et al. sequence, by reducing the k-space sampling percentage from 100% to 30%, we reduced the acquisition time by an acceleration factor equal to 3.3. Considering the Zhao et al. sequence, by sampling 40% of the k-space lines, the acquisition time was reduced by a factor equal to 2.5.

We tested the reconstruction accuracy of the DL model and of the traditional dictionary-based method for different k-space Cartesian sampling percentages considering the Gao and Zhao full-length experiments, Figure III.9. We performed different scans for the two dataset by varying the k-space sampling percentage from 100% to 10%.

III.2.5 Computation time

All simulations were performed on a 2.60 GHz Intel Core i7 central processing unit with 12 cores and 16 GB RAM. The DL model training process, realized on the dataset of $128 \times 128 \times 5$ examples, took around 5 min and 38.4 s for the Gao et al. sequence and 8 min and 39.6 s for the Zhao et al. sequence. T_1 and T_2 map predictions, each consisting of 128×128 pixels, required about 0.23 s for the Gao et al. dataset and 0.67 s for the Zhao et al. dataset. Considering the dictionary-based reconstruction method, the dictionary generation took around 57 and 34 min for the Gao et al. and the Zhao et al. datasets, respectively. The map estimation process took 26.26 s for the Gao et al. dataset and 24.58 s for the Zhao et al. dataset. The DL method enables speeding up the reconstruction process by a factor of 114 for the Gao et al. sequence and by a factor of 37 for the Zhao et al. sequence.

III.2.6 Comparison with other DL methods

The proposed method was compared with other DL techniques to evaluate the reconstruction performances using the same dataset of preclinical MRF images. We considered the following DL architectures: convolutional neural network 1D (CNN 1D), convolutional neural network 2D (CNN 2D), convolutional encoder-decoder neural network (CED), 2D Unet, and 3D Unet. These five models are some of the state-of-the-art methods in the field of DL techniques for image analysis. The specific architecture of each of these NNs is described in Appendix III.A of this document. Table 2 shows the results of the compared techniques in terms of PRE and SSIM of the reconstructed maps of the independent test set. We can see that our method reached the lowest reconstruction error considering both metrics and outperformed all the other considered models.

III.3 Discussion and conclusions

In the current study, we propose a DL framework for the reconstruction of T_1 and T_2 maps acquired with two different MRF schemes. The image analysis system is composed of a DL NN that receives the MRF signal evolution trajectory voxel-wise as input and outputs the estimated values of the two T_1 and T_2 relaxation maps, and of a hyperparameter optimization algorithm to automatically choose the best structure of the DL model. The NN was trained using two sets of MRF data of ex vivo rat brain phantoms acquired with a 7-T preclinical scanner. Both the datasets were randomly split into two separate subsets for training and validation phases then tested on an independent dataset. We compared the reconstruction performances of the DL technique with those achieved from the traditional dictionary-based method, evaluating the PRE and the SSIM. The DL reconstruction method achieved better results compared with the dictionary-based one for both the Gao et al. and the Zhao et al. datasets. In particular, we proved that our DL method decreased the mean PRE by a factor of 3 for T_1 and

Model		PRE mean (%)		PRE SD (%)		SSIM	
		T_1	T_2	T_1	T_2	T_1	T_2
CED	Gao et al.	18.8	31.85	18.76	29.46	0.47	0.46
	Zhao et al.	17.58	27.7	19.33	30.84	0.51	0.49
2D Unet	Gao et al.	15.18	29.56	17.45	31.0	0.51	0.49
	Zhao et al.	16.38	20.28	18.06	21.46	0.52	0.50
3D Unet	Gao et al.	31.26	84.91	19.23	3.99	0.48	0.33
	Zhao et al.	41.47	84.91	27.42	3.99	0.51	0.33
2D CNN	Gao et al.	6.79	11.73	8.55	17.63	0.88	0.84
	Zhao et al.	6.59	12.72	9.22	15.63	0.88	0.86
1D CNN	Gao et al.	7.12	11.99	9.76	17.02	0.86	0.83
	Zhao et al.	6.29	11.73	9.40	15.95	0.90	0.86
Proposed method	Gao et al.	**5.61**	**11.36**	**8.25**	**12.99**	**0.92**	**0.88**
	Zhao et al.	**5.92**	**10.91**	**8.14**	**14.95**	**0.91**	**0.89**

Table III.2: Comparison with some of the state-of-the-art models for image analysis for the Gao et al. and Zhao et al. datasets. All metrics were evaluated on the independent test set. CED, convolutional encoder-decoder neural network; CNN, convolutional neural network; PRE, percentage relative error; SD, standard deviation; SSIM, structural similarity index. The best result in each column is highlighted in bold text.

by a factor of 2 for T_2. In terms of SSIM, both reconstruction methods achieved a SSIM equal to 0.91 for T_1 maps, while for T_2 maps, the DL method increased the SSIM from 0.77 to 0.89. Besides this improvement of the estimation performances, we also demonstrated that the proposed method significantly accelerates the computational time of the reconstruction process.

We extended current DL-based methods for the analysis of MRF data in two main directions: (i) we performed a supervised training procedure on experimental data excluding the use of theoretical MRI signal simulators; and (ii) we proposed an automatic procedure to automatically optimize the structure of the NN and the training process. As regards aspect (i), by excluding the MRI signal simulator from the reconstruction process, we exploit the flexibility of the NN, which is also able to recognize data stochasticities related to the acquisition process. This phenomenon is evident in Figure III.5 if we compare the T_2 maps of both the acquisition schemes reconstructed with the dictionary-based method with those generated by the DL method. In the first case, the T_2 maps present artifacts in the area surrounding the phantom, probably because of imperfections in the RF excitation field, while in the second case, these artifacts are significantly reduced, because the network has learned to recognize and discard them. Furthermore, this approach is particularly convenient for quantitative parameters whose MRI signal cannot be described with well-studied models, and for which, therefore, the dictionary cannot be generated by any

theoretical simulator.

As regards point (ii), we implemented a hyperparameter optimization strategy to select the best combination of the DL parameters. The key aspect of this optimization process is to include all the parameters in the fit, allowing the simultaneous optimization of the NN architecture, the structure of the DL model, and the supervised learning algorithm. Because the behavior of a DL model depends on configuration settings that are often interconnected to each other, for a correct hyperparameter tuning it is essential to evaluate the combined effect of all the parameters at the same time. This methodology is useful because it allows you to choose the optimal hyperparameter setup in a semiautomatic way with a systematic pipeline, avoiding performing such searches by hand. In particular, we used the TPE optimization algorithm, which performs a Bayesian scan of the search space by sampling more densely the most promising regions of the space to find the best configuration that can optimally solve the MRF problem. Using this strategy, we compared two different architectures of the NN by testing the MLP and a recurrent NN composed of an LSTM block. The results showed that for the Gao et al. dataset the MLP model outperformed the RNN, both in terms of lower reconstruction errors and stability of the performances, while for the Zhao et al. dataset, the RNN was the best architecture of the NN. Architectures with convolutional layers achieved lower accuracy. This can be explained by considering the fact that the values of FA and TR defined in Gao et al. and Zhao et al. are not ordered. In future studies, it would be valuable to explore the potential of different architectures, like the self-attention model proposed by Hong et al. [37]

The MRF acquisition scheme was analyzed to determine the best design of the MRF sequence necessary to obtain accurate quantitative maps with a reduced acquisition time. We evaluated the performances of both the reconstruction approaches with different numbers of MRF images and with different k-space sampling percentages. The results of Figure III.8 demonstrated that, for the Gao et al. dataset, the number of MRF images can be reduced to 100 images for good reconstruction of both the T_1 and T_2 maps, compared with the 300 images necessary for the dictionary-based reconstruction. As regards the Zhao et al. dataset, Figure 8 shows that at least 100 time points are required for a good reconstruction of both the T_1 and T_2 maps, in contrast to the dictionary-based method, where 400 time points are required. Considering the k-space sampling, we showed that for both of the MRF sequences, 40% of the k-space sampling is sufficient for a reliable estimation of the T_1 and T_2 maps with the DL method and 50% of the k-space sampling for the dictionary-based method. Therefore, the DL approach enables not only speeding up the reconstruction phase avoiding the pattern-matching process, but also reduces the time duration of the MRF acquisition, enabling the use of a smaller number of time points as well as a minor sampling percentage of the k-space. These results are particularly significant for simplifying in vivo applications of MRF sequences in the field of preclinical imaging.

We extensively compared two different MRF acquisition schemes, considering the sequence designs proposed in the work of Gao et al. [28] and Zhao et al. [29]

Regarding the DL reconstruction, the graphs in Figure III.8 show that the model achieves similar performances on both datasets, demonstrating the stability of the methodology to changes in the acquisition parameter scheme. On the other hand, if we consider the dictionary-based reconstruction method, some differences in the maps calculated from the two MRF sequences are evident. As already discussed in the work of Gao et al. [28], T_2 maps present some inhomogeneities, probably because of imperfections in the RF excitation field that may lead to errors in the matching process between the acquired voxel fingerprints and the dictionary ones. By comparing the T_2 error maps of Figure III.5, we can observe that these inhomogeneities are significantly reduced by the Zhao et al. acquisition scheme compared with the Gao et al. scheme. These results suggest that the TR and FA profiles used in the Zhao et al. scheme generate MRF signal evolution trajectories that are less susceptible to hardware imperfections and that therefore make the pattern-matching algorithm more robust. Furthermore, these inhomogeneities are further reduced with the DL-based reconstruction method. Thus, our study demonstrates that the combination of the FA/TR profile proposed by Zhao et al. and our DL-based method for map reconstruction effectively minimizes RF excitation field inhomogeneities, overcoming the limitations highlighted in the work of Gao et al. [28]

Although satisfactory results have been achieved, this work can be improved in different ways. First, our method is focused on estimating T_1 and T_2 maps of ex vivo brains of healthy rats. To enhance the robustness and generalizability of our findings, it would be relevant to increase the sample size of our study and extend the proposed methodology to other anatomical regions and pathological tissues. A good strategy would be to include samples with heterogeneous relaxometric properties in the training dataset so that the model is able to predict T_1 and T_2 over a wider range of values with the same accuracy. Furthermore, the results obtained in the current work suggest that the two proposed MRF sequences combined with the DL reconstruction method could be reasonably applied to in vivo imaging of rodents. In fact, the acquisition time of both the MRF sequences makes quantitative mapping compatible with the duration of an in vivo acquisition protocol. However, it is worth emphasizing that, for a proper translation of the experimental approach in vivo, it is necessary to make some adjustments to the entire MRI acquisition protocol, because animal health also becomes a priority when setting the experimental conditions and parameters. In particular, a careful optimization is required to shorten the typical duration of several hours of SE and IR-SE acquisitions of relaxation times maps, which serve as the gold standard necessary to validate the MRF ones. Once these adjustments and optimizations are made, the proposed method used ex vivo can be implemented in vivo, allowing for quantitative mapping of T_1 and T_2 relaxation times in future animal studies. Other potential improvements to further decrease the duration of the MRF sequence involve considering the utilization of an echo-planar imaging (EPI)-based sequence in the preclinical field, as demonstrated in clinical studies [38]–[40]. EPI sequences employ a more efficient k-space sampling strategy, enabling a significant reduction in acquisition time. This approach could be explored to achieve more time-efficient MRF acquisitions, also in the

preclinical setting. Moreover, additional MRI parameters can be included in the quantified maps, such as diffusion [41], perfusion [42], or magnetization transfer maps [43]. For example, in the work of Yu et al. [41], the authors formulated a MRF framework to simultaneously measure quantitative maps of T_1 and T_2, as well as the apparent diffusion coefficient, demonstrating the potential of MRF to also quantify diffusion maps.

Appendix III.A
Additional descriptions of DL methods compared

Convolutional Encoder-Decoder Neural Network (CED)

A Convolutional Encoder-Decoder Neural Network (CED) [44] was used to perform a 2D regression of T_1 and T_2 maps taking as input $16 \times 16 \times T$ patches of MRF images (where T is the number of time-points of the MRF sequence) and returning $16 \times 16 \times 2$ patches of T_1 and T_2 maps as output. The model consisted of an encoder for feature extraction followed by a decoder to reconstruct the desired matrix size. The encoder was composed of three down-sampling blocks each consisting of two consecutive convolutional layers with kernel size equal to $3 \times 3 \times 3$ and rectified linear unit (ReLU) activation function followed by a Maxpooling layer with pool size equal to $2 \times 2 \times x$. The pool size of the time channel x was set such that the output of the third down-sampling block has size 2 on the third dimension. The architecture of the decoder was analogous to the encoder except for the Maxpooling layer which was replaced with an Upsampling layer with upsampling factor equal to $2 \times 2 \times 1$. Encoder and decoder are connected together by a middle block composed by two convolutional layers.

2D Unet

2D Unet [45] was used to perform a 2D regression of T_1 and T_2 maps taking as input $16 \times 16 \times T$ patches of MRF images and returning as output $16 \times 16 \times 2$ patches of T_1 and T_2 maps. The model consisted of an encoder followed by a decoder connected together through skip connections. The encoder was composed by three down-sampling blocks each consisting of two consecutive convolutional layers with kernel size equal to 3×3 and ReLU activation function followed by a Maxpooling layer with pool size equal to 2×2. The first layer of the encoder treats the input patch as a multi-channel image, performing 2D convolutions for each channel individually and then combining together to form a single-channel image. The architecture of the decoder was analogous to the encoder except for the Maxpooling layer which was replaced with an Upsampling layer with upsampling factor equal to 2×2. A middle block composed by two convolutional layers connected together encoder and decoder.

3D Unet

Two 3D Unets [45] were used to perform two 3D regressions of the T_1 and T_2 maps separately, taking as input $16 \times 16 \times T$ MRF image patches and returning the corresponding map patches as output. The model consisted of an encoder followed by a decoder connected together through skip connections. To connect each down-sampling block to the corresponding upsampling block, we repeated the ground truth patch T times along the third axis to form a $16 \times 16 \times T$ patch. The encoder was composed by three down-sampling blocks each consisting of two consecutive convolutional layers with kernel size equal to $3 \times 3 \times 3$ and ReLU activation function followed by a Maxpooling layer with pool size equal to $2 \times 2 \times 2$. The architecture of the decoder was analogous to the encoder except for the Maxpooling layer which was replaced with an Upsampling layer with upsampling factor equal to $2 \times 2 \times 2$. A middle block composed by two convolutional layers connected together encoder and decoder. The predicted $16 \times 16 \times T$ patches are finally averaged along the third axis to form 16×16 patches of T_1 or T_2 maps.

2D Convolutional Neural Network (2D CNN)

A 2D Convolutional Neural Network (2D CNN) was used to perform a pixel-wise regression of T_1 and T_2 maps. The CNN takes as input the MRF time trajectory of a single pixel suitably reshaped to form a 2D array and outputs T_1 and T_2 values of that pixel. The CNN was made up of a series of two convolutional blocks followed by a flattening layer and three fully connected layers for output predictions. Each convolutional block consisted of a 2D convolutional layer with kernel size equal to 3×1 and ReLu activation, Batch Normalization and a MaxPooling layer with pool size equal to 2×1.

1D Convolutional Neural Network (1D CNN)

A 1D Convolutional Neural Network (1D CNN) was used to perform a pixel-wise regression of T_1 and T_2 maps. The CNN takes as input the 1D MRF time trajectory of a single pixel (mono-dimensional array of length T) and outputs T_1 and T_2 values of that pixel. The CNN was made up of a series of two convolutional blocks followed by a flattening layer and three fully connected layers for output predictions. Each convolutional block consisted of a 1D convolutional layer with kernel size equal to 3 and ReLu activation, Batch Normalization and a MaxPooling layer with pool size equal to 2.

Conflicts of interest

None of the authors report conflict of interest related to this study.

Funding

This research was funded by the INFN *next_AIM* project.

Acknowledgements

R.F.C., L.B, D.C., S.F., M.M., A.L. thank the INFN-CSN5 research project *next_AIM (Artificial Intelligence in Medicine: next steps)*, https://www.pi.infn.it/aim/. A.L. thanks the national project "MIUR, Dipartimenti di Eccellenza Program (2018-2022), project F11I18000680001". M. P. thanks ALSA, grant nr. 20-IIA-525.

References

[1] Cooper, G., Hirsch, S., Scheel, M., *et al.*, "Quantitative multi-parameter mapping optimized for the clinical routine," *Frontiers in Neuroscience*, vol. 14, p. 611 194, 2020.

[2] Weiskopf, N., Suckling, J., Williams, G., *et al.*, "Quantitative multi-parameter mapping of R_1, PD*, MT, and R_2^* at 3T: A multi-center validation," *Frontiers in Neuroscience*, vol. 7, p. 95, 2013.

[3] Granziera, C., Wuerfel, J., Barkhof, F., *et al.*, "Quantitative magnetic resonance imaging towards clinical application in multiple sclerosis," *Brain*, vol. 144, no. 5, pp. 1296–1311, 2021.

[4] Reitz, S. C., Hof, S.-M., Fleischer, V., *et al.*, "Multi-parametric quantitative MRI of normal appearing white matter in multiple sclerosis, and the effect of disease activity on T_1," *Brain Imaging and Behavior*, vol. 11, pp. 744–753, 2017.

[5] Syed, A. K., Whisenant, J. G., Barnes, S. L., Sorace, A. G., and Yankeelov, T. E., "Multiparametric analysis of longitudinal quantitative MRI data to identify distinct tumor habitats in preclinical models of breast cancer," *Cancers*, vol. 12, no. 6, p. 1682, 2020.

[6] Salerno, M. and Kramer, C. M., "Advances in parametric mapping with CMR imaging," *JACC: Cardiovascular Imaging*, vol. 6, no. 7, pp. 806–822, 2013.

[7] Iles, L., Pfluger, H., Phrommintikul, A., *et al.*, "Evaluation of diffuse myocardial fibrosis in heart failure with cardiac magnetic resonance contrast-enhanced T_1 mapping," *Journal of the American College of Cardiology*, vol. 52, no. 19, pp. 1574–1580, 2008.

[8] Wolf, M., Boer, A. de, Sharma, K., *et al.*, "Magnetic resonance imaging T_1-and T_1-mapping to assess renal structure and function: A systematic review and statement paper," *Nephrology Dialysis Transplantation*, vol. 33, no. suppl_2, pp. ii41–ii50, 2018.

[9] Girometti, R., Cereser, L., Bonato, F., and Zuiani, C., "Evolution of prostate MRI: From multiparametric standard to less-is-better and different-is better strategies," *European Radiology Experimental*, vol. 3, pp. 1–14, 2019.

[10] Sherrer, R. L., Glaser, Z. A., Gordetsky, J. B., Nix, J. W., Porter, K. K., and Rais-Bahrami, S., "Comparison of biparametric MRI to full multiparametric MRI for detection of clinically significant prostate cancer," *Prostate Cancer and Prostatic Diseases*, vol. 22, no. 2, pp. 331–336, 2019.

[11] Ma, D., Gulani, V., Seiberlich, N., *et al.*, "Magnetic resonance fingerprinting," *Nature*, vol. 495, no. 7440, pp. 187–192, 2013.

[12] Bipin Mehta, B., Coppo, S., Frances McGivney, D., *et al.*, "Magnetic resonance fingerprinting: A technical review," *Magnetic Resonance in Medicine*, vol. 81, no. 1, pp. 25–46, 2019.

[13] Poorman, M. E., Martin, M. N., Ma, D., *et al.*, "Magnetic resonance fingerprinting Part 1: Potential uses, current challenges, and recommendations," *Journal of Magnetic Resonance Imaging*, vol. 51, no. 3, pp. 675–692, 2020.

[14] Panda, A., Mehta, B. B., Coppo, S., *et al.*, "Magnetic resonance fingerprinting–an overview," *Current Opinion in Biomedical Engineering*, vol. 3, pp. 56–66, 2017.

[15] Weigel, M., "Extended phase graphs: Dephasing, RF pulses, and echoes-pure and simple," *Journal of Magnetic Resonance Imaging*, vol. 41, no. 2, pp. 266–295, 2015.

[16] Barbieri, M., Brizi, L., Giampieri, E., *et al.*, "A deep learning approach for magnetic resonance fingerprinting: Scaling capabilities and good training practices investigated by simulations.," *Physica Medica*, vol. 89, pp. 80–92, 2021.

[17] McGivney, D. F., Pierre, E., Ma, D., *et al.*, "SVD compression for magnetic resonance fingerprinting in the time domain," *IEEE Transactions on Medical Imaging*, vol. 33, no. 12, pp. 2311–2322, 2014.

[18] Hoppe, E., Körzdörfer, G., Würfl, T., *et al.*, "Deep learning for magnetic resonance fingerprinting: A new approach for predicting quantitative parameter values from time series," *GMDS*, vol. 243, pp. 202–206, 2017.

[19] Fang, Z., Chen, Y., Liu, M., *et al.*, "Deep learning for fast and spatially constrained tissue quantification from highly accelerated data in magnetic resonance fingerprinting," *IEEE Transactions on Medical Imaging*, vol. 38, no. 10, pp. 2364–2374, 2019.

[20] Balsiger, F., Shridhar Konar, A., Chikop, S., *et al.*, "Magnetic resonance fingerprinting reconstruction via spatiotemporal convolutional neural networks," in *Machine Learning for Medical Image Reconstruction: First International Workshop, MLMIR 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Proceedings 1*, Springer, 2018, pp. 39–46.

[21] Cohen, O., Zhu, B., and Rosen, M. S., "MR fingerprinting deep reconstruction network (DRONE)," *Magnetic Resonance in Medicine*, vol. 80, no. 3, pp. 885–894, 2018.

[22] Liao, C., Wang, K., Cao, X., *et al.*, "Detection of lesions in mesial temporal lobe epilepsy by using MR fingerprinting," *Radiology*, vol. 288, no. 3, pp. 804–812, 2018.

[23] Ma, D., Jones, S. E., Deshmane, A., *et al.*, "Development of high-resolution 3D MR fingerprinting for detection and characterization of epileptic lesions," *Journal of Magnetic Resonance Imaging*, vol. 49, no. 5, pp. 1333–1346, 2019.

[24] Keil, V. C., Bakoeva, S. P., Jurcoane, A., *et al.*, "MR fingerprinting as a diagnostic tool in patients with frontotemporal lobe degeneration: A pilot study," *NMR in Biomedicine*, vol. 32, no. 11, e4157, 2019.

[25] Keil, V. C., Bakoeva, S. P., Jurcoane, A., *et al.*, "A pilot study of magnetic resonance fingerprinting in Parkinson's disease," *NMR in Biomedicine*, vol. 33, no. 11, e4389, 2020.

[26] Chen, Y., Jiang, Y., Pahwa, S., *et al.*, "MR fingerprinting for rapid quantitative abdominal imaging," *Radiology*, vol. 279, no. 1, pp. 278–286, 2016.

[27] Hamilton, J. I., Jiang, Y., Eck, B., Griswold, M., and Seiberlich, N., "Cardiac cine magnetic resonance fingerprinting for combined ejection fraction, T_1 and T_2 quantification," *NMR in Biomedicine*, vol. 33, no. 8, e4323, 2020.

[28] Gao, Y., Chen, Y., Ma, D., *et al.*, "Preclinical MR fingerprinting (MRF) at 7T: Effective quantitative imaging for rodent disease models," *NMR in Biomedicine*, vol. 28, no. 3, pp. 384–394, 2015.

[29] Zhao, B., Haldar, J. P., Liao, C., *et al.*, "Optimal experiment design for magnetic resonance fingerprinting: Cramér-rao bound meets spin dynamics," *IEEE Transactions on Medical Imaging*, vol. 38, no. 3, pp. 844–861, 2018.

[30] Hoppe, E., Thamm, F., Körzdörfer, G., *et al.*, "Magnetic resonance fingerprinting reconstruction using recurrent neural networks," in *GMDS*, 2019, pp. 126–133.

[31] Abadi, M., Barham, P., Chen, J., *et al.*, "TensorFlow: A system for Large-Scale Machine Learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283.

[32] Bergstra, J., Yamins, D., Cox, D. D., *et al.*, "Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms," in *Proceedings of the 12th Python in Science Conference*, Citeseer, vol. 13, 2013, p. 20.

[33] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P., "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[34] Satopaa, V., Albrecht, J., Irwin, D., and Raghavan, B., "Finding a" kneedle" in a haystack: Detecting knee points in system behavior," in *2011 31st International Conference on Distributed Computing Systems Workshops*, IEEE, 2011, pp. 166–171.

[35] Glorot, X. and Bengio, Y., "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[36] Kingma, D. P. and Ba, J., "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[37] Hong, J.-S., Hermann, I., Zöllner, F. G., *et al.*, "Acceleration of magnetic resonance fingerprinting reconstruction using denoising and self-attention pyramidal convolutional neural network," *Sensors*, vol. 22, no. 3, p. 1260, 2022.

[38] Rieger, B., Zimmer, F., Zapp, J., Weingärtner, S., and Schad, L. R., "Magnetic resonance fingerprinting using echo-planar imaging: Joint quantification of T_1 and T_2^* relaxation times," *Magnetic Resonance in Medicine*, vol. 78, no. 5, pp. 1724–1733, 2017.

[39] Rieger, B., Akçakaya, M., Pariente, J. C., *et al.*, "Time efficient whole-brain coverage with mr fingerprinting using slice-interleaved echo-planar-imaging," *Scientific Reports*, vol. 8, no. 1, p. 6667, 2018.

[40] Hermann, I., Chacon-Caldera, J., Brumer, I., *et al.*, "Magnetic resonance fingerprinting for simultaneous renal T_1 and T_2^* mapping in a single breath-hold," *Magnetic Resonance in Medicine*, vol. 83, no. 6, pp. 1940–1948, 2020.

[41] Yu, A. C., Badve, C., Ponsky, L. E., *et al.*, "Development of a combined MR fingerprinting and diffusion examination for prostate cancer," *Radiology*, vol. 283, no. 3, pp. 729–738, 2017.

[42] Su, P., Mao, D., Liu, P., *et al.*, "Multiparametric estimation of brain hemodynamics with MR fingerprinting ASL," *Magnetic Resonance in Medicine*, vol. 78, no. 5, pp. 1812–1823, 2017.

[43] Wang, C. Y., Liu, Y., Huang, S., Griswold, M. A., Seiberlich, N., and Yu, X., "31P magnetic resonance fingerprinting for rapid quantification of creatine kinase reaction rate in vivo," *NMR in Biomedicine*, vol. 30, no. 12, e3786, 2017.

[44] Badrinarayanan, V., Kendall, A., and Cipolla, R., "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[45] Ronneberger, O., Fischer, P., and Brox, T., "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, Springer, 2015, pp. 234–241.

Conclusions

This PhD thesis proposes novel computational methods for addressing biomedical image segmentation and generation problems. The developed algorithms are designed to achieve one of the following objectives: accelerate operations typically performed manually or with slow computational methods, increase the accuracy of the results, create methods applicable across various imaging techniques and anatomical areas, and develop explainable models to ensure easy understanding of the computed results. In particular, the thesis explores three main research topics: i) the development of a new method based on statistical mechanics for segmenting biomedical images, ii) the creation of a deep learning (DL) system for the automatic segmentation and quantification of COVID-19 lung lesions from computed tomography (CT) images, and iii) the development of a DL method for the rapid generation of quantitative maps acquired using the magnetic resonance fingerprinting (MRF) methodology.

Considering the first research project, we developed a kinetic version of a bounded confidence consensus model for biomedical segmentation problems. The key idea of this approach is to represent each pixel as a particle with an evolving position and a static gray level. The particles interact with each other by grouping together and forming regions of segmentation. To increase the model's flexibility we added a non-constant diffusion term, which allows us to account for stochastic variations in the image acquisition process. We derived the Boltzmann formulation of the model, which was efficiently simulated using a direct Monte Carlo approach. Additionally, we proposed an optimization strategy to fine-tune parameters for improved segmentation results

We tested this system using three different biomedical datasets: the HL60 cell nuclei dataset, the brain tumor dataset, and the thigh muscles dataset, which included both healthy and facioscapulohumeral dystrophy (FSHD) patients. We evaluate the performance of the segmentation algorithm using the Dice similarity coefficient (DSC_{metric}) which quantifies the overlap between the predicted and the true masks. Encouraging results were obtained for HL60 cell nuclei and brain tumor datasets, with DSC_{metric} equal to or greater than 0.91. However, the segmentation precision was comparatively lower for the thigh muscles dataset, where DSC_{metric} scores were 0.73 for FSHD subject and 0.60 for healthy subject. To address this challenge, we introduced a patch-based approach, dividing the image into smaller arrays and applying the segmentation system to subregions, which improved the quality of the segmentation masks in the thigh muscles dataset, resulting in a DSC_{metric} value of 0.67 for the healthy subject. Additionally, we observed that the optimal diffusion coefficient is higher in FSHD patients compared to healthy subjects, suggesting its potential as a

marker for FSHD muscle impairment.

The main advantages of this method are summarized as follows:

- *Explainability*: since the model is based on a statistical mechanics clustering model, the segmentation process is transparent and all computations are easy to comprehend.

- *Unsupervised*: the system is an unsupervised model that doesn't require a training process to learn the segmentation task. This is particularly useful when dealing with small datasets where a training process may not be feasible.

- *Scalability to different dataset and imaging techniques*: because it doesn't rely on a training process, the model can be applied to various anatomical areas and imaging modalities without the need for modifications to the segmentation algorithm.

- *Flexibility to data acquisition imperfections*: by including the diffusion term in the model, it takes into account potential imperfections in the data acquisition processes and provides a way to quantify them.

Despite these positive aspects, our work can be extended and improved in several directions. Firstly, it is necessary to validate the model on larger and different datasets to assess the reliability and generalizability of the model. Additionally, we observed that segmentation performance is lower in complex segmentation problems where distinguishing boundaries between regions of interest (ROIs) is challenging. Therefore, future research should aim to enhance segmentation performance in more complex segmentation tasks. One possible approach is to incorporate external manual information to help or guide the segmentation process. This may involve specifying the initial points for segmentations or the total number of ROIs to be created.

In the second study, we proposed the LungQuant system, a fully-automatic DL pipeline aimed at segmenting and quantifying COVID-19 lung lesions. Pneumonia is one of the most prevalent complications of COVID-19, even in its early stages. However, assessing the extent of lung lesions in CT scans of COVID-19 patients can be a challenging task, mainly due to the intricate nature of these lesions. The automated assignment of a severity score to CT scans of subjects afflicted with COVID-19 pneumonia has the potential to alleviate the workload in radiology departments.

The LungQuant system consists of a cascade of two U-nets, which are specialized convolutional neural network (CNN) architectures designed for image segmentation tasks. The first U-net identifies the lung parenchyma, whereas the second one operates within a bounding box enclosing the segmented lungs to identify the regions affected by COVID-19. A morphological refinement step is inserted between the two U-nets to eliminate small regions within the segmented mask that are not connected to the lungs. The system's output includes the

computed percentage of affected lung, which is further converted into the CT-Severity Score (CT-SS). The pipeline was trained and tested on public datasets to ensure the verifiability and reproducibility of our results. In particular, a completely independent dataset was used to evaluate the final performance of the algorithm. The segmentation tasks, assessed in terms of the DSC_{metric}, achieved values of 0.95 ± 0.01 for lung segmentation and 0.66 ± 0.13 for lesions segmentation. In terms of CT-SS classification, the system demonstrated a good level of accuracy, achieving a value of 90% on this independent test dataset.

The key features of the LungQuant system are summarized as follows:

- *Fully automatic*: the LungQuant system operates completely automatically, speeding up and simplifying operations that are typically done manually and visually.
- *Comprehensive output*: the system provides outputs for all main phases, including lung and COVID-19 lesion segmentation masks, the computed percentage of affected lung, and the corresponding CT-SS. These outputs not only offer a comprehensive evaluation of disease severity but also make it easier to identify possible errors within the system and determine at which stage they occur.
- *Verifiability and reproducibility*: the complete system was trained and tested on public datasets to ensure the verifiability and reproducibility of our results. Furthermore, the complete code of the LungQuant pipeline is publicly available in the following public repository: https://doi.org/10.15161/oar.it/76937.

Regarding future perspectives on this research topic, our work is currently undergoing three distinct extensions. Firstly, we have proposed an improvement to the pipeline [1] by incorporating an initial CNN to crop CT scans to the lung region, excluding anatomical regions not essential for this application. In reference [1], it was demonstrated that this modification also improves the system's performance in severe cases, where performance was previously lower. This modified system was subsequently validated in a multicentric study involving 14 radiologists from 5 different centers [2]. This study proved the clinical utility of an automated system to quantify COVID-19 lung lesions. Finally, the system's output is being utilized in a radiomics study to predict the clinical outcomes of COVID-19 patients. This involves extracting radiomic features from the segmented masks of COVID-19 lesions and passing them as input of a Machine Learning classification pipeline to obtain clinical outcome predictions.

In the third research project, we presented an optimized magnetic resonance (MRF) framework to provide quantitative multiparametric maps for preclinical studies. MRF is an imaging technique for performing simultaneous measurements of multiple tissue properties through a single and time-efficient acquisition schedule. While MRF data acquisition is faster compared to standard mapping

techniques, the traditional post-processing procedure has been relatively slow and demanded significant storage capacity. To address these aspects, we proposed a DL method and an optimization strategy for reconstructing T_1 and T_2 maps from ex vivo rat brain phantoms using two MRF acquisition profiles. This approach allowed the simultaneous optimization of the neural network architecture, the structure of the DL model and the supervised learning algorithm. The DL system improved the reconstruction performance in the estimation of both T_1 and T_2 maps and the computational time necessary for the calculation. Furthermore, we demonstrated that the DL system allows us to use a lower number of MRF images and a lower k-space sampling percentage, significantly reducing the acquisition time of MRF examinations.

Through an optimized acquisition and reconstruction framework, we improved the following aspects of the preclinical MRF methodology:

- *Speed-up the reconstruction process*: the DL method and hyperparameter optimization allowed for faster and more efficient reconstruction of T_1 and T_2 maps, reducing the computational time required for calculation by at least a factor of 37.

- *Speed-up the acquisition process*: the DL method allowed to maintain comparable reconstruction performance even with a lower number of MRF images and a reduced k-space sampling percentage with respect to the traditional reconstruction method, significantly reducing the acquisition time of MRI examinations.

- *Enhanced performance and reduction of map artifacts*: the DL method decreased the mean percentage relative error by a factor of 3 for T_1 and by a factor of 2 for T_2. Additionally, we observed that the DL method effectively minimized artifacts in T_1 and T_2 maps, potentially attributed to imperfections in the RF excitation field.

As for future perspectives on this research topic, there are several directions for further improvement and development. Firstly, to enhance the robustness and generalizability of our method, it is crucial to increase the sample size and extend our methodology to include various anatomical regions and pathological tissues. Furthermore, our findings suggest that the proposed MRF sequences, when combined with DL reconstruction, have the potential to be adapted for in vivo preclinical imaging. However, this transition necessitates careful adjustments to the acquisition protocol to prioritize the health of the animals. Another potential direction for improvement is the utilization of compressed and faster acquisition sequences, such as echo-planar imaging-based sequences, to further accelerate MRF acquisitions. Finally, to enhance the information provided by our method, we can incorporate additional biological and physical parameters among the evaluated maps. This expansion may offer a more comprehensive quantification of tissue properties, thus broadening the potential applications of MRF in preclinical studies.

References

[1] Lizzi, F., Postuma, I., Brero, F., *et al.*, "Quantification of pulmonary involvement in COVID-19 pneumonia: An upgrade of the LungQuant software for lung CT segmentation," *The European Physical Journal Plus*, vol. 138, no. 4, pp. 1–10, 2023.

[2] Scapicchio, C., Chincarini, A., Ballante, E., *et al.*, "A multicenter evaluation of a deep learning software (LungQuant) for lung parenchyma characterization in COVID-19 pneumonia," *European Radiology Experimental*, vol. 7, no. 1, p. 18, 2023.

List of Papers

Published

1. **R. F. Cabini**, L. Barzaghi, D. Cicolari, P. Arosio, S. Carrazza, S. Figini, M. Filibian, A. Gazzano, R. Krause, M. Mariani, M. Peviani, A. Pichiecchio, D. Pizzagalli, A. Lascialfari. *Fast Deep Learning reconstruction techniques for preclinical Magnetic Resonance Fingerprinting.* NMR in Biomedicine 2023, e5028. DOI: https://doi.org/10.1002/nbm.5028.

2. F. Lizzi, I. Postuma, F. Brero, **R. F. Cabini**, M. E. Fantacci, A. Lascialfari, P. Oliva, L. Rinaldi, A. Retico. *Quantification of pulmonary involvement in COVID-19 pneumonia: an upgrade of the LungQuant software for lung CT segmentation.* The European Physical Journal Plus 2023, 138.4: 1-10. DOI: https://doi.org/10.1140/epjp/s13360-023-03896-4.

3. C. Scapicchio, A. Chincarini, E. Ballante, L. Berta, E. Bicci, C. Bortolotto, F. Brero, **R. F. Cabini**, G. Cristofalo, S. C. Fanni, M. E. Fantacci, S. Figini, M. Galia, P. Gemma, E. Grassedonio, A. Lascialfari, C. Lenardi, A. Lionetti, F. Lizzi, M. Marrale, M. Midiri, C. Nardi, P. Oliva, N. Perillo, I. Postuma, L. Preda, V. Rastrelli, F. Rizzetto, N. Spina, C. Talamonti, A. Torresin, A. Vanzulli, F. Volpi, E. Neri, A. Retico. *A multicenter evaluation of a deep learning software (LungQuant) for lung parenchyma characterization in COVID-19 pneumonia.* European Radiology Experimental 2023, 7.1: 18-18. DOI: https://doi.org/10.1186/s41747-023-00334-z.

4. **R. F. Cabini**, A. Pichiecchio, A. Lascialfari, S. Figini, M. Zanella. *A kinetic approach to consensus-based segmentation of biomedical images.* arXiv preprint 2022, arXiv:2211.05226. DOI: https://doi.org/10.48550/arXiv.2211.05226.

5. **R. F. Cabini**, F. Brero, A. Lancia, O. Oneta, C. Bortolotto, E. Ballante, E. Puppo, L. Bianchini, C. Stelitano, E. Alì, V. Bartolomeo, M. Montesano, E. Merizzoli, F. Agustoni, G. M. Stella, R. Sun, L. Preda, S. Figini, A. Lascialfari, E. Deutsch, A. R. Filippi. *Preliminary report on harmonization of features extraction process using the ComBat tool in the multi-center "Blue Sky Radiomics" study on stage III unresectable NSCLC.* Insights into imaging 2022, 13.1: 1-9. DOI: https://doi.org/10.1186/s13244-022-01171-1.

6. F. Lizzi, A. Agosti, F. Brero, **R. F. Cabini**, M. E. Fantacci, S. Figini, A. Lascialfari, F. Laruina, P. Oliva, S. Piffer, I. Postuma, L. Rinaldi, C. Talamonti, A. Retico. *Quantification of pulmonary involvement in COVID-19 pneumonia by means of a cascade of two U-nets: training*

and assessment on multiple datasets using different annotation criteria. International Journal of Computer Assisted Radiology and Surgery 2021, 26: 1-9. DOI: https://doi.org/10.1007/s11548-021-02501-2.

7. C. Varesio, S. Gana, A. Asaro, E. Ballante, **R. F. Cabini**, E. Tartara, L. Pasca, S. Orcesi, R. Borgatti, E. M. Valente, V. De Giorgis. *Diagnostic yield and cost-effectiveness of "dynamic" exome analysis in epilepsy with neurodevelopmental disorders: a tertiary-center experience in northern Italy.* Diagnostics 2021, 11.6: 948. DOI: https://doi.org/10.3390/diagnostics11060948.

Submitted

1. **R. F. Cabini**, A. Pichiecchio, A. Lascialfari, S. Figini, M. Zanella. *A kinetic approach to consensus-based segmentation of biomedical images.* Under review in Mathematics and Computers in Simulation.

2. C. Scapicchio, E. Ballante, A. Benfante, L. Berta, C. Bortolotto, F. Brero, **R. F. Cabini**, A. Chincarini, D. Cicolari, P. E. Colombo, S. C. Fanni, M. E. Fantacci, S. Figini, E. Grassedonio, A. La Fiura, A. Lascialfari, C. Lenardi, A. Lionetti, F. Lizzi, M. Marrale, C. Nardi, E. Neri, I. Postuma, L. Preda, F. Rizzetto, N. Scichilone, N. Spina, C. Talamonti, A. Torresin, L. Ubaldi, A. Vanzulli, G. Zorzi, A. Retico. *Prediction of the COVID-19 severity outcome based on radiomic features extracted from lung lesions detected in CT scans by the LungQuant software.* Under review in Physica Medica.

3. F. Lizzi, E. Ballante, F. Brero, **R. F. Cabini**, M. E. Fantacci, S. Figini, A. Lascialfari, G. Paternó, I. Postuma, P. Oliva, C. Scapicchio, A. Retico. *Fully automated deep learning based system for COVID-19 patient outcome prediction.* Under review in Intelligence-Based Medicine.

4. C. Bortolotto, A. Pinto, F. Brero, G. Messana, **R. F. Cabini**, I. Postuma, A. Robustelli Test, G. M. Stella, G. Galli, M. Mariani, S. Figini, A. Lascialfari, A. R. Filippi, O. M. Bottinelli, L. Preda. *Multi-messenger Radiomics of Lung cancer: preliminary results for radiomic features with different imaging modalities and extraction software.* Under review in European Journal of Radiology.

5. M. C. Ramusino, C. Imbimbo, M. Capelli, **R. F. Cabini**, S. Bernini, F. P. Lombardo, L. Mazzocchi, L. M. Farina, A. Pichiecchio, G. Perini, A. Costa. *Morphological changes in the fronto-limbic circuit are associated with the occurrence of neuropsychiatric symptoms in neurocognitive disorders.* Under review in Frontiers in Psychiatry.

6. F. Garofoli, V. Franco, S. Aversa, M. Angelini, **R. F. Cabini**, S. Cociglio, S. Longo, M. Palmisani, C. Pisoni, S. Ghirardello, S. Orcesi on behalf of "Oral Melatonin in Preterm Newborn Study Group". *Fate of melatonin*

orally administered in preterm newborns: anti-oxidant performance and basis for neuroprotection. Under review in Journal of Pineal Research.

In Progress

1. D. U. Pizzagalli, **R. F. Cabini**, A. Cozzi, B. Thelen, A. Lascialfari, S. Figini, R. Krause, S. M. R. Rizzo. *CompositIA, automated quantification of body composition scores from toraco-abdominal CT scans.*

Conference abstracts

1. A. Robustelli Test, F. Brero, A. Pinto, C. Bortolotto, **R. F. Cabini**, M. Mariani, I. Postuma, L. Preda, A. Lascialfari *Multi-messenger radiomics of lung cancer: preliminary results for radiomic features stability among different imaging modalities and extraction software.* In Proceedings of 109° Congresso Società Italiana di Fisica 2023.

2. **R. F. Cabini**, L. Barzaghi, D. Cicolari, A. Pichiecchio, S. Figini, M. Filibian, A. Gazzano, M. Peviani, A. Pichiecchio, A. Lascialfari. *Magnetic Resonance Fingerprinting for spinal cord T_1 and T_2 mapping.* In Proceedings of the European Magnetic Resonance Congress (EUROMAR) 2023.

3. L. Barzaghi, **R. F. Cabini**, D. Cicolari, A. Pichiecchio, S. Figini, M. Filibian, A. Gazzano, M. Peviani, A. Pichiecchio, A. Lascialfari. *Fast Deep Learning techniques for Magnetic Resonance Fingerprinting.* In Proceedings of the European Magnetic Resonance Congress (EUROMAR) 2023.

4. A. Robustelli Test, F. Brero, A. Pinto, C. Bortolotto, **R. F. Cabini**, M. Mariani, I. Postuma, L. Preda, A. Lascialfari *MRI and CT of lung cancer patients: a comparison between radiomic features from two different imaging modalities.* In Proceedings of the European Magnetic Resonance Congress (EUROMAR) 2023.

5. I. Postuma, F. Lizzi, A. Agosti, F. Brero, **R. F. Cabini**, M. E. Fantacci, S. Figini, A. Lascialfari, F. Laruina, P. Oliva, S. Piffer, C. Scapicchio, C. Talamonti, A. Retico. *The LungQuant system: a software tool for the quantification of pulmonary involvement in Covid-19.* In Proceedings of Congresso Nazionale Associazione Italiana Fisica Medica (AIFM) 2023.

6. C. Scapicchio, A. Chincarini, E. Ballante, L. Berta, E. Bicci, C. Bortolotto, F. Brero, **R. F. Cabini**, G. Cristofalo, S. C. Fanni, M. E. Fantacci, S. Figini, M. Galia, P. Gemma, E. Grassedonio, A. Lascialfari, C. Lenardi, A. Lionetti, F. Lizzi, M. Marrale, M. Midiri, C. Nardi, P. Oliva, N. Perillo, I. Postuma, L. Preda, V. Rastrelli, F. Rizzetto, N. Spina, C. Talamonti,

A. Torresin, A. Vanzulli, F. Volpi, E. Neri, A. Retico. *Covid-19 severity prediction based on radiomic features extracted from lung CT scans using the LungQuant segmentation software.* In Proceedings of Congresso Nazionale Associazione Italiana Fisica Medica (AIFM) 2023.

7. F. Brero, **R. F. Cabini**, I. Villa, A. Lancia, O. Oneta, C. Bortolotto, E. Ballante, E. Puppo, L. Bianchini, C. Stelitano, E. Alì, V. Bartolomeo, M. Montesano, E. Merizzoli, F. Agustoni, G. M. Stella, R. Sun, L. Preda, S. Figini, E. Deutsch, A. R. Filippi, A. Lascialfari. *Blue Sky study: radiomics features as predictive markers of progression-free-survival in stage III, PD-L1 positive NSCLC.* In Proceedings of Congresso Nazionale Associazione Italiana Fisica Medica (AIFM) 2023.

8. A. Robustelli Test, F. Brero, A. Pinto, C. Bortolotto, **R. F. Cabini**, M. Mariani, I. Postuma, L. Preda, A. Lascialfari *MRI and CT images of lung cancer patients: radiomic analysis and features extraction software comparison.* In Proceedings of Congresso Nazionale Associazione Italiana Fisica Medica (AIFM) 2023.

9. **R. F. Cabini**, S. Figini, A. Pichiecchio, A. Lascialfari, M. Zanella. *Application of Mean Field Interaction Algorithm to medical image Segmentation.* In Proceedings of Congresso Nazionale Associazione Italiana Fisica Medica (AIFM) 2023.

10. A. R. Filippi, J. Saddi, E. Ballante, F. Brero, **R. F. Cabini**, M. Mariani, I. Villa, C. Bortolotto, F. Agustoni, G. M. Stella, S. La Mattina, G. Facheris, P. Borghetti, P. Pedrazzoli, L. Preda, A. Lascialfari. *Radiomic features and PFS post-PACIFIC in the Blue Sky Crossectional Study on stage 3 PD-L1-SSCLC.* In Proceedings of ESTRO 2023.

11. **R. F. Cabini**, L. Barzaghi, D. Cicolari, A. Pichiecchio, S. Figini, P. Arosio, M. Peviani, M. Filibian, S. Carrazza, A. Lascialfari. *Fast Deep Learning models for Magnetic Resonance Fingerprinting.* In Proceedings of: Joint Annual Meeting ISMRM-ESMRMB & ISMRT 32nd Annual Meeting, 03-08 June 2023, Toronto, ON, Canada.

12. C. Scapicchio, E. Ballante, F. Brero, **R. F. Cabini**, A. Chincarini, M. E. Fantacci, S. Figini, A: Lascialfari, F. Lizzi, I. Postuma, A. Retico. *Integration of a Deep Learning-Based Module for the Quantification of Imaging Features into the Filling-in Process of the Radiological Structured Report.* In Proceedings of Cognitive Computing for Healthcare 2023.

13. **R. F. Cabini**, L. Barzaghi, D. Cicolari, A. Pichiecchio, S. Figini, P. Arosio, M. Filibian, S. Carrazza, A. Lascialfari. *Fast Deep Learning reconstruction techniques for magnetic resonance Fingerprinting.* In Proceedings of XIII Congresso Associazione Italiana Risonanza Magnetica in Medicina 2022.

14. F. Brero, A. Pinto, C. Bortolotto, **R. F. Cabini**, M. Mariani, I. Postuma, A. Robustelli Test, L. Preda, A. Lascialfari. *MRI and CT images of lung*

cancer patients: preliminary results for radiomic features stability among imaging modalities and comparison of extraction softwares. In Proceedings of XIII Congresso Associazione Italiana Risonanza Magnetica in Medicina 2022.

15. C. Ibimbo, M. Cotta Ramusino, G. Perini, **R. F. Cabini**, L. Melazzini, S. Bernini, F. Lombardo, L. Farina, A. Costa. *Association of neuropsychiatric symptoms and morphologic alterations of limbic system in patients with different types of dementia.* In Proceedings of XVII Convegno Nazionale Sindem Firenze, Villa Vittoria, 13-15 ottobre 2022.

16. **R. F. Cabini**, L. Barzaghi, D. Cicolari, A. Pichiecchio, S. Figini, P. Arosio, M. Filibian, S. Carrazza, A. Lascialfari. *Fast Deep Learning reconstruction techniques for magnetic resonance Fingerprinting.* In Proceedings of Italian-French International Conference on Magnetic Resonance, Milano, 27-30 September 2022.

17. **R. F. Cabini**, F. Brero, I. Villa, A. Lancia, O. Oneta, C. Bortolotto, E. Ballante, E. Puppo, L. Bianchini, C. Stelitano, E. Alì, V. Bartolomeo, M. Montesano, E. Merizzoli, F. Agustoni, G. M. Stella, R. Sun, L. Preda, S. Figini, E. Deutsch, A. R. Filippi, A. Lascialfari. *Harmonization of features extraction process using the ComBat tool in a multicenter study on stage III unresectable Non-Small-Cell Lung Cancer.* In Proceedings of 108° Congresso Società Italiana di Fisica 2022.

18. I. Postuma, F. Brero, **R. F. Cabini**, S. Panebianco, L. Preda, A. Filippi, C. Bortolotto, A. Lascialfari. *Cascata di convolutional neural network per il contornamento dei polmoni e di noduli tumorali del polmone.* In Proceedings of 108° Congresso Società Italiana di Fisica 2022.

19. M. Marrale, A. La Fiura, G. Collura, M. C. D'Oca , F. Lizzi, F. Brero, **R. F. Cabini**, I. Postuma, L. Rinaldi, C. Scapicchio, I. Castiglioni, G. Cristofalo, E. Grassedonio, G. M. Galia, N. Scichilone, A. Retico. *Machine learning classification for COVID19 patients performed on small datasets of CT scans.* In Proceedings of 108° Congresso Società Italiana di Fisica 2022.

20. **R. F. Cabini**, L. Barzaghi, D. Cicolari, A. Pichiecchio, S. Figini, P. Arosio, M. Peviani, M. Filibian, S. Carrazza, A. Lascialfari. *Application of Deep Learning techniques to Magnetic Resonance Fingerprinting.* In Proceedings of: Joint Annual Meeting ISMRM-ESMRMB & ISMRT 31st Annual Meeting, 07-12 May 2022, London, UK.

21. F. Lizzi , F. Brero, **R. F. Cabini**, M. E. Fantacci, S. Piffer, I. Postuma, L. Rinaldi, A. Retico. *Making Data Big for a Deep-learning Analysis: Aggregation of Public COVID-19 Datasets of Lung Computed Tomography Scans.* In Proceedings of: 10th International Conference on Data Science, Technology and Applications - DATA, pages 316-321 2021.

22. F. Lizzi, A. Agosti, F. Brero, **R. F. Cabini**, M. E. Fantacci, S. Figini, A. Lascialfari, F. Laruina, P. Oliva, S. Piffer, I. Postuma, L. Rinaldi, C. Talamonti, A. Retico. *The role of AI in COVID-19 management: quantification of lung involvement using CT scans.* In Proceedings of: CoViD-19: Inflammation and Molecular Imaging 2021.

23. A. Lancia, F. Brero, O. Oneta, C. Bortolotto, E. Ballante, **R. F. Cabini**, E. Puppo, M. Mariani, C. Stelitano, E. Alì, V. Bartolomeo, M. Montesano, E. Merizzoli, F. Agustoni, G. M. Stella, R. Sun, L. Preda, L. Bianchini, S. Figini, A. Lascialfari, E. Deutsch, A. R. Filippi. *Radiomics features extraction harmonization process in the Blue Sky study on stage III NSCLC patients undergoing chemoradiation and immunotherapy.* In Proceedings of: Biology-Guided Adaptive Radiotherapy - Acta Oncologica Conference Virtual Symposia 2021.