Universitat de Barcelona

Facultat de Matemàtiques i Informàtica

Double Bachelor's degree in Mathematics and Computer Science

Bachelor's Thesis

MACHINE LEARNING AND FRACTAL-BASED ANALYSIS FOR THE AUTOMATED DIAGNOSIS OF CARDIOVASCULAR DISEASES USING MAGNETIC RESONANCE

Autor: Sergi Romero Gris

Director:	Dra. Polyxeni Gkontra
	Dr. Joan Carles Tatjer Montaña
Realitzat a:	Departament de Matemàtiques i Informàtica
Barcelona,	13 de juny de 2023

Contents

Abstract

Cardiac magnetic resonance (CMR) is the reference imaging modality for the diagnose of cardiovascular diseases. Traditionally, simple CMR parameters related to the volume and shape of the cardiac structures are calculated by the medical professionals by means of manual or semi-automated approaches. This process is time-consuming and prone to human errors. Moreover, despite the importance of these traditional CMR indexes, they often fail to fully capture the complexity of the cardiac tissue.

In this work, we propose a novel approach for automated cardiovascular disease diagnosis, using ischemic heart disease as an example use case. Towards this aim, we will use a state-of-the-art technology, supervised machine learning, and a promising mathematical tool, fractal-based analysis.

In order to undertand the potential information that can be derived from fractal-based features, we introduce and explore the concepts of Haussdorff dimension, box-counting dimension and lacunarity. We describe the interrelationships among these concepts and present computational algorithms for calculating box-counting dimension and lacunarity.

The study is based on data from a large-cohort study, UK Biobank, to extract box-counting dimension and lacunarity from CMR textures focusing on three cardiac structures of medical interest: the left ventricle, the right ventricle and the myocardium. The extraction of these features allows us to obtain quantitative parameters regarding the complexity and heterogeneity of the tissue. These fractal features, both individually and in conjunction with other vascular risk factors and CMR traditional indexes, are employed as inputs to state-of-the-art machine learning models, including SVM, XGBoost, and random forests. The objective is to determine if the inclusion of fractal features enhances the performance of currently employed parameters.

The performance evaluation of our models is based on metrics such as balanced accuracy, F1 score, precision, and recall. The results obtained demonstrate the potential of fractal-based features in improving the accuracy and reliability of cardiovascular diseases diagnosis.

Resum

Les ressonàncies magnètiques cardiovasculars (RMC) son la modalitat de referència per a la diagnosi de malalties cardiovasculars. Tradicionalment, alguns

2020 Mathematics Subject Classification. 92C55, 68T01 , 28A78

paràmetres simples de les RMC relacionats amb el volum i la forma de les estructures cardíaques són calculats pels professionals mèdics per mitjans manuals o semi automatitzats. Aquest és un procés lent i exposat a l'error humà. A més a més, tot i la importància d'aquests índexs, sovint no poden capturar completament la complexitat del teixit cardíac.

En aquest treball presentem una nova proposta per a la diagnosi automàtica de malaties caríaques, amb les malalties de les artèries coronaries com exemple. Amb aquest objectiu, utilitzarem models d'aprenentatge autònom, i una prometedora eina matemàtica, l'anàlisi fractal.

Per tal de d'entendre la informació potencial que podem extreure de l'anàlisi fractal, indroduirem i explorarem els conceptes de dimensió de Haussdoff, dimensió de comptatge de cel·les i lacunaritat. Descriurem les relacions entre aquests conceptes i presentarem algorismes per al càlcul de la dimensió de comptatge de cel·les i la lacunaritat.

Aquest estudi utilitza dades d'un estudi grupal, el UK Biobank, per a l'extracció de la dimensió de comptatge de cel·les i la lacunaritat de la textura d'imatges de RMC, posant l'ull en tres estructures cardíaques d'interès mèdic: el ventricle esquerre, el ventricle dret i el miocardi. L'extracció d'aquests paràmetres ens permet obtenir una mesura quantitativa de la complexitat i la heterogeneïtat del teixit. Aquests paràmetres fractals, sols i juntament amb altres factors de risc vascular i índexs tradicionals de la RMC, són emprats com a dades d'entrada per a moderns models d'aprenentatge autònom com SVM, XGBoost i boscos aleatoris. L'objectiu es determinar si incloure les característiques fractal millora el funcionament dels paràmetres utilitzats fins ara.

El rendiment dels nostres models serà evaluat en termes d'exactitud balancejada, puntuació F1, precissió i sensibilitat. Els resultats obtinguts demostren el potencial de l'anàlisi fractal en la millora de l'exactitud i la fiabilitat del diagnòsic de malaties cardiovasculars.

Resumen

Las resonancias magnéticas cardiovasculares (RMC) son la modalidad de referencia para diagnosis de enfermedades cardiovasculares. Tradicionalmente, algunos parámetros simples de las RMC relacionados con el volumen y la forma de las estructuras caríacas son calculados por los professionales médicos por medios manuales o semi automatizados. Este proceso es lento y esta expuesto a errores humanos. Además, a pesar de la importancia de estos índices, a menudo no pueden capturar completamente la complejidad del tejido cardíaco.

En este trabajo presentamos una nueva propuesta para diagnosticar automáti-

camente enfermedades cardíacas, com la enfermedad de las arterias coronarias como ejemplo. Con este fin usaremos modelos de aprendizaje autónomo, y una prometedora herramienta matemática, el análisis fractal.

Para entender la información potencial que podemos extraer del análisis fractal, introduciremos y exploraremos los conceptos de dimensión de Haussdorff, dimensión de contaje de celdas y lacunaridad. Describiremos las relaciones entre estos conceptos y presentaremos algoritmos para calcular la dimensión de contaje de celdas y la lacunaridad.

Este estudio utiliza datos de un estudio grupal, el UK Biobank, para extraer la dimensión de contaje de celdas y la lacunaridad de la textura de imágenes de RMC, centrándonos en tres estructuras cardíacas de interés médico: el ventrículo izquierdo, el ventrículo derecho y el miocardio. La extracción de estos parámetros nos permite obtener medidas cuantitativas de la complejidad y la heterogeneidad del tejido. Estos parámetros fractales, solos y junto a otros factores de riesgo vascular e índices tradicionales de la RMC, son usados como datos de entrada para modernos modelos de aprendizaje autónomo como SVM, XGBoost y bosques aleatorios. El objetivo es determinar si incluir las características fractales mejora el desempeño de los parámetros usados hasta ahora.

El rendimiento de nuestros modelos será evaluado en terminos de exactitud balanceada, puntuación F1, presisión y sensibilidad. Los resultados obtenidos demuestran el potencial del análisis fractal en la mejora de la exactitud y la fiabilidad del diagnóstico de enfermedades cardiovasculares.

Chapter 1

Introduction

1.1 Motivation

Cardiovascular diseases (CVDs) are one of the main causes of death worldwide. In European Union CVDs accounts for nearly 40% of all deaths [42]. Among the various forms of CVDs, ischemic heart disease (IHD) stands out as the main global cause of death. Although mortality of IHD in Western countries has drastically decreased through the last decades, developing countries pose new challenges for public health. Globalisation and adoption of Western lifestyles can lead to higher prevalence of cardiovascular risk factors, leading to a raise of mortality caused by CVDs [30].

Invasive diagnostic methods are associated with complications. To overcome this issue, several noninvasive imagine techniques are available. Among them, cardiac magnetic resonance images (CMR) are widely used as they provide valuable information regarding heart function and structure, and, thus, diagnosing cardiac diseases. [40]. To this end, automated CMR quantitative analysis pipelines that can deal with the vast amount of data produced by current CMR scanners are of paramount importance for timely and accurate disease diagnosis.

In the other hand, in the field of pattern recognition, fractal-based analysis, it is been emerging as a promising technique for classifying textures and geometric shapes used in different science disciplines [44][48]. Fractal-based metrics such as box-counting dimension and lacunarity provide texture information that is shown to be useful in many fields including in medical imaging and cardiology [10].

1.2 Objectives

The main aim of this thesis is to evaluate the potential of fractal-based analysis for improving diagnosis of cardiac diseases using machine learning and CMR images. The following main objectives had to be fulfilled to achieve the aim of this thesis:

1. Theoretical study of fractal-based parameters for the analysis of texture in 3D images.
2. Quantification of fractal-based properties for three cardiac structures (left ventricle, right ventricle and myocardium) from 3D CMR obtained at two time-points of the cardiac cycle: end-systole (ES) and end-diastole (ED).
3. Use of the extracted fractal-based features to develop and validate an automated system for cardiac diseases diagnosis from CMR images based on machine learning, focusing on ischemic heart disease as an example use case.

1.3 State of the art

The concept of fractal was first introduced by B. B. Mandelbrot [27] along with the concept of lacunarity and with the help of the existing senses of Hausdorff dimension and its computable approximation box-counting dimension. In 1992, N. Skara and B. B. Chaudhuri proposed an algorithm to compute this box-dimension for textured images, the widely known as differential box counting algorithm (DBC)[38]. More refined versions of the algorithm have been proposed [23] [46] with most recent being the work of Liu et al. in 2014 [24]. This method deals with two known problems of the previous versions: over-counting boxes along grey scale direction and under-counting block on neighbourhoods with sharp grey-level differences. For lacunarity a computable-oriented definition was introduced by C. Allain and M. Cloitre [2] for binary images and was extended by P. Dong [15] for textured images. More recently more efficient methods have been published for both binary and texture scenarios [45][4].

Fractal-based parameters have been the subject of investigation in medical imaging research, specifically in relation to heart diseases, over the past decade. These parameters have shown promise as analytical tools in this field [20][9]. For instance, G. Captur et al. used box-counting dimension for binary images to analyse the endocardial border based on CMR images, resulting in accurate results for left ventricle non-compaction of the myocardium diagnosis [10]. Furthermore, Gkontra et al. [18] observed significant differences in 3D reconstructions of the coronary microvasculature between healthy and infarcted tissues in pig models. These differences were identified in terms of for box-dimension, lacunarity and succolarity. Captur et al. [11] characterised the complexity of left ventricular trabeculation among different populations concluding that Chinese American patients have slightly less complexity in endocardial structures. Meanwhile, African

American patients, Hispanic patients, hypertensive patients and those with hypertrophy tend to have more complex endocardial geometry. S. Lorthois and F. Cassot introduced a two-scale analysis method for fractal characterisation of vascular networks[25]. They state that at small scale, healthy vascular structures are organised as a homogeneous space filling mesh and, at large scale, they form a quasi-fractal tree structure.

1.4 Document organization

This thesis is structured in the following manner:

- **Fractal dimension and lacunarity:** In this section the concepts of Haussdorff dimension, box-counting dimension and lacunarity are introduced. Results such as geometric invariance and relation among Haussdorff and box-counting dimensions are proved. Algorithm for box-counting dimension and lacunarity computation are explained.
- **Classification models:** In this section we will review the theoretical background of used classification models: SVM, random forests and XGboost.
- **Image feature extraction:** In this section we will explain the process of feature extraction, from CMR raw data to definitive feature values.
- **Methodology:** In this section the experimental design and proposed machine learning pipeline is described. It covers the selection and description of the study sample, the feature selection and the feature combination.
- **Results:** In this section the experimental results of the study are summarised and analysed. Moreover, detailed explainability analysis of the best performing model is included using of a state-of-the-art explainability method.
- **Conclusions and further work:** Conclusions of the project and ideas for further work regarding medical imaging fractal-based analysis.

Chapter 2

Fractal dimensions and lacunarity

In the following chapter, key metrics of fractal sets are presented. We will define Haussdorff dimension, box-counting dimension and lacunarity.

Haussdorff dimension is used in some theoretical results in fractal geometry and other fields. Box-counting dimension and lacunarity are used in applied problems such as medical classification problems [10], geological analysis [48] and remote sensing [44]. Box-dimension has a close relation with Haussdorff dimension and it is usually used as a computable attractive approximation. They both provide a metric of how complex is the set.

On the other hand, Lacunarity has less relation with dimensions but provides complementary information on how homogeneously are holes distributed through the fractal set and allows us to differentiate sets with same dimensions.

These concepts are defined and an algorithm is explained for box-counting dimension and lacunarity allowing us to approximate them in our practical experiments.

2.1 Hausdorff measure and dimension

The habitual concept of dimension is what is called **topological dimension** and is always a natural number. There are several definitions of topological dimension, but for this case we will define it as follows:

Definition 2.1. *Let F be a subset of $\mathbb{R}^n$. We say that:*

- *F has **dimension zero**, if each of its points has arbitrarily small neighbourhoods whose boundaries do not intersect the set.*
- *F has **dimension** k, if each point of F has arbitrarily small neighbourhoods whose boundaries intersect F in a set of dimension $k-1$, and k is the smallest natural number for which this happens.*

We will note topological dimension of a set F as $dim_T(F)$

This topological dimension is not good enough to distinguish between fractals with different degree of density. We can see that Sierpinski triangle (Figure A.1) and Koch snowflake (Figure A.2) both have topological dimension 1. In order to differentiate fractal sets we need a new notion to measure the "complexity" of fractals. We will introduce the notion of **Haussdorff Dimension**.

2.1.1 Haussdorff dimension

Let F be a non-empty subset of n-dimensional Euclidean space $\mathbb{R}^n$. Given a set U we will denote by $|U| = \sup\{|x-y| : x, y \in U\}$ the diameter of U.

Definition 2.2. *A δ-cover of the set F is a countable (or finite) collection of sets $\{U_i\}$ with diameters $0 < |U_i| \leq \delta$ that covers F. i.e. $F \subset \bigcup_{i=1}^{\infty} U_i$.*

Definition 2.3. *Let $\{U_i\}$ be a δ-cover of F and $s \geq 0$ we define*

$$\mathcal{H}^s_\delta(F) = \inf\left\{ \sum_{i=0}^{\infty} |U_i|^s : \{U_i\} \textit{ is a } \delta\textit{-cover of } F \right\} \tag{2.1}$$

Thus, we look at all covers of F by sets of diameter at most δ and seek to minimise the sum of the sth powers of the diameter. As δ decreases, the possible covers of F is reduced. Therefore, the infimum of $\mathcal{H}^s_\delta(F)$ increases, as $\delta \to 0$ and so, approaches a limit.

Definition 2.4. *We define the s-dimensional Hausdorff measure of F, $\mathcal{H}^s(F)$ as*

$$\mathcal{H}^s(F) = \lim_{\delta \to 0} \mathcal{H}^s_\delta(F)$$

Note that this limit exists for any subset $F \subset \mathbb{R}^n$, although it can be 0 or ∞.

$\mathcal{H}^s$ may be shown to be a measure for sets $F \in \mathbb{R}^n$, this proof can be found in *Real Analysis: Measure Theory, Integration, and Hilbert Spaces, section VII* [41].

It is clear that for any given set $F \subset \mathbb{R}^n$ and $\delta < 1$, $\mathcal{H}^s_\delta(F)$ is none-increasing with s, so $\mathcal{H}^s(F)$ is also non-increasing. In fact, rather more is true: if $t > s$ and $\{U_i\}$ is a δ-cover of F, then

$$\sum_i |U_i|^t \leq \sum_i |U_i|^{t-s} |U_i|^s \leq \delta^{t-s} \sum_i |U_i|^s$$

so, taking infima over all δ-covers,

$$\mathcal{H}^t_\delta(F) \leq \delta^{t-s} \mathcal{H}^s_\delta(F)$$

Letting $\delta \to 0$, we see that if $\mathcal{H}^s(F) \leq \infty$, then $\mathcal{H}^t(F) = 0$ for $t > s$. Thus, a graph of $\mathcal{H}^s(F)$ against s shows that there is a critical value of s at which $\mathcal{H}^s(F)$ "jumps" from ∞ to 0. This critical value is called the *Hausdorff dimension* of F for any set $F \subset \mathbb{R}^n$.

Definition 2.5. *Let $F \subset \mathbb{R}^n$, $F \neq \emptyset$. Let $\mathcal{H}^s$ the s-dimensional Haussdorff measure, we define the Hausdorff-Besicovitch dimension of F, $dim_H F$ as*

$$dim_H F = \inf\{s \geq 0 : \mathcal{H}^s(F) = 0\} = \sup\{s : \mathcal{H}^s(F) = \infty\}$$

so that,

$$\mathcal{H}^s(F) = \begin{cases} \infty & \text{if } 0 \leq s < dim_H F \\ 0 & \text{if } s > dim_H F \end{cases}$$

Some basic properties of *Haussdorff dimension* follow from the definition and the definition of *Haussdorff measure* itself.

- *Monotonicity.* If $E \subset F$, then $dim_H E \leq dim_H F$. This is immediate from the measure property that $\mathcal{H}^s(E) \leq \mathcal{H}^s(F)$ for each s.
- *Range of values.* If $F \subset \mathbb{R}^n$, then $0 \leq dim_H F \leq n$. Clearly, Haussdorff dimensions are non-negative. If F is bounded and $s > n$, then $\mathcal{H}^s(F) = 0$. Thinking about F as a countable union of bounded sets we have $\mathcal{H}^s(F) = 0$ if $s > n$ $\forall F \subset \mathbb{R}^n$ as we have countable additivity for Haussdorff measure. In conclusion, we have $dim_H F \leq n$.

Some other properties can be found in *Fractal Geometry: Mathematical Foundations and Applications* [16] chapter 3.

2.1.2 Geometric invariant

In topology, we say that two sets are equivalent if there exists a homeomorphism between them. As topological dimension defines a topological invariant, two equivalent sets will always have the same topological dimension. We can approach this sense of equivalence in fractal geometry, defining two sets as equivalent if there exists a bi-Lipschitz mapping between them. We will see in Proposition 2.8 that if two sets are bi-Lispchitz equivalent they must have the same *Haussdorff dimension*. Note that from this proposition we can not state all set with same dimension are bi-Lispchitz equivalent. With this proposition we can see that *Haussdorff dimension* is invariant for geometric transformations.

The concepts introduced in this section are stated in *Fractal Geometry: Mathematical Foundations and Applications* [16] chapter 3.

Definition 2.6. *A function $f : X \to Y$ is called a Hölder function of exponent α (or f satisfies the α-Hölder condition) if*

$$|f(x) - f(y)| \leq c|x-y|^{\alpha} \ x, y \in X$$

for some constant $c \geq 0$. f is Lipschitz if is a Hölder function of exponent 1.

Proposition 2.7. *Let $F \subset \mathbb{R}^n$ and $f : F \to \mathbb{R}^m$ be a mapping such that*

$$|f(x) - f(y)| \leq c|x-y|^{\alpha} \ x, y \in F$$

for constants $\alpha > 0$ and $c > 0$. Then for each s

$$\mathcal{H}^{s/\alpha}(f(F)) \leq c^{s/\alpha}\mathcal{H}^{s}(F).$$

In particular, if f is a Lipschitz mapping, that is if $\alpha = 1$, then

$$\mathcal{H}^{s}(f(F)) \leq c^{s}\mathcal{H}^{s}(F)$$

Proof. If $\{U_i\}$ is a δ-cover of F, then since $|f(F \cap U_i)| \leq c|F \cap U_i|^{\alpha} \leq c|U_i|^{\alpha}$, it follows that $\{f(F \cap U_i)\}$ is a $c\delta^{\alpha}$-cover of $f(F)$. Thus, $\sum_i |f(F \cap U_i)|^{s/\alpha} \leq c^{s/\alpha} \sum_i |U_i|^{s}$, so that $\mathcal{H}^{s/\alpha}_{c\delta^{\alpha}}(f(F)) \leq c^{s/\alpha}\mathcal{H}^{s}_{\delta}(F)$. Letting $\delta \to 0$ we obtain $\mathcal{H}^{s/\alpha}(f(F)) \leq c^{s/\alpha}\mathcal{H}^{s}(F)$.

The result for Lipschitz mappings follows immediate when setting $\alpha = 1$. □

Proposition 2.8. *(a) Let $F \subset \mathbb{R}^n$ and suppose that $f : F \to \mathbb{R}^n$ satisfies the α-Hölder condition then $dim_H f(F) \leq (1/\alpha) dim_H F$. In particular, if f is a Lipschitz mapping, that is, $\alpha = 1$, then $dim_H f(F) \leq dim_H F$.*

(b) If $f : F \to \mathbb{R}^m$ is a bi-Lispchitz transformation, that is

$$c_1|x-y| \leq |f(x) - f(y)| \leq c|x-y| \ x, y \in F \tag{2.2}$$

where $0 < c_1 \leq c < \infty$, then $dim_H f(F) = dim_H F$.

Proof. (a) If $s > dim_H F$, then by Proposition 2.7 $\mathcal{H}^{s/\alpha}(f(F)) \leq c^{s/\alpha}\mathcal{H}^{s}(F) = 0$, so $dim_H f(F) \leq s/\alpha$ for all $s > dim_H F$. The conclusion for Lipschitz mappings is immediate on taking $\alpha = 1$.

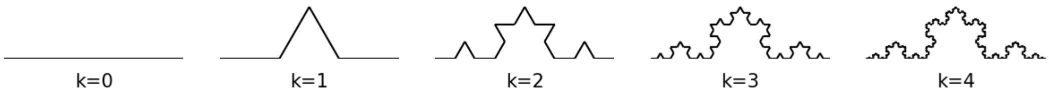


Figure 2.1: Von Koch's curve construction. k represents the construction's iteration step.

(b) If f is *bi-Lipschitz*, then $f : F \to f(F)$ is a bijection with inverse $f^{-1} : f(F) \to F$. For $u, v \in f(F)$ let $x = f^{-1}(u), y = f^{-1}(v)$ in the left-hand inequality of (2.2). Then

$$c_1|f^{-1}(u) - f^{-1}(v)| \leq |f(f^{-1}(u)) - f(f^{-1}(v))| = |u - v|$$

so f^{-1} is Lipschitz. Applying part (a) to f^{-1} gives $dim_H F = dim_H f^{-1}(f(F)) \leq dim_H f(F)$. From (a) we have $dim_H f(F) \leq dim_H F$ thus necessarily $dim_H f(F) = dim_H F$.

□

From Proposition 2.8 we have geometric invariance as dimension is not affected by congruence, similarities or affine transformations of the original set. We also have from (a) a lower bound for $dim_H F$.

2.1.3 Computing Hausdorff dimension

There is not a general method that allows us to calculate the *Haussdorff dimension*. We can see some dimension computation examples for middle third Cantor set C (Figure 2.2) (later in this section), uniform Cantor set [16], $C \times [0,1]$ [16] or generalised middle Cantor set [29]. Nevertheless, when we compute the dimension of self similar sets such as middle third Cantor set or von Knoch's curve it becomes much easier. In this section, we will prove a theorem (theorem 2.12) which provides us with a simple way to compute *Haussdorff dimension* for self similar sets, and we will compute the dimension of a middle-third Cantor set using upper and lower estimate.

Self-similar sets

In basic terms, a subset of $\mathbb{R}^n$ is self-similar if it can be split into parts which are geometrically similar to the whole set.

Let us ceonter our attention to von Koch's curve (see Figure 2.1). In its construction, every segment of length d is replaced by four segments of length $d/3$. The von Koch's curve K is the limit of the polygonal curves obtained in this way. It is a non-rectifiable curve (i.e. has infinite length). It can also be presented in terms of similarity maps S_i in the form

$$K = S_1(K) \cup S_2(K) \cup S_3(K) \cup S_4(K)$$

where S_i are the orientation-preserving similarities of ration $1/3$ of the plane which map the first initial segment onto the four next ones.

Let us introduce the concepts in order to define self-similar sets.

Definition 2.9. *A mapping $S : \mathbb{R}^n \to \mathbb{R}^n$ is called a similitude if there is $r \in (0,1)$ such that*

$$|S(x) - S(y)| = r|x - y|, \textit{ for } x, y \in \mathbb{R}^n$$

Similitudes are exactly those maps S that can be written as

$$S(x) = r \cdot g(x) + z, \ x \in \mathbb{R}^n$$

for some $g \in O(n)$[1], $z \in \mathbb{R}^n$ and $0 < r < 1$. Suppose $S = S_1, ..., S_n$, $n \geq 2$, is a finite sequence of similitudes with contraction ratios $r_1, ..., r_n$. We say that a non-empty compact set K is invariant under S if

$$K = \bigcup_{i=1}^{n} S_i(K)$$

Then for any such S there exists a unique invariant compact set. This invariant set is the unique fixed point of the map $\tilde{S} : E \to \cup_{i=1}^{n} S_i(E)$. Note that $\tilde{S}$ is a contraction in the Haussdorff metric[2] as it has a unique fixed point. By definition this point is the invariant set we wanted [29].

In addition, from the properties of contractions in complete metric spaces, it follows that independently from the initial compact set $F \subset \mathbb{R}^n$ chosen, the iterations

$$\tilde{S}^m(F) = \tilde{S} \circ ... \circ \tilde{S}(F) = \bigcup_{i_1=1}^{n} ... \bigcup_{i_m=1}^{n} S_{i_1} \circ ... \circ S_{i_m}(F)$$

will converge to K. Moreover, for any m the set K satisfies

$$K = \bigcup_{i_1=1}^{n} ... \bigcup_{i_m=1}^{n} S_{i_1} \circ ... \circ S_{i_m}(K).$$

[1] $O(n)$ represents the orthogonal group i.e. all linear maps $g : \mathbb{R}^n \to \mathbb{R}^n$ preserving euclidean distance.

[2] Haussdorff metric is defined, for X, Y compact sets in a metric space with metric d, as $d_H(X,Y) = \max\{d(x,Y), d(y,X) : x \in X, y \in Y\}$.

Since

$$|S_{i_1} \circ ... \circ S_{i_m}(K)| \leq (\max_{1 \leq i \leq n} r_i)^m |K| \to 0, \textit{ as } m \to \infty$$

an invariant set can be expressed as a union of arbitrary small sets geometrically similar to itself.

Definition 2.10. *We define an invariant set K under S to be self-similar if, with $s = dim_H K$,*

$$\mathcal{H}^s(S_i(K) \cap S_j(K)) = 0 \textit{ for } i \neq j$$

This definition might seem awkward but its construction is useful because for some known sets we can take a stronger separation condition called *open set condition* in order to determine the dimension of K explicitly from the contraction ratios of the similitude S_i in S.

Definition 2.11. *We say that S satisfies the open set conditions if there exists a non-empty open set O such that*

$$\bigcup_{i=1}^{n} S_i(O) \subset O$$

and

$$S_i(O) \cap S_j(O) = \emptyset \textit{ for } i \neq j$$

Theorem 2.12. *If S satisfies the open set condition, then the invariant set K is self-similar and $0 < \mathcal{H}^s(K) < \infty$. Then $s = dim_H K$, where s is the unique number for which*

$$\sum_{i=1}^{n} r_i^s = 1$$

For a proof of this theorem see Hutchinson (section 5.3) [21].

For the von Koch's curve, taking O as the open triangle which contains the first four line segments, we can check that it satisfies the open set condition. As ratios r_i are all equal to $1/3$ we have $4 \cdot 3^{-s} = 1$ thus $s = \frac{\log 4}{\log 3}$.

In general, if the ratios are equal $r = r_1 = ... = r_n$ we have $dim_H K = \frac{\log n}{\log 1/r}$.

Computing the middle-third Cantor set dimension

Now we will compute the *Haussdorff dimension* of the middle-third Cantor set (Figure 2.2) on a line C using upper and lower bounds. From theorem 2.12 we can expect *Haussdorff dimension* to be $\frac{\log 2}{\log 3}$ as the parts of C are disjoint, thus its construction satisfies the open set condition.

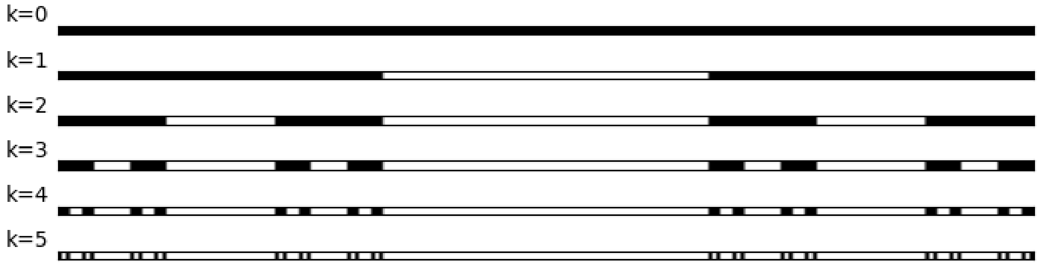


Figure 2.2: First five iterations of the middle-third cantor set. Black pixels represent points in the set and k values represent iteration step.

Its construction starts with a closet interval, let us take $[0,1]$, we will call it C_0. At the first step, the middle third $(\frac{1}{3},\frac{2}{3})$ is removed and we get the set $C_1 = [0,\frac{1}{3}] \cup [\frac{2}{3},1]$. Next step is to repeat the same procedure of removing the middle third of each interval of C_1 resulting in 4 intervals of length $\frac{1}{9}$. This process repeats again and again, so C_n consists of 2^n intervals of length 3^{-n}. The Cantor set C is defined as the limit of C_n in the Haussdorff metric, which in this case is the same as $\bigcap_{n=1}^{\infty} C_n$.

With this definition we can now compute the *Haussdorff dimension*. We will see that, for $s = \log 2/\log 3$ we get finite $\mathcal{H}^s(C)$. Therefore $dim_H(C) = \log 2/\log 3$.

Taking the intervals of C_n as a 3^{-n}-covers of C gives that $\mathcal{H}^s_{3^{-k}}(C) \leq 2^k 3^{-ks} = 1$ if $s = \frac{\log 2}{\log 3}$. Letting $k \to \infty$ gives $\mathcal{H}^s(C) \leq 1$.
Now we will show that

$$\sum |U_i|^s \geq \frac{1}{2} = 3^{-s}$$

for any cover $\{U_i\}$ of C in order to prove that $\mathcal{H}^s(C) \geq \frac{1}{2}$. Clearly, it is enough to assume that $\{U_i\}$ are intervals, and by expanding them slightly and using the compactness of C, we need only to verify the last inequality if $\{U_i\}$ is a finite collection of closed sub-intervals of $[0,1]$. For each U_i, let k be the integer such that

$$3^{-(k+1)} \leq |U_i| < 3^{-k} \tag{2.3}$$

The U_i can intersect at most one level-k interval since the separation of these level-k intervals is at least 3^{-k}. If $j \geq k$, then, by construction, U_i intersects at most $2^{j-k} = 2^j 3^{-sk} \leq 2^j 3^s |U_i|^s$ level-j intervals of C_j, using (2.3). If we choose j large enough so that $3^{-(j+1)} \leq |U_i|$ for all U_i, then since the $\{U_i\}$ intersect all 2^j basic intervals of length 3^{-j}, counting intervals gives $2^j \leq \sum_i 2^j 3^s |U_i|^s$, which reduces to $\sum |U_i|^s \geq \frac{1}{2} = 3^{-s}$. Thus, $\mathcal{H}^s(C) \geq 1/2$ so it is finite and not zero as we wanted to see.

2.2 Box-counting dimension

In this section we will mainly concern about **box-counting** dimension, which has a simple formulation and is widely used as a computational approach for fractal dimensions due to its intuitively definition that provides a direct algorithm to calculate it. As done for *Haussdorff dimension* we will see that this dimension is invariant for geometric transformations (proposition 2.20). Also we will see how *box-counting dimension* provides an upper bound for *Haussdorff dimension* (Proposition 2.17). This upper bound results into equality for self similar sets (Corollary 2.19).

2.2.1 Box-counting dimension

Let F be a non-empty bounded subset of $\mathbb{R}^n$ and let $N_\delta(F)$ be the least numbers of sets of diameter at most δ which can cover F, that is the least number of set of any δ-cover of F.

Definition 2.13. *The lower and upper box-counting dimensions of F, are defined respectively as*

$$\underline{dim}_B F = \underline{\lim}_{\delta\to 0} \frac{\log N_\delta(F)}{-\log\delta}$$

$$\overline{dim}_B F = \overline{\lim}_{\delta\to 0} \frac{\log N_\delta(F)}{-\log\delta}$$

If they are equal we refer to the common value as the box-counting dimension of F

$$dim_B F = \lim_{\delta\to 0} \frac{\log N_\delta(F)}{-\log\delta}$$

We can see that with this definition we get that $N_\delta \simeq c\delta^{-s}$ for small δ, where $s = dim_B F$, more precisely,

$$\begin{array}{ll} N_\delta(F)\delta^s \to \infty & \text{if } s < dim_B F \\ N_\delta(F)\delta^s \to 0 & \text{if } s > dim_B F \end{array}$$

Note that to solve s in $N_\delta \simeq c\delta^{-s}$, we take logarithms

$$\log N_\delta(F) \simeq \log c - s\log\delta$$

so

$$s \simeq \frac{\log N_\delta(F)}{-\log\delta} + \frac{\log c}{\log\delta}$$

and we might hope to obtain s as

$$s = \lim_{\delta \to 0} \frac{\log N_\delta(F)}{-\log \delta} \tag{2.4}$$

with the second term disappearing in the limit.

There are several equivalent definition for box counting dimension that might be more convenient for our purposes. Instead of using sets of diameter at most δ which can cover F, we can use disjoint balls of radius δ with centres in F or take the family of cubes of the form

$$[m_1\delta, (m_1+1)\delta] \times ... \times [m_n\delta, (m_n+1)\delta]$$

where $m_1, ..., m_n$ are integers. This is called the δ-mesh or δ-grid of $\mathbb{R}^n$

First we will prove the equivalence for the δ-mesh.

Proposition 2.14. *Let $N_\delta(F)$ be the smallest number of sets of diameter δ that can cover F. Let $N'_\delta(F)$ be the number of δ-mesh cubes that intersect F. Then*

$$\lim_{\delta \to 0} \frac{\log N_\delta(F)}{-\log \delta} = \lim_{\delta \to 0} \frac{\log N'_\delta(F)}{-\log \delta}$$

Proof. We have a collection of $N'_\delta(F)$ sets of diameter $\delta\sqrt{n}$ (as they are cubes) that cover F. By construction of $N_\delta(F)$ we have

$$N_{\delta\sqrt{n}}(F) \leq N'_\delta(F)$$

We also have that any set of diameter at most δ is contained in 3^n mesh cubes of side δ. So we have

$$N'_\delta(F) \leq 3^n N_\delta(F)$$

Combining these inequalities and dividing by $-\log \delta$,

$$\frac{\log N_{\delta\sqrt{n}}(F)}{-\log \delta} \leq \frac{\log N'_\delta(F)}{-\log \delta} \leq \frac{\log 3^n + \log N_\delta(F)}{-\log \delta} \tag{2.5}$$

so taking lower limits as $\delta \to 0$,

$$\underline{\lim}_{\delta \to 0} \frac{\log N_\delta(F)}{-\log \delta} \leq \underline{\lim}_{\delta \to 0} \frac{\log N'_\delta(F)}{-\log \delta} \leq \underline{\lim}_{\delta \to 0} \frac{\log N_\delta(F)}{-\log \delta} \tag{2.6}$$

Thus the definition of lower box dimension does not change working with $N_\delta(F)$ or $N'_\delta(F)$.
Analogously, we can see the same result for upper bound. □

Note that with this equivalence, we can see that the *box-counting dimension* is independent of the origin and orientation chosen for the δ-mesh.

Before proving the equivalence for disjoint balls of radius δ with centres in F, we will define the packing number

Definition 2.15. *Let F be a non-empty bounded set of $\mathbb{R}^n$, we will define the packing number of F with diameter δ as the largest number of disjoint balls of radius δ with centres in F. And we will note it as*

$$P(F,\delta) = \max\{k: \textit{ there are disjoint balls } B(x_i,\delta), i = 1,...,k, \textit{ with } x_i \in F\}$$

Proposition 2.16. *Let $N_\delta(F)$ be the smallest number of sets of diameter δ that can cover F. Let $P(F,\delta)$ be the packing number of F with diameter δ. Then*

$$\lim_{\delta\to 0}\frac{\log N_\delta(F)}{-\log\delta} = \lim_{\delta\to 0}\frac{\log P(F,\delta)}{-\log\delta}$$

The equivalence is given by the following proposition [16].

Proof. Let $B_1,...B_{P(F,\delta)}$ a collection of disjoint balls of radius δ and centres in F.

If x belongs to F, then x must be within distance δ of one of the B_i, otherwise the ball of centre x and radius δ can be added to form a larger collection of disjoint balls. Thus, the $P(F,\delta)$ balls concentric with the B_i but of radius 2δ (and diameter 4δ) cover F, giving

$$N_{4\delta}(F) \leq P(F,\delta)$$

Suppose now that $B_1,..B_{P(F,\delta)}$ are disjoint balls of radius δ with centres in F. Let $U_1,...,U_k$ be any collection of sets of diameter at most δ which cover F. Since U_j must cover the centres of the B_i, each B_i must contain at least one of the U_j. As the B_i are disjoint, there are at least as many U_j as B_i. Hence,

$$P(F,\delta) \leq N_\delta(F)$$

Just as in (2.5) - (2.6), on taking logarithms of these inequalities, dividing by $-\log\delta$ and taking the limit, we see that the values of definition 2.13 are unaltered if $N_\delta(F)$ is replace by $P(F,\delta)$. □

Analogously to *Haussdorff dimension* some basic properties follow from the definition.

- *Monotonocity.* If $E \subset F$, then $\underline{dim}_B E \leq \underline{dim}_B F$ and $\overline{dim}_B E \leq \overline{dim}_B F$.

- *Range of values.* If $F \subset \mathbb{R}^n$, $F \neq \emptyset$ and F bounded. $0 \leq \underline{dim}_B F \leq \overline{dim}_B F \leq n$.

Some other properties can be found in *Fractal Geometry: Mathematical Foundations and Applications* [16].

It is reasonable to inquire about the connection between the Hausdorff dimension and the box-counting dimension, and this relationship is summarised in the following proposition.

Proposition 2.17. *For every non-empty bounded $F \subset \mathbb{R}^n$*

$$dim_H F \leq \underline{dim}_B F \leq \overline{dim}_B F \tag{2.7}$$

Proof. Suppose that $1 < \mathcal{H}^s(F) = \lim_{\delta \to \infty} \mathcal{H}^s_\delta(F)$ for some $s \geq 0$. Then, for all sufficiently small δ

$$1 < \mathcal{H}^s(F) \leq N_\delta(F)\delta^s$$

where $N_\delta(F)$ is the least number of sets of diameter δ that can cover F, using (2.1). Taking logarithms, $0 < \log N_\delta(F) + s \log \delta$, and it follows that

$$s \leq \underline{\lim}_{\delta \to 0} \frac{N_\delta(F)}{-\log \delta}$$

□

Let us focus on how "regular" must be a set in order to have equality in 2.7. Mattila [29] proposed the following characterisation.

Theorem 2.18. *Let F be a non-empty subset of $\mathbb{R}^n$. Suppose there are a Borel measure*[3] *$\mu : \mathbb{R}^n \to \mathbb{R}^n$ and positive numbers a, b, r_0 and s such that*

$$0 < ar^s \leq \mu(B(x,r)) \leq br^s < \infty \; \forall x \in F, 0 < r \leq r_0$$

Then $dim_H F = \underline{dim}_B F = \overline{dim}_B F = s$.

Proof. If F is covered by subsets U_i such that $0 < |U_i| \leq r_0$ and $F \cap U_i \neq \emptyset$, we can pick points $x_i \in F \cap U_i$ and F is then also covered by the balls $B(x_i, |U_i|)$. Thus

$$b \sum_i |U_i|^s \geq \sum_i \mu(B(x_i, |U_i|)) \geq \mu(F) > 0$$

[3] We say that $\mu : \mathbb{R}^n \to \mathbb{R}^n$ is a Borel measure on $\mathbb{R}^n$ if all Borel sets in $\mathbb{R}^n$ are μ measurable. Recall that the family of Borel sets in $\mathbb{R}^n$ is the smallest σ-algebra containing the open subsets of $\mathbb{R}^n$.

This gives $\mathcal{H}^s(F) \geq \mu(F)/b$, whence

$$s \leq dim_H F \leq \underline{dim_B} F \leq \overline{dim_B} F$$

On the other hand, let $0 < \delta \leq r_0$, $k = P(F,\delta)$ the packing number of F with diameter δ and choose disjoint balls $B(x_i,\delta), x_i \in F$, $i = 1,...,k$. Then

$$aP(F,\delta)\delta^s \leq \sum_{i=1}^{k} \mu(B(x_i,\delta)) \leq \mu(\mathbb{R}^n),$$

Which means

$$\frac{a \cdot P(F,\delta)}{\sum_{i=1}^{k} \mu(B(x_i,\delta))} \leq \delta^{-s}$$

taking logarithms and upper limit we obtain

$$\overline{\lim_{\delta \to 0}} \frac{\log P(F,\delta)}{-\log \delta} \leq s$$

And for proposition 2.16 it means $\overline{dim_B} F \leq s$. □

Mattila stated that, combining this result with theorem 2.12, we obtain equality in (2.7) for self-similar sets [29].

Corollary 2.19. *Let K be a self-similar set generated by similitudes for with the open set condition holds. Then $\overline{dim_B} K = dim_H K$.*

2.2.2 Geometric invariant

As for *Haussdorff dimension*, *box-counting dimension* defines a bi-Lipschitz invariant metric for subsets of $\mathbb{R}^n$.

Proposition 2.20. *(a) If $F \subset \mathbb{R}^n$ and $f : F \to \mathbb{R}^m$ is a Lipschitz transformation, that is,*

$$|f(x) - f(y)| \leq c|x - y| \ \forall x,y \in F$$

then $\underline{dim}_B f(F) \leq \underline{dim}_B F$ and $\overline{dim}_B f(F) \leq \overline{dim}_B F$

(b) If $F \subset \mathbb{R}^n$ and $f : F \to \mathbb{R}^m$ is a bi-Lipschitz transformation, that is,

$$c_1|x - y| \leq |f(x) - f(y)| \leq c|x - y| \ x,y \in F$$

where $0 < c_1 \leq c < \infty$, then $\underline{dim}_B f(F) = \underline{dim}_B F$ and $\overline{dim}_B f(F) = \overline{dim}_B F$.

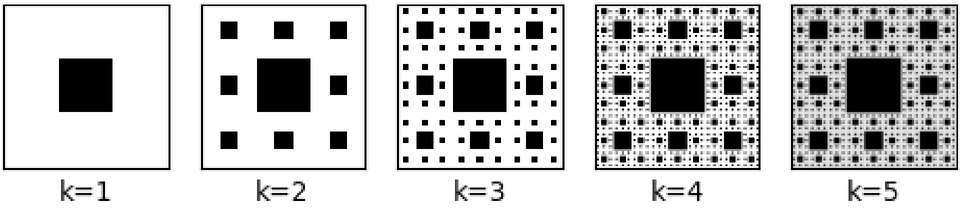


Figure 2.3: First five iterations of Sierpinski carpet construction. White pixels represent point in the set and k values represent iteration step.

Proof for this proposition can be found in [16] Chapter 2, Section 2, and a proof is not provided here as it is similar to Proposition 2.8.

Analogously to *Haussdorff dimension* this proposition tells us that if two sets have different *box-counting dimension*, a bi-Lipschitz mapping does not exist. Providing us with the sense of "equivalence" between fractal sets. Additionally, following from this proposition we have geometric invariance as dimension is not affected by congruence, similarities or affine transformations of the original set. This observation has huge relevance in our case of study because we can compute the *box-counting dimension* of an image without taking into account scales, orientations or rotations.

2.2.3 Computing box-counting dimension

As for Haussdorf dimension there are several ways for computing box-counting dimension. The present dimension offers an advantage over the Hausdorff dimension, because it allows a computationally feasible solution to approximate it. In this subsection, we introduce an algorithm for computing the present dimension and investigate its performance in approximating the theoretically calculated dimensions the Sierpinski carpet. We will also provide an example of the theoretical calculation of a box-dimension for Sierpinski carpet (Figure 2.3).

Computing the Sierpinski carpet set dimension

Sierpinski carpet construction starts from the set F_0 which is a square, consider $[0,1] \times [0,1]$. In stage $k = 1$, the previous square is divided in nine equal squares $F_1^1, F_1^2, F_1^3, F_1^4, F_1^5, F_1^6, F_1^7, F_1^8, F_1^9$ and the central square (we will take F_1^5 as central square) is removed from the set. Next step will repeat this process for every $F_1^i, i = 1, ...9, i \neq 5$. The process is iterated and we define the Sierpinski carpet F as the limit of iterating this construction.

In terms of similarities if $x \in [0,1] \times [0,1]$ we can consider

$$\begin{array}{ll} S_1(x) = x/3 & S_5(x) = x/3 + (2/3,0) \\ S_2(x) = x/3 + (0,1/3) & S_6(x) = x/3 + (2/3,1/3) \\ S_3(x) = x/3 + (0,2/3) & S_7(x) = x/3 + (1/3,2/3) \\ S_4(x) = x/3 + (1/3,0) & S_8(x) = x/3 + (2/3,2/3) \end{array}$$

Then, $F = \lim_{i \to \infty} S_1(F_i) \cup S_2(F_i) \cup S_3(F_i) \cup S_4(F_i) \cup S_5(F_i) \cup S_6(F_i) \cup S_7(F_i) \cup S_8(F_i)$

Let F be the Sierpinski carpet with side length 1. Then $\underline{dim}_B F = \overline{dim}_B F = \log 8 / \log 3$

We can observe that, by construction of F, the kth stage of construction consists of 8^k squares of side length 3^{-k} and diameter $\sqrt{2} \cdot 3^{-k}$. If $\sqrt{2} \cdot 3^{-k} < \delta \leq \sqrt{2} \cdot 3^{-k+1}$, the 8^k squares of kth stage give a δ-cover of F, so $N_\delta(F) \leq 8^k$. Then

$$\overline{dim}_B F = \overline{\lim_{\delta \to 0}} \frac{\log N_\delta(F)}{-\log \delta} \leq \overline{\lim_{\delta \to 0}} \frac{\log 8^k}{-\log \sqrt{2} \cdot 3^{-k+1}} = \frac{\log 8}{\log 3}$$

For the lower dimension, consider any plane set of diameter δ with $\sqrt{2} \cdot 3^{-k-1} \leq \delta < \sqrt{2} \cdot 3^{-k}$. This plane set can intersect at most eight squares of the kth stage. There are 8^k squares so at least $8^k/8$ squares or more are needed to cover F. Hence, $N_\delta(F) \geq 8^{k-1}$, so

$$\underline{dim}_B F = \underline{\lim_{\delta \to 0}} \frac{\log N_\delta(F)}{-\log \delta} \geq \underline{\lim_{k \to \infty}} \frac{\log 8^{k-1}}{-\log 3^{-k-1}} = \frac{\log 8}{\log 3}$$

Thus

$$dim_B F = \frac{\log 8}{\log 3}$$

Algorithm description

The general idea behind box-counting algorithm is to cover the set F with a δ-grid and count $N_\delta(F)$ the least number of elements of the grid that can cover F. Then $dim_B F$ is obtained by using a regression plot between N_δ and δ. This method comes directly from the equation (2.4) and works well for binary n-dimensional images, which are sets in $\mathbb{R}^n$.

Nevertheless our problem involves a more specific domain. We want to compute the box-counting dimension of a 3-dimensional grey level image, which is a 4-dimensional space where three dimensions correspond to space and one dimension represents the grey-scale values of the image. A method for this kind of problems was introduced first by Sarkar and Chaudhuri [38]. They proposed a method for box-counting dimension in 2-dimensional grey scale images called the Differential Box Counting (DBC) technique. It consist on applying formula (2.4) using δ-mesh. For the grey level dimension δ-mesh uses a different δ and it is considered that all block within maximum and minimum grey level values are needed to cover the set. DBC has some limitations as it over counts and under counts the number of boxes needed to cover F simultaneously. Other methods where proposed but no one could deal with the two problems at same time until Liu et al. [24] presented an improved version of DBC. Our version is based in this method but input dimension is increased in order to compute the fractal dimension for 3-dimensional grey level images.

Consider an image of size $P \times Q \times R$ as a 4-dimensional spatial surface with (x, y, z) denoting voxel position on the image space, and the fourth coordinate (t) denoting the voxel grey level. The (x, y, z) space is partitioned into non-overlapping blocks of size $\delta \times \delta \times \delta$, where $\min(P, Q, R)/2 \geq \delta \geq 2$. Since this is the usual practice, P, Q, R are required to be multiple of 2 [24]. We also require δ to divide P, Q, R thus $\delta = 2^r$ for some $r > 1$, $r \in \mathbb{N}$. On each grid we define boxes of size $\delta \times \delta \times \delta \times \delta'$ where $G \cdot \delta = \delta' \cdot \min(P, Q, R)$ and G is the total number of grey levels i.e. $G = 255$.

Our goal is to cover the set surface. Consider the (i, j, k)th block and let I_{min}, I_{max} be the minimum and maximum intensity of grey level within this block. Then we need $n_\delta(i, j, k)$ boxes to cover the set with (i, j, k)th block. With

$$n_\delta(i,j,k) = \begin{cases} ceil(\frac{I_{max} - I_{min} + 1}{\delta'}), & I_{max} \neq I_{min} \\ 1, & I_{max} = I_{min} \end{cases} \tag{2.8}$$

The subscript δ denotes the result using scale δ. With (2.8) we can compute the minimum needed boxes used to cover the (i, j, k)th block. It provides a more finer cover than counting blocks from an static grid over (t) dimension.

Now we define a list of shifts $\epsilon_0, ...\epsilon_p$. We will compute $n_\delta(i, j, k)$ after shifting the block ϵ_q voxels (we will call this value $n_\delta^{\epsilon_q}(i, j, k)$) and we will consider $n_\delta(i, j, k) = \max_{1=0}^{p} n_\delta^{\epsilon_1}(i, j, k)$.

Considering contributions from all blocks, N_r is counted for different values of δ as

$$N_\delta = \sum n_\delta(i,j,k) \tag{2.9}$$

Then $dim_B F$ can be estimated from the slope of the least squares linear fit of $\log(N_\delta)$ versus $\log(1/\delta)$. The algorithm is exposed in pseudocode in algorithm 1. Some notes are presented bellow:

- volume is the n-dimensional grey level image.

- q is a parameter and represents the number of offsets that we will use during *box-counting dimension* computation.

Algorithm 1: Differential Box Counting algorithm

```
Input: volume
Data: $q$
$[P, Q, R] \leftarrow size(volume)$;
$divisors \leftarrow divisors(\min\{P, Q, R\})$;
foreach $\delta \in divisors$ do
    $\delta' \leftarrow 255 \cdot \delta / \min\{P, Q, R\}$;
    $offsets \leftarrow equidistants(\delta, q)$;
    $gird \leftarrow grid(r, volume)$;
    $N_\delta \leftarrow 0$;
    foreach $block \in grid$ do
        $n_\delta^{max} = 0$;
        foreach $\epsilon \in offsets$ do
            $b \leftarrow shift(block, \epsilon)$;
            $n_\delta^{max} = \max\{n_\delta^{max}, n_\delta(b)\}$;
        end
        $N_\delta \leftarrow N_\delta + n_\delta^{max}$;
    end
end
$fit(\log N_\delta, \log(\delta^{-1}))$;
```

- $divisors(n)$ is a function that returns the divisors of a given number n.

- $equidistants(r, n)$ is a function that returns a list of unique rounded integer points that divide the interval $[0, n]$ in r equal intervals.

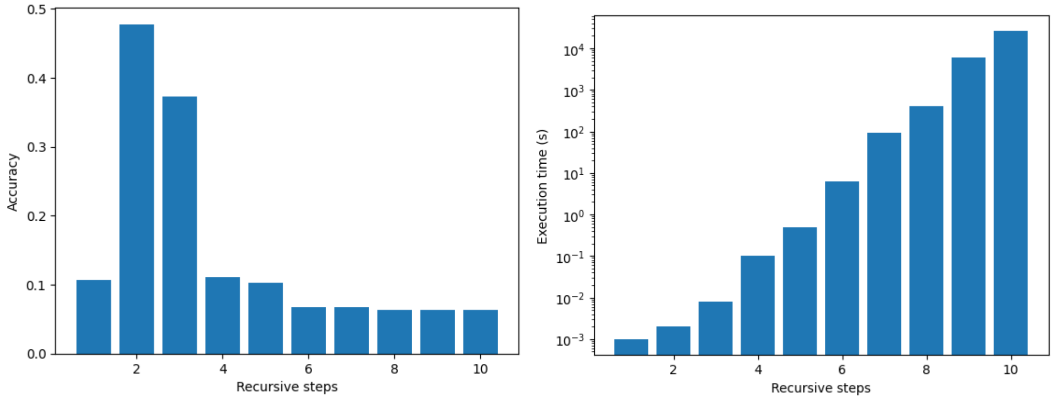
- $grid(r, volume)$ is a function that returns a list of blocks that result of partitioning volume (3D finite space) into blocks of size r.

- $shift(block, \epsilon)$ is a function that shifts all points of $block$ ϵ voxels. Blocks are translated in every axis. For each axis, shift is done in the positive direction of the axis if shifted block is still inside volume original shape. Otherwise block is shifted in negative direction of the axis.

- $fit(x, y)$ is a function that performs the least square lineal fit.

For this project the code is implemented in `python3` and has strong dependence of `numpy` library[4], that is used to perform the complex calculations.

[4] Official documentation can be found at `https://numpy.org/doc/stable/`



(a) (b)

Figure 2.4: Results computing BCD for Sierpinski carpet. In (a) accuracy is calculated with the difference of expected dimension ($\frac{\log 8}{\log 3}$) and computed dimension. In (b) execution time in seconds is ploted depending on the recursive steps. Note that in (b) y axis uses logarithmic scale.

Algorithm accuracy

In order to analyse the accuracy of box-counting dimension computations we must take into account the problem of memory size. The set used for accuracy estimation is self similar, because it is easy to generate and escalate and because we can compute precisely its box-counting dimension.

We expect to see an improvement in precision as we increase the number of points in the set. For this purpose we generated a 2-dimensional array containing the points of the set. The size of the array increases as we increase the number of steps used to generate it. This sense of improving accuracy comes from the fact that if we have more steps, the generated set will cover the fractal set with less surface, thus will be more similar the set. Also, as we have bigger images, we can perform more iterations for windows size, so we would have a more accurate regression estimation.

We have made computations for Sierpinski carpet (Figure 2.3). In order to improve memory usage an implementation of the algorithm for 2-dimensional images has been done. We expect memory usage and execution time to increase exponentially.

We have see that dimension has a better approximation when more recursive steps are used (see Figure 2.4). Although, approximation is not accurate as it gives an error of 0.062798 in the tenth iteration. Time, as memory usage, increases exponentially.

The technical limit for this experiment has been the memory usage. As we increase the size of the image it reserves more size in computer memory and when memory is exceeded we can not compute bigger images. A possible approach for this problem will be to free the memory during computation and looking to the image by parts. All in all, for our purposes, does not make sense to implement this dynamic algorithm as images used in CMR problem do not exceed the memory even for low profile computers.

2.3 Lacunarity

As we have seen in Theorem 2.12, *Haussdorff dimension* for self-similar sets can be computed directly from the contraction ratios of the similitudes that generate the fractal and the number of similitudes. So it is not difficult to generate sets with same kind of generators and exactly the same dimension.

For a more generalised cantor set, we can take n similarities with contraction ratios $r_1 = ... = r^n = n^{-2}$ and remove the even partitions in the $[0,1]$ interval. With this construction, different cantor sets C_j with same *Haussdorff dimension* are generated and $dim_H(C_j) = \log n / \log n^2 = 1/2$ for all j. At a glance, we can see that these sets are different, as they have different "holes". This example shows us that there is more information of the set that is not given by *Haussdorff dimension*. With the purpose of obtaining this information, B. B. Mandelbrot introduced the concept of *fractal lacunarity* in *The Fractal Geometry of Nature* [27] in order to represent the sense of how homogeneous was the set in relation to its holes. He extended this concept and provided an analytical approach in a later article [28] introducing some different definitions of the sense of *lacunarity*. In this section, we will focus on the defined as *shell lacunarity* as it has an efficient and intuitive computational approach introduced by C. Allain and M. Cloitre [2]. We will note it as the Brute Force Gliding-Box Lacunarity (BF GBL) and its domain are binary images. It was extended in [15] calling it Differential Gliding-Box Lacunarity (DGBL) with grey-level textured images domain. BF GBL was refined in [45] where Gliding-Box Lacunarity algorithm (GBL) is proposed. GBL provides a more efficient approach for n-dimensional sets. Finally, the Differential Lacunarity algorithm with fixed grid (DFGL) was proposed in [4].

As lacunarity depends on scale, objects that are heterogeneous at small scales can be homogeneous when examined at larger scales or vice versa. Thus, lacunarity can be considered a scale dependent measure of heterogeneity [32].

2.3.1 Lacunarity

Lacunarity is defined as a fluctuation of the mass distribution of a set. Given a set $F \subset \mathbb{R}^n$, we consider a δ-mesh U in $\mathbb{R}^n$. We define $m_F : U \to \mathbb{R}$ the mass application. This application should provide us a sense of how filled are the intersection of elements of U and F. Given a set F we define the mass application for $u \in U$ as $m_F(u) = \#(u \cap F)$ where # represents cardinality.

Let us define the distribution of mass of set F as $n(M, \delta)$, the number of δ-mesh cubes with mass M. With the mass application defined before, $n(M, \delta)$ is the number of elements $u \in U$ with $M = \#(u \cap F)$. This distribution can be converted into a probability distribution, $Q(M, \delta)$, dividing by $\#U$, i.e. number of cubes that cover F. $Q(M, \delta)$ represents the probability for a cube $u \in U$ of having mass M.

Now we compute the first and second moment of the mass probability distribution

$$Z_Q^{(1)} = \sum_M M \cdot Q(M, \delta)$$

$$Z_Q^{(2)} = \sum_M M^2 \cdot Q(M, \delta)$$

Definition 2.21. *We define shell lacunarity of a set F at scale δ, $\Lambda(\delta)$ as the mean-square deviation of the fluctuations of mass distribution probability $Q(M, \delta)$ divided by its square mean*

$$\Lambda(\delta) = \frac{Z_Q^{(2)}}{[Z_Q^{(1)}]^2} \tag{2.10}$$

From this definition, shell lacunarity takes a probability meaning since it can be interpreted as the width of the mass distribution function $Q(M, \delta)$. Note that for homogeneous sets $\Lambda(\delta) = 1$ and it is independent of δ. Sets with voids of all sizes are expected to be very lacunar with lacunarity much grater than 1, while sets with single-sized voids have low lacunarities close to 1 [2]. For textured images, homogeneous textures would have low lacunarities, close to 1, while lacunarity would increase as heterogeneity increases.

2.3.2 Computing lacunarity

This subsection will contain an explanation of the used algorithm (with pseudocode) and how we deal with dependence on δ.

Algorithm description

For this project we decided to employ an extended version of Gliding-Box Lacunarity algorithm (GBL). The algorithm was enhanced to enable computation of lacunarity within the context of 3-dimensional structures.

Consider an image of size $P \times Q \times R$ as a 4-dimensional spatial surface with (x, y, z) denoting voxel position on the image space, and the forth coordinate (t) denoting the voxel grey level. The (x, y, z) space is partitioned into overlapping blocks of size $\delta \times \delta \times \delta$, where δ is a divisor of $\min(P, Q, R)$. As in DBC, P, Q, R are required to be multiples of 2, so $\delta = 2^i$ for some $r > 1$, $r \in \mathbb{N}$. We will consider a distance of ϵ voxels between the overlapping blocks. On each grid we define boxes of size $\delta \times \delta \times \delta \times \delta'$ where $G \cdot \delta = \delta \cdot \min(P, Q, R)$ and G is the total number of grey levels i.e. $G = 255$.

Now we want to compute the mass of each block. Consider the (i, j, k)th block and let I_{min}, I_{max} be the minimum and maximum intensity grey level within this block. We define the mass function m_δ as done in section 2.2.3:

$$m_\delta(i, j, k) = \begin{cases} ceil(\frac{I_{max} - I_{min} + 1}{\delta'}), & I_{max} \neq I_{min} \\ 1, & I_{max} = I_{min} \end{cases} \tag{2.11}$$

Now we will compute the first and second order moments of the mas probability distribution. As all blocks cover the image, we consider n to be the total number of blocks. Note that

$$Z_Q^{(2)} = \sum_M M^2 \cdot Q(M, \delta) = \frac{\sum_M M^2 \cdot n(M, \delta)}{n} = \frac{\sum_{i,j,k} m_\delta(i, j, k)^2}{n}$$

$$Z_Q^{(1)} = \sum_M M \cdot Q(M, \delta) = \frac{\sum_M M \cdot n(M, \delta)}{n} = \frac{\sum_{i,j,k} m_\delta(i, j, k)}{n}$$

Note that in the last equality we changed summation ranges to iterate all blocks in the grid. Using the right hand equality we will compute the moments and applying (2.10) we have the lacunarity for a given δ.

Unlike the calculation of box-counting dimension, for lacunarity, we cannot apply a linear regression because it is scale dependent. To avoid this problem, in order to have a single value of lacunarity for every object, we computed the mean of lacunarity at different scales and considered this value as the lacunarity of the object.

The algorithm is presented in pseudocode in algorithm 2. Some notes are presented bellow:

- volume is the n-dimensional grey level image.

Algorithm 2: Differential Lacunarity algorithm

```
Input: volume
Data: $\epsilon$
$[P, Q, R] \leftarrow size(volume);$
$divisors \leftarrow divisors(\min\{P, Q, R\});$
foreach $\delta \in divisors$ do
    $\delta' \leftarrow 255 \cdot \delta / \min\{P, Q, R\};$
    $gird \leftarrow overlapped\_grid(r, volume, \epsilon);$
    $\Lambda(\delta) \leftarrow \frac{n \cdot \sum_{(i,j,k) \in grid} m_\delta(i,j,k)^2}{(\sum_{(i,j,k) \in grid} m_\delta(i,j,k))^2};$
end
$mean(\Lambda(\delta));$
```

- ϵ is a parameter and represents the distance between two blocks of the overlapped grid. In our case $\epsilon = 2$.
- $divisors(n)$ is a function that returns the divisors of a given number n.
- $overlapped\_grid(r, volume, \epsilon)$ is a function that returns a list of blocks that result in partitioning volume (3D finite space) into overlapping blocks of size r with distance ϵ between blocks.
- $mean(X)$ is a function that computes the mean for a set X.

For this project the code is implemented in `python3`. Complex computations such as *overlapped_grid* use `numpy` library, so it has a strong dependence on this library.

Chapter 3

Classification models

In this section the classification models employed in this work are presented offering the reader an in-depth comprehension of the working principles behind the adopted machine learning models. Furthermore, the concept of explainability in machine learning is introduced in this section. We provide an intuitive overview overview of SHapley Additive exPlanations (SHAP), a state-of-the-art intrepretability approach.

3.1 Support Vector Machines

Support Vector Machine (SVM) is a classification model that separates data of different classes using hyperplanes. The first approach demonstrated poor generalisation as it attempted to classify all the training data points without error. Further implementations presented by Cortes and Vapnik [13] suggested a soft-margin hyperplane that allows controlled error avoiding data overfit. Also, to encourage data separation there exists a non linear version based on "the kernel trick" [6]. In this kind of approach data is transferred to a higher dimensional space using a kernel function. In this new space we expect data to be linearly separable so we can use a hyperplane in the new space. This non linear approach is covered in this section.

The following description of the SVM model can be found with more detail in [5] (chapter 7). Extend used of Lagrange multiplier is used in this section, for an introduction on Lagrange multipliers see Appendix B or the reference book [5].

We consider a linear model to face the classification problem

$$y(x) = w^T \Phi(x) + b$$

where $\Phi(x)$ denotes the feature space transformation, $\Phi : \mathbb{R}^p \to \mathbb{R}^q$, $b \in \mathbb{R}$ is a bias parameter and $w \in \mathbb{R}^q$ determines the direction of the hyperplane. In general in order to improve data separation $p < q$. N input vectors $x_1, ..., x_n$, $x_i \in \mathbb{R}^p$ are considered with corresponding target values $t_1, ..., t_N$ where $t_n \in \{-1, 1\}$ and new data points x are classified according to the sign of $y(x)$. We define **margin** to be the smallest distance between the decision boundary (i.e. the hyperplane $0 = w^T\Phi(x) + b$) and any of the samples. The objective is to maximise this margin. The distance of any point x to the hyperplane is given by $|y(x)|/|w|$. As we want to classify correctly all points $t_n y(x_n) > 0$ for all n, thus distance is given by

$$\frac{t_n(w^T\Phi(x_n) + b)}{|w|}$$

We wish to optimise w and b to maximise the distance. Thus the maximum margin is found by

$$\arg\max_{w,b}\{\frac{1}{|w|}\min_n[t_n(w^T\Phi(x_n) + b)]\}$$

Now the problem is reformulated to an easier equivalent problem. Parameters are rescaled as $w \rightarrow \kappa w$ and $b \rightarrow \kappa b$ and the distance function keeps unaltered. We can use this freedom to set $t_n y(x_n) = 1$ for the closest point to the surface and $t_n y(x_n) \geq 1$ for the others. Now the problem is reduced to maximise $|w^{-1}|$ which is minimising $|w^2|$ subject with the constraint $t_n y(x_n) \geq 1$.

As we do not want to do an exactly classification to avoid overfit, slack variables $\xi_n \geq 0$ are introduced. These are defined by $\xi_n = 0$ for points that are correctly classified, and $\xi_n = |t_n - y(x_n)|$ for other points (including the margin). Now the non-exact classification constraints are

$$t_n y(x_n) \geq 1 - \xi_n \tag{3.1}$$

This is known as a soft margin, as allows points to be miss classified. Our goal is to maximise the margin while penalising points that lie on the wrong side of the margin boundary. We minimise

$$C\sum_{n=1}^{N}\xi_n + \frac{1}{2}|w|^2 \tag{3.2}$$

where $C > 0$ controls the trade-off between the slack variable penalty and the margin. We want to minimise (3.2) with constraints (3.1) and $\xi_n \geq 0$. Lagrange multipliers are used and results into

$$\mathcal{L}(w, b, \xi, a, \mu) = \frac{1}{2}|w|^2 + C\sum_{n=1}^{N}\xi_n - \sum_{n=1}^{N}a_n[t_n y(x_n) - 1 + \xi_n] - \sum_{n-1}^{N}\mu_n\xi_n$$

where $a_n \geq 0$ and $\mu_n \geq 0$ are Lagrange multipliers. The corresponding set of Karush-Kuhn-Tucker conditions (KKT conditions) are

$$\begin{aligned} a_n &\geq 0 \\ t_n y(x_n) - 1 + \xi_n &\geq 0 \\ a_n(t_n y(x_n) - 1 + \xi_n) &= 0 \\ \mu_n &\geq 0 \\ \xi_n &\geq 0 \\ \mu_n \xi_n &= 0 \end{aligned}$$

where $n = 1, ..., N$. We now optimise out w, b, and $\{\xi_n\}$ making use of the definition of $y(x)$

$$\begin{array}{l} \frac{\partial L}{\partial w} = 0 \Longrightarrow w = \sum_{n=1}^{N} a_n t_n \Phi(x_n) \\ \frac{\partial L}{\partial b} = 0 \Longrightarrow \sum_{n=1}^{N} a_n t_n = 0 \\ \frac{\partial L}{\partial \xi_n} = 0 \Longrightarrow a_n = C - \mu_n \end{array}$$

Using this results to eliminate w, b, $\{\xi_n\}$ from the Lagrangian, we obtain

$$\tilde{\mathcal{L}}(x) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(x_n, x_m)$$

where $k(x, x') = \Phi(x)^T \Phi(x')$ is called the kernel function. Constraints for this problem are $0 \leq a_n \leq C$, and $\sum_{n=1}^{N} a_n t_n = 0$ for $n = 1, ..., N$. This is again a quadratic programming problem. Now we minimise with respect to $\{a_n\}$.

With the given results, substituting into the original prediction formula, we have

$$y(x) = \sum_{n=1}^{N} a_n t_n k(x, x_n) + b$$

This results are now interpreted as follows. Points that have $a_n = 0$ do not contribute to the predictive model. The remaining data points (known as support vectors) have $a_n > 0$ and hence must satisfy

$$t_n y(x_n) = 1 - \xi_n$$

If $a_n < C$ then $\mu_n > 0$ which requires from KKT conditions $\xi_n = 0$ and hence such points are on the margin. If $a_n = C$ points are inside the margin and can be miss classifieds or not depending on $\xi_n > 1$.

To determine b we note that support vectors for which $0 < a_n < C$ have $\xi_n = 0$ so that $t_n y(x_n) = 1$ and hence will satisfy

$$t_n(\sum_{m \in S} a_m t_m k(x_n, x_m) + b) = 1$$

where S denotes the indexes of the support vectors.

Note that parameter C determines how soft is our margin. If $C \to \infty$ we have the case where all points must be correctly classified while if $C = 0$ we have no penalisation.

For this project the `sklearn` implementation of SVM classifier is used[1].

3.2 Random forest

Random forests were first introduced by L. Breiman in 2001 [7]. The aim of this ensemble method was to avoid problems given by decision trees. These are known to be unstable in the sense that if the data is perturbed slightly, the tree can change substantially [14]. Intuitively a bunch of trees are trained with a sub-sample of the data and a sub-sample of the features. Then the prediction is given by the aggregation of classifications of the different trees. This procedure solves the overfitting problem of the single tree classifier. It is also shown that adding more trees to the forest does not produce overfit [7].

Consider a p-dimensional random vector $X = (X_1, ..., X_p)^T$ representing the real-value input and a random variable Y representing the real-value response, we assume an unknown join distribution $P_{XY}(X, Y)$. The goal is to find a prediction function $f(X)$ for predicting Y. f is determined by a loss function $L(Y, f(X))$ and defined to minimise the expected value of the loss $E_{XY}(L(Y, f(X)))$ where the subscripts denote expectation with respect to the join distribution of X and Y. Intuitively , L is a measure of how close $f(X)$ is to Y. It penalises values of $f(X)$ that are a long way from Y. A typical choice for classification is zero-one loss function

$$L(Y, f(X)) = I(Y \neq f(X)) = \begin{cases} 0, & Y = f(X) \\ 1, & otherwise \end{cases}$$

If the set of possible values of Y is denoted by $\mathcal{Y}$, minimising $E_{XY}(L(Y, f(x)))$ for zero-one loss gives

$$f(x) = arg \max_{y \in \mathcal{Y}} P(Y = y | X = x)$$

also known as the Bayes rule [14].

In Random Forests f is constructed in terms of a collection of "base learners" $h_1(x), ..., h_J(x)$ they are combined as follows:

$$f(x) = arg \max_{y \in \mathcal{Y}} \sum_{j=1}^{J} I(y = h_j(x))$$

[1] Official scikit-learn documentation can be found at `https://scikit-learn.org/stable/`

Base learners are fitted using different training sets. The first step on training the Random Forest is to generate J independent samples form the original training data. This samples are known as bootstrap sample and each sample is used to train a different base learner. With this technique we introduce the first randomisation of the algorithm. As training data is different for every learner, the fitted trees will be different too.

The second randomisation is introduced when doing the split of a node. This split is performed in order to minimise the impurity of the child nodes (i.e. probability of each class in the sample nodes). This impurity can be computed using measures such as entropy and gini, E and G respectively, defined as

$$E = -\sum_j q_j log_2(q_j) \quad G = 1 - \sum_j q_j^2$$

where q_j is the probability if the class j.

For classification trees, we start with a root node. This node contains all observations. We select m features randomly from the p available features. The best binary split among all binary splits on the m predictors is found. Then the observations of the node are split using the best split found before and two child nodes are generated with this observations. The process is repeated recursively for all nodes until a stopping criterion is met.

For predictions, given a new point and based on the splits, for each tree in the forest, it follows the corresponding branch until a leaf node is reached. The leaf node will return as predicted label the majority label of its observations. Predicted label is, the majority class predicted by all trees in the forest.

About stopping criterion, we consider that it should stop when all the observations have the same label. This measure is known as purity of the node. Note that a node with a single observation is always pure so the iterated splitting is finite. In the worst case we will have as many leaf nodes as observations.

For this project the `sklearn` implementation of random forest classifier is used.

3.3 Extreme Gradient Boosting

Extreme Gradient Boosting (XGB) is a classification model introduced by T. Chen and C. Guestrin [12]. This model is based on tree ensemble. Unlike random forest, in XGB trees are combined one before each other, the tth tree improves learns from previous trees.

In this model, predictions are made by adding the prediction of base learners h_i. Consider a set with n observations and m features $\mathcal{D} = \{(x_i, y_i)\}$ where $|\mathcal{D}| = n, x_i \in \mathbb{R}^m, y_i \in \mathbb{R}$. The tree ensemble model uses K additive function to predict the output

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F}$$

where $\mathcal{F} = \{f(x) = w_{q(x)}\}$ $(q : \mathbb{R}^n \to T, w \in \mathbb{R}^T)$ is the space of classification trees. q represents the structure of each tree that maps an observation to the corresponding leaf index. T is the number of leaves in the tree. Each f_k corresponds to an independent tree structure. We will use q to, given an observation, classify it into the corresponding leaf and calculate the final prediction by summing up the score of all trees. Note that for regression trees, leaf nodes contain continuous scores, unlike decision trees exposed in section 3.2.

The loss function minimised is the following regularised objective.

$$L(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \tag{3.3}$$

where $\Omega(f) = \gamma T + \frac{1}{2}\lambda|w|^2$, $\gamma \in \mathbb{R}, \delta \in \mathbb{R}$, and l is a differentiable convex loss function that measures difference between the prediction and the target. Ω penalises the complexity of the model. Intuitively, the regularised objective will tend to select a model employing simple and predictive functions [12]. γ intuitively penalises the depth of trees as it is related to the number of leaves. λ penalises high weight values in leafs. This regularisation on trees can also be done using parameters on trees such as maximum depth.

The model includes functions as parameters in equation (3.3) and cannot be optimised using traditional methods. Instead the model is trained in an additive manner. Let $\hat{y}_i{}^{(t)}$ be the prediction of the i-th instance at the t-th iteration, the function f_t is added to minimise

$$L^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i{}^{(t-1)} + f_t(x_i)) + \Omega(f_t)$$

Second order approximation is used to quickly optimise the objective. The result is simplified by removing the constant terms. The following objective function at step t is obtained

$$\tilde{L}^{(t)} = \sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)] + \Omega(f_t) \tag{3.4}$$

where $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ and $h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}_i{}^{(t-1)})$ are first and second order gradient statistics of the loss function. Define $I_j = \{i | q(x_i) = j\}$ as the instance set of leaf j. We can rewrite (3.4) by expanding Ω as follows

$$\tilde{L}^{(t)} = \sum_{i=1}^{n}[g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)] + \gamma T + \frac{1}{2}\sum_{j=1}^{T} w_j^2 = \sum_{j=1}^{T}[(\sum_{i\in I_j} g_i) w_j + \frac{1}{2}(\sum_{i \in I_j} h_i + \lambda) w_j^2] + \gamma T$$

For a fixed structure $q(x)$, we can compute the optimal weight w_j^* of leaf j by

$$w_j^* = -\frac{\sum_i \in I_j g_i}{\sum_{i \in I_j} h_i + \lambda}$$

and calculate the corresponding optimal value by

$$\tilde{L}^{(t)}(q) = -\frac{1}{2}\sum_{j=1}^{T}\frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T$$

This last equation can be used as scoring function to measure the quality of a tree structure q, like impurity score for decision trees.

As it is impossible to enumerate all the possible tree structures q it employs a similar method as seen in section 3.2, but with the difference that it utilises the difference of loss between the child nodes and the parent node instead of impurity.

In order to avoid overfitting two additional techniques are used. The first is shrinkage. It scales newly added weights by a factor of η after each step of the tree boosting. Similar to a learning rate it reduces the influence of each individual tree and leaves space for future trees to improve the model. The second technique is feature sub sampling, as seen for random forests (section 3.2).

For this project the `xgboost` implementation of XGB classifier is used[2].

3.4 Explainability: SHAP

Machine learning models, use to be called "black box AI", referring to the fact that these models learn complex functions that are inaccessible and often incomprehensible to humans. The lack of explainability and transparency is hindering the implementation of machine learning in clinical workflows as artificial intelligence solutions can not be accepted without understanding of the underlying principles, even if the algorithms outperform experts [22]. For this reason machine learning model analysis should contain an explainability analysis.

SHAP (SHapley Additive exPlanations) is a game theory approach to explain the output of any machine learning model. It is based on Shapley values [39] from game theory and introduced in S. M. Lundberg et al. [26]. This method was proposed to provide explanations to the machine learning models. For each feature a sense of the classification impact is given. This method is implemented in a open library for multiple programming languages, in particular for python under the name `shap`[3].

[2] Official documentation can be found at `https://xgboost.readthedocs.io/en/stable/`

[3] Official library documentation can be found at `https://shap.readthedocs.io/en/latest/`

The idea of this method is to compute Shapley values for all test observations (or at least a subset that gives us confidence). This values can be aggregated to give a general view of the model or displayed individually. Computations are done using only the classifier interface and making modifications on the observation values.

Other explainability metrics, such as node impurity in tree bases model, depend exclusively to the model. Moreover, several studies have pointed out that node impurity might lack stability and be biased towards variables with many categories or numerical variables [43] [8]. This kind of metrics do not allow fair comparison among different models as they depend on internal model details. One advantage of SHAP method is that is agnostic to the model. Shapley values depend only on samples and its classification. For this reason internal implementation of model is ignored. This allow us to reproduce same analysis over different models with comparable results. To tackle these limitations, in this work SHAP was used to identify the most influential features in model's decision.

Chapter 4

Image feature extraction

In this section we will see the process of extracting features from the CMR images. The pipeline for data preprocessing follows the diagram of Figure 4.1. The first step is to understand and read the provided raw data. The next step is to deal with voxel anisotropy observed in CMR imaging. We conducted a study to evaluate the potential added value of interpolating the data. After interpolating (or not) data is normalised. The last step is to use the algorithms described in section 2.2.3 and 2.3.2 to compute the final features.

4.1 Dataset

In this study we used data from the United Kingdom Biobank (UKBB). UKBB is a large cohort study with over 500000 participants requited between 2006 and 2010 from across United Kingdom [35]. UKBB includes detailed CMR imaging of a subset of participants. Data from around 32K participants available to the Artificial Intelligence in Medicine laboratory of the Universitat de Barcelona where used in this study. Each patient's CMR images are provided as a 4-dimensional matrix consisting of 8-bit values. The three dimensions of the matrix represent spatial

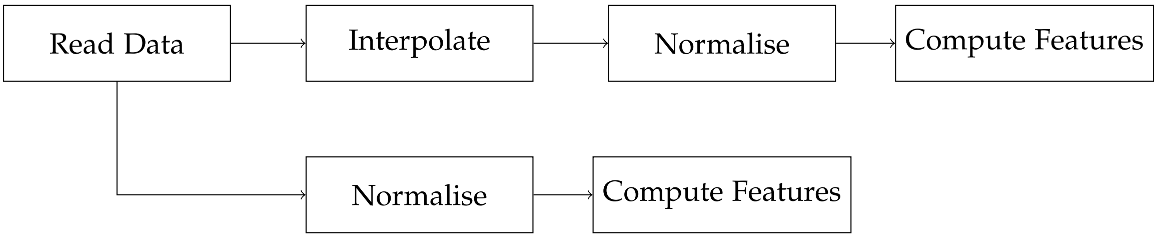


Figure 4.1: Flow diagram for feature extraction

coordinates, while the fourth dimension represents time.

UKBB CMR images are provided with segmentation of the three regions of interest (ROI): the left ventricular (LV), the right ventricular (RV) and the myocardium (M), as depicted in Figure 4.2. Segmentations are provided only for two specific time points end-diastole and end-systole, which are the two time-frames that cardiologists traditionally focus on. Segmentations were obtained manually for the first 5000 CMR scans using the protocol described in Peterson et al. [31]. This ground truth manual analysis data set was then used to develop a fully automated image analysis pipeline with inbuilt quality control, which has been applied to the remaining 27000 images. More details on the segmentation algorithm can be found at [3].

CMR data is provided in `.nii.gz` format, along with its metadata. Segmentation information is also provided in separate `.nii.gz` files. Information regarding which frame is end-systole and end-diastole has been provided by medical doctors from Queen Mary University.

4.2 Interpolation

4.2.1 The anisotropy problem

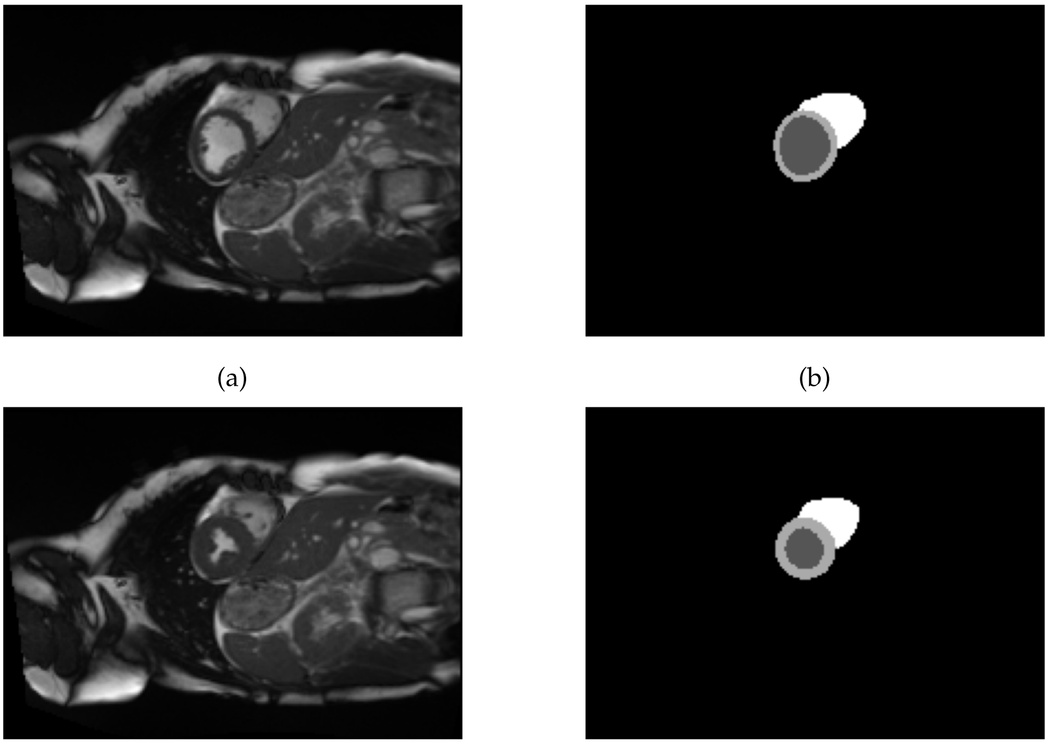
Due to the physical limitations involved in acquiring CMR data, discrepancies arise in the scaling of the various axes. In the spatial dimensions, each voxel at the x and y axes represents the same distance. However, voxels along z-axis represent higher physical distances. It should be noted that we refer to each plane in the x-y axes as a "slice" (see Figure 4.3a for slice model of segmentation). We consider an ordered set of slices as a "volume". For example, we have that a voxel represents $1,82692311mm$ in x, y dimension but represents $10mm$ in z dimension (as we can see in Figure 4.3a).

We considered two ways to approach this problem. Calculate the fractal features with anisotropic voxels, i.e. do not perform any transformation on the data, or generate the missing slices in z dimension with the aim that every voxel represents the same real-world distance along z as along x and y axis.

As we can see in figure 4.3b interpolating images and segmentation provide higher degree of smoothness, making it visually appealing to human observers.

4.2.2 Interpolating images

Let $f : \mathbb{N}^3 \to \mathbb{R}$ be the function that defines the grey level value for a given voxel. We consider $(x_0, y_0) \in \mathbb{N}^2$. Now we define $g : \mathbb{N} \to \mathbb{R}$ as $g(z) :=$



(a)

(b)

(c)

(d)

Figure 4.2: CMR slices and segmentation slices. (a) slice of end-diastole, (b) corresponding end-diastole segmentation. (c) slice of end-systole, (d) corresponding end-systole segmentation. All CMR images are from the same patient. For segmentation we have indicated RV (white), M (light grey) and LV (dark grey).

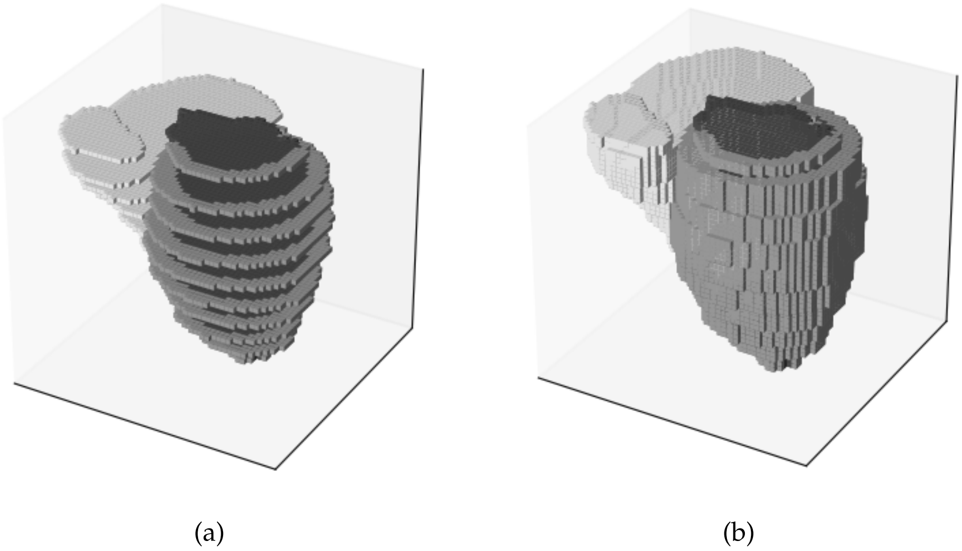


Figure 4.3: Reconstructions of 3D cardiac segmentations. (a) is an anisotropic segmentation, (b) is an interpolated segmentation. We have indicated RV (white), M (light grey) and LV (dark grey).

$f(x_0, y_0, z)$. We will interpolate g in order to compute the extra slices between two consecutive slices.

For solving this problem, we used Picewise Cubic Hermite Interpolating Polynomial, for 1-D and monotonic interpolation. This method was chosen because its minimum and maximum matches the minimum and maximum of the data and is monotonic over intervals where data is monotonic [34]. As we are working with images we do not want grey level values to exceed its domain, so interpolated points should no be larger or smaller than original maximum or minimum respectively.

To implement the Picewise Cubic Hermite Interpolation (PCHIP), we used the `scipy.interpolation` library[1]. This library provides us with a high-level interface that made it easier to work with this method. Given an image of size $P \times Q \times R$, for every pair $(x, y) \in P \times Q$ it will interpolate the function g using Cubic Hermite Splines matching z_k, $g(z_k)$ and $g'(z_k)$ where $R = \{z_0, ..., z_n\}$, $k = 0, ..., n$. $g'(z_k)$ are determined by the following method.

Let $h_k = z_{k+1} - z_k$, and $d_k = (g_{k+1} - g_k)/h_k$. d_k are the slopes at internal points z_k. If the signs of d_k and d_{k-1} are different or either of them equals zero, then

[1] Official library documentation can be found at `https://docs.scipy.org/doc/scipy/reference/interpolate.html`

$g'_k = 0$. Otherwise, it is given by the weighted harmonic mean

$$\frac{w_1 + w_2}{g'_k} = \frac{w_1}{d_{k-1}} + \frac{w_2}{d_k}$$

where $w_1 = 2 \cdot h_k + h_{k-1}$ and $w_2 = h_k + 2 \cdot h_{k-1}$.

Once derivatives are computed we interpolate each interval with a polynomial of degree three.

First of all, we should see how harmonic mean approximates the derivative. In our particular case we have equal distance between z_i for all i. So we can say $h_k = h, k = 0, ..., n$. Given $z \in z_0, ..., z_k$, from weighted harmonic mean formula, we have

$$\hat{g}'(z) = \frac{2(g(z) - g(z-h))(g(z+h) - g(z))}{h(g(z+h) - g(z-h))} \tag{4.1}$$

Now we want see how does this $\hat{g}'(z)$ behave when h is near 0. We start computing the Taylor expansion of g in $z + h$ and $z - h$.

$$g(z+h) = g(z) + h \cdot g'(z) + \frac{h^2}{2} g''(z) + \frac{h^3}{6} g'''(z) + O(h^4)$$

$$g(z-h) = g(z) - h \cdot g'(z) + \frac{h^2}{2} g''(z) - \frac{h^3}{6} g'''(z) + O(h^4)$$

We make the difference so we have

$$g(z+h) - g(z-h) = 2h \cdot g'(z) + \frac{h^3}{3} g'''(z) + O(h^4)$$

With the obtained results we compute the following operation

$$\frac{2(g(z) - g(z-h))(g(z+h) - g(z))}{h \cdot (g(z+h) - g(z-h))} = \frac{2h^2 g'(z)^2 + h^4(g'(z)g'''(z) - g''(z)) + O(h^5)}{2h^2 g'(z) + \frac{h^4}{3} g'''(z) + O(h^5)}$$

$$= \frac{2h^2 g'(z)^2 + h^4(g'(z)g'''(z) - g''(z)) + O(h^5)}{(2h^2 g'(z) + \frac{h^4}{3} g'''(z))(1 + O(h^5))}$$

Using the Taylor expansion $\frac{1}{1+x} = 1 - x + x^2 + ...$ we obtain

$$\frac{2h^2 g'(z)^2 + h^4(g'(z)g'''(z) - g''(z)) + O(h^5)}{2h^2 g'(z) + \frac{h^4}{3} g'''(z)} = \frac{2h^2 g'(z)^2 + h^4(g'(z)g'''(z) - g''(z)) + O(h^5)}{2h^2 g'(z)(1 + \frac{h^2}{6} \frac{g'''(z)}{g'(z)})}$$

Using again the previous Taylor expansion results into

$$\frac{2h^2 g'(z)^2 + h^4(g'(z)g'''(z) - g''(z)) + O(h^5)}{2h^2 g'(z)} \cdot ((1 + \frac{h^2}{6} \frac{g'''(z)}{g'(z)}) + O(h^4))$$

$$= g'(z) + h^2(\frac{g'''(z)}{6} - \frac{g''(z)^2}{g'(z)}) + O(h^3)$$

Thus

$$\hat{g}'(z) - g'(z) = h^2(\frac{g'''(z)}{6} - \frac{g''(z)^2}{g'(z)}) + O(h^3)$$

$$|\hat{g}'(z) - g'(z)| \leq C \cdot h^2, C \in \mathbb{R}$$

As in the left hand we have the difference of (4.1) and the derivative, we can say that the approximation decreases at least as h^2 when $h \to 0$.

On the other hand we have the problem about extreme values, as derivative $g'(z_k)$ requires information about z_{k-1} and z_{k+1}. For this cases the slopes are computed using the following one-side scheme. We want to compute $g'(z_0)$ and $g'(z_n)$, we will give the example for $g'(z_0)$, the other is analogous. First, we take $q = \frac{(2h_1+h_2)d_1-h_1d_2}{h_1+h_2}$, if the sign of q and d_1 are different, we will take $g'(z_0) = 0$ as we want zeros of derivative to be in no-interpolated points. If d_1 and d_2 have different sign and $|q| > |3d_1|$ we will take $g'(z_0) = 3d_1$, otherwise $g'(z_0) = q$.

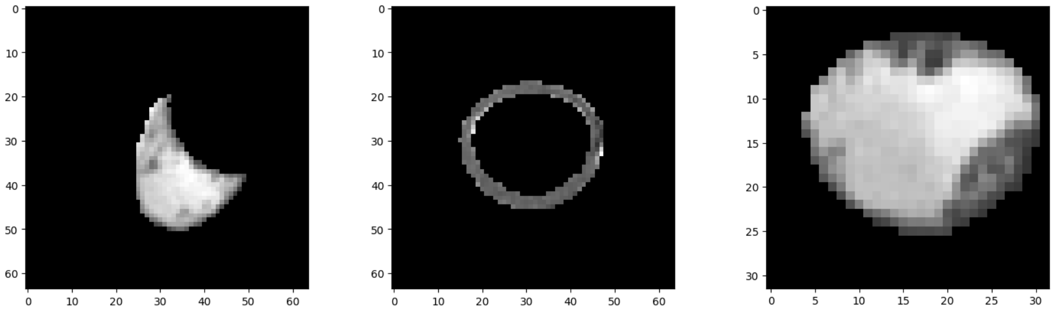
4.3 Normalisation

In order to minimise the variation in image signal intensities caused by the acquisition process, we conducted intensity normalisation on CMR images, within the heart region, using histogram matching [17] . For this purpose, we used one of the CMR from the data set as the reference.

4.4 Computing box-counting dimension and lacunarity

Before computing box-counting dimension and lacunarity, three volumes containing the texture of every ROI are created by using the hadamard product between CMR image and the corresponding cardiac structure segmentation(see Figure 4.4). Once ROIs are segmented from image, all non interest voxels are removed using the minimum box that contains this regions. With this method we reduce the voxels of every image to process so the used algorithms can perform faster. Therefore, for every CMR image in total we have six different volumes to process. This volumes correspond to RV, LF and M in end-systole and end-diastole frames. For each volume two features, box-counting dimension and lacunarity, are computed using algorithms explained in sections 2.2.3 and 2.3.2 respectively.

This process results in twelve different features for each patient. This set of features will be referenced in this thesis as fractal features (FF).



(a) (b) (c)

Figure 4.4: Slice of segmented ROIs in end-diastole. (a) RV , (b) M, (c) LV. Note that (c) axis is smaller than (a), (b).

The average time consumed for calculating all fractal-based features is 17.727 (±5.498)s. For the anisotropic features, the average time is 1.938(±0.462)s. Therefore, the calculation of anisotropic features is ten 9.143 faster. Algorithms 1 and 2 take much more time because grids will contain more blocks to be iterated. Also the number of used δ increases when the z axis has more resolution and hence $\min\{P, Q, R\}$ is larger. Thus will have, potentially, more divisors.

Chapter 5

Methodology

In this section we will explain the selection of our study sample, provide a comprehensive description of the the features used in the study for comparison purposes and perform an exploratory data analysis. Additionally, this section includes the details of the proposed machine learning pipeline.

5.1 Definition of the study sample

The first step was to assign each patient, whose CMR image was available, in the disease or control group.

In this study, we focused in the diagnostic[1] scenario of IHD as an illustrative example. However, it is important to note that the proposed pipeline is adaptable and applicableto the diagnosis of other diseases but also to disease prognosis. Table C.1 presents a summary of diagnosed diseases over the data set.

To define participants with a diagnosis of IHD we used ICD10[2] codes I20, I21, I22, I23, I24 and I25.

The diseased group includes all individuals who were diagnosed with one of the aforementioned code up until the date of imaging. As a control group of the study, we used all patients that have not had received a diagnose of IHD at the time of the imaging. Note that the union of this sets is equal to all the data set, so if a patient has diagnosis for a different disease is also added to control group. In plain words, control group does not contain only healthy patients.

[1] We encounter a diagnostic scenario when the CMR image is acquired after disease is diagnosed. In this scenario we can potentially find if a patient its currently suffering from a disease.

[2] International Classification of Diseases 10th revision.

5.2 Feature classification

We define three sets of features:

- Demographic features (**DF**): DF are demographic information. This features are age and sex.

- Vascular risk factors (**VRF**): VRFs features are selected based on biological plausibility and reported associations in the literature [33]. This set includes: current smoking, town send deprivation score, body mass index, body surface area, IPAQ score[3], alcohol intake and IPAQ group.

- Conventional CMR measures (**CMR**): This set contains features related to information extracted from CMR images and where provided by the group of Prof. Steffen Petersen from Queen Mary University, UK. This features have relation with the shape of the cardiac structures. The following conventional CMR indices were considered: LV end-diastolic volume (LVEDV), LV end-systolic volume (LVESV), LV stroke volume [4] (LVSV), LV ejection fraction [5] (LVEF), LV mass (LVM), RV end-diastolic volume (RVEDV), RV end-systolic volume (RVESV), RV stroke volume (RVSV), RV ejection fraction (RVEF), LV end-diastolic volume index[6] (LVEDVi), LV end-systolic volume index (LVESVi), LV stroke volume index (LVSVi), LV mass index (LVMi), RV end-diastolic volume index (RVEDVi), RV end-systolic volume index (RVESVi) and RV stroke volume index (RVSVi).

- Fractal features (**FF**): FF contains all features computed as described in section 4.4. Our fractal features have relation with the texture of the CMR images and thus with the tissue of cardiac estructures. The following features are considered: LV end-systolic BCD (LVESD), LV end-diastolic BCD (LVEDD), LV end-systolic lacunarity (LVESL), LV end-diastolic lacunarity (LVEDL), M end-systolic BCD (MESD), M end-diastolic BCD (MEDD), M end-systolic lacunarity (MESL), M end-diastolic lacunarity (MEDL), RV end-systolic BCD (RVESD), RV end-diastolic BCD (RVEDD), RV end-systolic lacunarity (RVESL) and RV end-diastolic lacunarity (RVEDL).

[3] International Physical Activity Questionnaire.

[4] Stroke volume is the volume of blood pumped from a ventricle per beat.

[5] Ejection fraction is the fraction of chamber volume ejected in systole in relation to the volume of the blood in the ventricle at the end of diastole.

[6] This index represents the relations among LVEDV and body surface area.

5.2.1 Feature combinations

In order to see if our novel fractal-based features add more information and improve classification models, we performed a detailed ablation study. More precisely, each of the models exposed in section 5.4 are trained using different feature combinations. It should be noted that DF features are always added as inputs as these variables impact the size and shape of the heart and are important for the models to know in order to provide a medical diagnosis. In summary we studied six feature combinations: DF+CMR, DF+FF, DF+VRF+CMR, DF+VRF+FF, DF+CMR+FF, DF+VRF+CMR+FF. DF+VRF are not considered as a features combination because these risk factors would not provide reliable information for CDV diagnose, although it could train a good performance model for this particular data set.

By performing this ablation study, we are able study the influence of each feature group in classification performance. In particular, we take special consideration on models where data is extended by FF comparing them to models that do not include fractal-based features to evaluate the potential added value of the inclusion of fractal-based features.

For the rest of the thesis we consider that all feature combinations contain DF although it is not written explicitly.

5.3 Exploratory analysis

5.3.1 Vascular Risk Factors and CMR measures

Subjects' characteristics are summarised in Appendix C.2. For the entire imaging data set, the average age was 63.27 (±7.54) while52% of the study participants are women. The sample consists of 32,096 participants, out of which 1592 have been diagnosed with IHD. Note the high imbalance in data set, participants with IHD diagnosis represent the 5.0% of all participants.

From CMR information, we can see that patients diagnosed with IHD have on average statistically significant higher LV volume index both in end-systole and end-diastole phase. However, LVSV index does not show a significant difference compared to the total population. For the RV, although average volume index in end-systole phase is higher, we can observe a lower mean for stroke volume index, which could mean that RV is not contracting as much as it should do.

5.3.2 Fractal Features

Subjects' fractal features are summarised in the Table 5.1.

As fractal features are computed from tissue texture, we will consider that dimension and lacunarity will give information about the complexity and homogeneity (respectively) of the tissue of the cardiac chambers of interest at the different timepoints of study.

LV tissue is more complex for participants in IHD group at end-systolic time. Nevertheless, it is also more homogeneous. The opposite behaviour is seen in end-diastole.

For the myocardium the opposite behaviour is observed. The tissue is on average less complex and more heterogeneous in end-systole and the inverse for end-diastole.

Despite differences on LV and M, RV features do not present significant difference in all cases. These differences are only observed for end-systolic box-counting dimension and for end-diastolic lacunarity. An interesting behaviour can be observed, mean end-systolic box-counting dimension is higher for the IHD group, while end-diastolic lacunarity decreases, not as LV and M.

LV appears to have the most complex tissue in both frames. Also show to be the less heterogeneous tissue.

The correlation among the features was studied using the PearsonÂ´s correlation coefficient. The results are provided in Figure 5.1. We observe that fractal features have a negative correlation among box-dimension and lacunarity for the same ROI at the same time. The reverse relationship between morphological complexity and heterogeneity has been previously described [20] [19]. Box-dimension and lacunarity of right ventricle at end-systole stands out with box-dimension and lacunarity of myocardium at end-diastole for being the most correlated pairs of fractal features.

About relation with CMR features, is seen that LVESD has positive correlation with all volume measures of LV and RV in both phases. It also has negative correlation with LVEF which means that as more complex is the LV end-systolic tissue less proportion of blood in LV is ejected.

In order to evaluate the impact of CMR image interpolation on fractal-based, we performed a Wilcoxon rank-sum test. The null hypothesis of the test is that two sets of measurements are drawn from the same distribution.

The statistical tests were performed using $\alpha = 0.05$ and it rejected the null hypothesis for almost all features pointing out the significant differences among fractal-based features when image is processed as explained in section 4.2.2. LV end-systolic lacunarity did not show statistical differences. All other features are statistically drawn from different distributions in the interpolated and anisotropic

Characteristics	WIS (n=32,096)	IHD (n=1592)
Anisotropic		
LVESD, mean (std)	2.26 (±0.13)	2.30 (±0.14)*
LVEDD, mean (std)	2.57 (±0.16)	2.54 (±0.17)*
LVESL, mean (std)	4.36 (±1.60)	4.25 (±2.07)*
LVEDL, mean (std)	3.65 (±2.27)	4.01 (±2.39)*
MESD, mean (std)	2.25 (±0.19)	2.22 (±0.21)*
MEDD, mean (std)	1.95 (±0.14)	1.97 (±0.12)*
MESL, mean (std)	4.92 (±3.53)	5.61 (±3.20)*
MEDL, mean (std)	7.06 (±2.19)	6.88 (±2.22)*
RVESD, mean (std)	2.08 (±0.18)	2.09 (±0.18)*
RVEDD, mean (std)	2.20 (±0.12)	2.20 (±0.12)
RVESL, mean (std)	7.11 (±4.45)	7.20 (±4.33)
RVEDL, mean (std)	6.37 (±1.80)	6.31 (±1.72)*
Interpolated		
LVESD, mean (std)	2.56 (±0.11)	2.58 (±0.11)*
LVEDD, mean (std)	2.82 (±0.11)	2.81 (±0.11)*
LVESL, mean (std)	4.34 (±1.49)	4.21 (±1.81)*
LVEDL, mean (std)	3.18 (±1.60)	3.42 (±1.73)*
MESD, mean (std)	2.66 (±0.11)	2.66 (±0.12)*
MEDD, mean (std)	2.41 (±0.10)	2.45 (±0.10)*
MESL, mean (std)	4.74 (±2.34)	5.32 (±2.69)*
MEDL, mean (std)	6.96 (±1.94)	6.77 (±1.97)*
RVESD, mean (std)	2.43 (±0.11)	2.45 (±0.11)*
RVEDD, mean (std)	2.61 (±0.07)	2.61 (±0.07)
RVESL, mean (std)	7.88 (±4.34)	7.91 (±4.31)
RVEDL, mean (std)	5.73 (±1.49)	5.69 (±1.47)*

"*" indicate statistical differences between the two populations using rank sum test for continuous values and chi-squared test for categorical variables ($\alpha < 0.05$).
Abbreviations: WIS Whole imaging set, IHD Ischaemic Heart Disease.

Table 5.1: Statistics of data comparing whole image set and population with Iscahemical Heart Disease for no-interpolated FF.

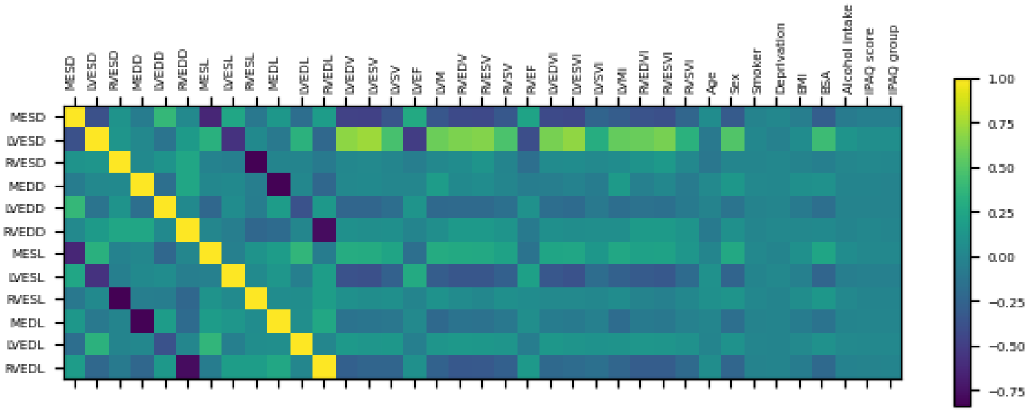


Figure 5.1: Correlation matrix for anisotropic fractal features and other used features.

cases. For this reason we will compute the experiments for both sets of features to ensure whether there is any difference during the classification step.

Although both samples are drawn from different distributions, the observed relation between all patients and diseased patients does not change. Nevertheless, the box-counting dimension provides higher values for interpolated images. Again, as expected from the feature correlation, lacunarity decreased when interpolation was performed.

5.4 Models

Three different models were trained and analysed: a SVM model, a Random Forest model and a XGB model. For all models hyper parameter tuning performed automatically by means of exhaustive search over predefined parameters and evaluated using a 5-fold cross validation (see section 5.4.1 for more detail). Over sampling and under sampling techniques have been tested. We tested SMOTE for over sampling, no balancing and random under sampling. Finally, the best results were achieved by random under sampling technique. This technique is present in the machine learning pipeline used in our study.

5.4.1 K-Fold Cross Validation

This technique is used to avoid overfitting and used to compute an averaged evaluation of the model for each parameter combination.

Given the learning set, it is split into k disjoint subsets of approximately equal

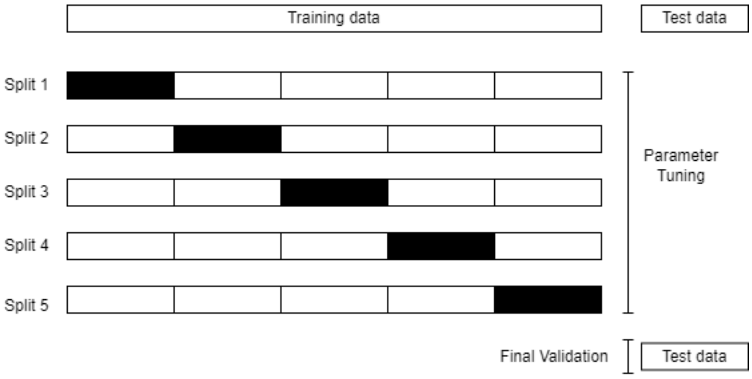


Figure 5.2: 5-fold cross validation schema. Training data is split into 5 folds. In each split black fold represents validation fold and white folds represent training folds.

size called folds. Now the model is trained using $k-1$ of the folds as training data. The resulting model is validated on the remaining fold and metrics are computed using this fold (see Figure 5.2). This process is repeated for each of the k folds.

This process is repeated for each parameter combination. With this technique we obtain an averaged metric for every combination so we can chose the optimal combination with less dependency on initial data split. 5-fold cross validation is only used for hyper parameter tuning.

Our study has a train/test split with relation 70/30 respectively. So 30% of the data set is used to asses models performance.

Note k-fold technique is only used for splitting training data into training and validation sets for hyper parameter tuning. Testing data set remains unaltered.

Split and train process is repeated seven times in order to obtain stable evaluation of the models.

5.4.2 Model A: SVM

This model uses Support Vector Machine (see section 3.1). The used kernel is the Radial Basis Function (RBF), this function is defined as follows:

$$rbf(x,x') = \exp(-\gamma|x-x'|^2)$$

Two parameters are considered for hyper parameter tuning: C and γ.

C represents the weight of penalisation for miss classified points as explained in section 3.1.

γ is a kernel parameter. Intuitively defines how far the influence of a single training example reaches, with low values meaning far and high values meaning close.

The values considered for each parameter can be found in Table 5.2.

C	$\{0.1, 0.5, 1, 5\}$
γ	$\{1, 0.5, 0.1\}$

Table 5.2: Values of SVM possible parameters. In each row presents the parameter and the set of possible values.

5.4.3 Model B: RF

This model uses Random Forests (see section 3.2). 100 base learners ($J = 100$) are used for each parameter combination. The parameters considered for hyper parameter tuning are: `max_depth`, `min_samples_split` and `bootstrap`.

`max_depth` represents the maximum depth of the trees. If `None`, nodes are expanded until stopping criterion is met.

`min_samples_split` parameter means the minimum number of samples required to split an internal node. If the number of samples of the node is less than `min_samples_split` it is considered that a stopping condition is met.

`bootstrap` indicates if bootstrap technique is used or not. If false, then the whole dataset is used to build each tree.

The values considered for each parameter can be found in the Table 5.3.

`max_depth`	$\{9, 5, 3, 2, None\}$
`min_samples_split`	$\{2, 3, 5, 7, 10\}$
`bootstrap`	$\{true, false\}$

Table 5.3: Values of Random Forest possible parameters. In each row presents the parameter and the set of possible values.

5.4.4 Model C: XGB

This model uses Extreme Gradient Boosting (see section 3.3). The parameters considered for hyper parameter tuning are: `learning_rate`, `min_child_weight`, `gamma`, `subsample`, `colsample_bytree` and `max_depth`.

`learning_rate` represents the factor η in section 3.3. It represents how feature weights shrinks after every boosting step.

`min_child_weight` for features of internal nodes, if the sum of weights is less than this value it is considered that a stopping condition is met.

`gamma` this value defines the minimum loss reduction required to make a further partition on a leaf node of the tree. If best loss reduction is smaller that this value a stopping condition is met.

`subsample` is the ratio of the training instances taken during bootstraping.

`colsample_bytree` is the ratio of features selected for training a new tree during the feature sub sample phase.

`max_depth` represents the maximum depth of the trees.

The values considered for each parameter can be found in the Table 5.4.

`learning_rate`	$\{0.01, 0.005, 0.1, 0.2\}$
`min_child_weight`	$\{1, 5, 10\}$
`gamma`	$\{0.5, 1, 5\}$
`subsample`	$\{0.6, 0.8, 1.0\}$
`colsample_bytree`	$\{0.6, 0.8, 1.0\}$
`max_depth`	$\{3, 4, 5\}$

Table 5.4: Values of XGB possible parameters. In each row presents the parameter and the set of possible values.

Chapter 6

Results

In this section experiment results are presented. At first we will define the used metrics in order to have a general view of which information they give. The obtained metrics of our experiments are summarised and analysed in this section. Finally feature impact in best model is discussed having special focus on impact of the introduced fractal features.

6.1 Evaluation metrics

In this section we will define the metrics used to evaluate and compare our models.

The prediction results of a models can be summarised in a confusion matrix. There results are presented in four sections: True positive (TP), true negatives (TN), false positives (FP) and false negatives (FN), see Figure (6.1).

The following metrics are used for assessment:

- **Balanced accuracy:** The balanced accuracy ratio $\frac{1}{2}(\frac{TP}{TP+FN} + \frac{TN}{TN+FP})$. This metric is an arithmetic mean of true positive rate and true negative rate. It is used to deal with imbalanced data sets. If classifier performs equally well on

		Predicted label	
		Positive	Negative
Real label	Positive	TP	FN
	Negative	FP	TN

Figure 6.1: Template of a confusion matrix. The following notation is used: True positive (TP), true negatives (TN), false positives (FP), false negatives (FN).

either class, this term reduces to the conventional accuracy (i.e. the number of correct predictions divided by total predictions).

- **Recall or True Positive Rate (TPR):** The recall is the ratio $\frac{TP}{TP+FN}$. It is, intuitively, the ability of the classifier to find all the positive samples.

- **Precision**: The precision is the ratio $\frac{TP}{TP+FP}$. It is, intuitively, the ability of the classifier not to label as positive a sample that is negative.

- **F1 score:** The F1 score can be interpreted as a harmonic mean of the precision and recall. Relative contribution of precision and recall to the F1 score are equal. The formula is: $2 \cdot \frac{precision \cdot recall}{precision + recall}$.

During training F1 score was optimised because it contains information of recall and precision. Has in consideration the ratio of well classified diseased patients and the ratio of wrong classified control patients. Nevertheless, we will also check those ratios independently. Finally we will also evaluate the general behaviour of our classifier by balanced accuracy metric.

During all the performance analysis `sklearn.metrics` library was used. It contains high level methods for computing desired metrics.

6.2 Models performances

In this section, detailed results on the performance of models are presented and analysed. The evaluation metrics are computed as exposed in section 6.1. For all models, experiments where conducted both for interpolated and no-interpolated features in order to determine if there is a feature extraction method that provides better performance on classification. All models have been trained seven times with different train/test split and results are summarised in terms of mean and standard deviation.

Data imbalance must be considered in order to interpret the obtained results. When dealing with imbalance results on precision can be very low [1].

6.2.1 Model A

The obtained results for model A are provided in Table 6.1.

We observe that overall balanced accuracy has values between 0.650 and 0.666. On the other hand we observe a poor performance with respect to precision. This is a normal behaviour in an imbalanced scenario because there are a lot of false cases that can lead to false positive classification. Finally we observe recall to reach

	CMR	FF	VRF+CMR	VRF+FF	CMR+FF	VRF+CMR+FF
Interpolated FF						
B. acc.	0.665 (±0.012)	0.656 (±0.013)	0.666 (±0.016)	0.666 (±0.015)	0.650 (±0.012)	0.657 (±0.016)
Rec.	0.686 (±0.030)	0.650 (±0.039)	0.691 (±0.031)	0.701 (±0.035)	0.680 (±0.031)	0.704 (±0.033)
Prec.	0.094 (±0.004)	0.095 (±0.004)	0.094 (±0.005)	0.093 (±0.005)	0.088 (±0.005)*	0.089 (±0.004)*
F1	0.166 (±0.007)	0.165 (±0.007)	0.166 (±0.008)	0.164 (±0.007)	0.156 (±0.008)*	0.158 (±0.007)*
Anisotropic FF						
B. acc.	0.665 (±0.012)	0.654 (±0.014)	0.666 (±0.016)	0.664 (±0.017)	0.650 (±0.012)	0.657 (±0.014)
Rec.	0.686 (±0.030)	0.648 (±0.040)	0.691 (±0.031)	0.701 (±0.047)	0.673 (±0.040)	0.702 (±0.031)
Prec.	0.094 (±0.004)	0.094 (±0.004)	0.094 (±0.005)	0.093 (±0.004)	0.089 (±0.005)*	0.089 (±0.004)*
F1	0.166 (±0.007)	0.164 (±0.007)	0.166 (±0.008)	0.164 (±0.007)	0.158 (±0.008)*	0.158 (±0.007)*

"*" indicate statistical differences between two populations using rank sum test ($\alpha < 0.1$). Populations compared are: CMR with CMR+FF and VRF+CMR with VRF+CMR+FF.
Abbreviations: VRF, vascular risk factors; CMR, cardio magnetic resonance; FF, fractal features.

Table 6.1: Ablation study for model A. Model is trained using different combination of features. Results, mean (std), are provided rounded to 3 decimals. Demographic features are considered in all feature combinations.

values between 0.648 and 0.704. This outcome does provide the sense that, in the best scenario, 70% of patients with IHD will be correctly classified.

In combinations with a single feature set, CMR has the best performance in both balanced accuracy and recall. We observe that shape related features and texture related features have similar performances regarding to balanced accuracy.

Significantly differences have been found in CMR compared to CMR+FF results for precision and F1 score. CMR+FF combination yield the poorest performance and represents the worst results in all feature combination. Extending the feature set might be reducing the separability of classes.

Recall has been incremented in VRF+CRM+FF without significant results, but the mean performance decreased significantly by 5.6% in precision and by 4.8% in F1 score. As for CMR+FF feature extensin might be reducing separability of features.

Relation among interpolated and anisotropic FF, VRF+FF, CMR+FF and VRF+CMR+FF does not show statistical differences ($\alpha < 0.1$).

6.2.2 Model B

The obtained results for model B are provided in Table 6.2.

We observe that balanced accuracy has values between 0.660 and 0.672, improving the model A minimum and maximum of this metric. As observed in model A, performance with respect to precision is poor. Best recall performance (0.687 for VRF+CMR) does not reach the maximum recall of model A (0.704 for

	CMR	FF	VRF+CMR	VRF+FF	CMR+FF	VRF+CMR+FF
Interpolated FF						
B. acc.	0.662 (±0.013)	0.662 (±0.015)	0.672 (±0.014)	0.669 (±0.013)	0.663 (±0.013)	0.668 (±0.017)
Rec.	0.662 (±0.040)	0.654 (±0.033)	0.687 (±0.030)	0.683 (±0.032)	0.670 (±0.029)	0.681 (±0.031)
Prec.	0.096 (±0.006)	0.097 (±0.004)	0.098 (±0.006)	0.097 (±0.005)	0.096 (±0.005)	0.097 (±0.006)
F1	0.168 (±0.008)	0.169 (±0.006)	0.172 (±0.009)	0.170 (±0.008)	0.167 (±0.007)	0.169 (±0.009)
Anisotropic FF						
B. acc.	0.664 (±0.012)	0.660 (±0.009)	0.671 (±0.014)	0.667 (±0.010)	0.664 (±0.013)	0.667 (±0.012)
Rec.	0.673 (±0.030)	0.661 (±0.028)	0.682 (±0.032)	0.675 (±0.026)	0.673 (±0.028)	0.683 (±0.036)
Prec.	0.096 (±0.007)	0.095 (±0.003)	0.098 (±0.004)	0.097 (±0.003)	0.096 (±0.005)	0.096 (±0.004)
F1	0.168 (±0.010)	0.166 (±0.004)	0.171 (±0.007)	0.170 (±0.004)	0.168 (±0.008)	0.168 (±0.007)

"*" indicate statistical differences between two populations using rank sum test ($\alpha < 0.1$). Populations compared are: CMR with CMR+FF and VRF+CMR with VRF+CMR+FF.
Abbreviations: VRF, vascular risk factors; CMR, cardio magnetic resonance; FF, fractal features.

Table 6.2: Ablation study for model B. Model is trained using different combination of features. Results, mean (std), are provided rounded to 3 decimals. Demographic features are considered in all feature combinations.

VRF+CMR+FF).

For combinations of one feature set, CMR has slightly better performance than FF. CMR has improved FF balanced accuracy by 0.6%. This result suggests that similar information can be found during classification for shape and texture related features.

Extending CMR features with FF results has almost same performance of CMR model. Nevertheless, this combination achieved better performance than FF. No significant differences where found among CMR and CMR+FF.

VRF+CMR feature combination yielded the overall best performance, in this model and in the whole experiment, in terms of balanced accuracy, with value of 0.672.

Extending VRF+CMR with FF do not show any significant improvement for this model.

As observed in model A, interpolation only produce slight changes when compared to anisotropic data, without statistically significant differences ($\alpha < 0.1$).

6.2.3 Model C

The obtained results for model C can be found in the Table 6.3.

We observe that balanced accuracy has values between 0.662 and 0.672. Thus this model has improved the minimum balanced accuracy when compared to model A and B.

CMR and FF feature sets have almost the same performance both in interpo-

	CMR	FF	VRF+CMR	VRF+FF	CMR+FF	VRF+CMR+FF
Interpolated FF						
B. acc.	0.664 (±0.011)	0.664 (±0.014)	0.671 (±0.010)	0.669 (±0.013)	0.662 (±0.014)	0.671 (±0.014)
Rec.	0.673 (±0.030)	0.667 (±0.042)	0.681 (±0.031)	0.679 (±0.026)	0.663 (±0.033)	0.687 (±0.026)
Prec.	0.096 (±0.005)	0.096 (±0.006)	0.098 (±0.005)	0.098 (±0.006)	0.096 (±0.006)	0.097 (±0.006)
F1	0.168 (±0.007)	0.168 (±0.009)	0.171 (±0.007)	0.171 (±0.010)	0.167 (±0.009)	0.170 (±0.009)
Anisotropic FF						
B. acc.	0.664 (±0.011)	0.662 (±0.012)	0.671 (±0.010)	0.670 (±0.017)	0.663 (±0.012)	0.672 (±0.013)
Rec.	0.673 (±0.030)	0.660 (±0.035)	0.681 (±0.031)	0.680 (±0.030)	0.666 (±0.033)	0.679 (± 0.036)
Prec.	0.096 (±0.005)	0.096 (±0.004)	0.098 (±0.005)	0.098 (±0.006)	0.096 (±0.004)	0.099 (±0.005)
F1	0.168 (±0.007)	0.168 (±0.006)	0.171 (±0.007)	0.171 (±0.009)	0.167 (±0.006)	0.172 (±0.008)

"*" indicate statistical differences between two populations using rank sum test ($\alpha < 0.1$). Populations compared are: CMR with CMR+FF and VRF+CMR with VRF+CMR+FF.
Abbreviations: VRF, vascular risk factors; CMR, cardio magnetic resonance; FF, fractal features.

Table 6.3: Ablation study for model C. Model is trained using different combination of features. Results, mean (std), are provided rounded to 3 decimals. Demographic features are considered in all feature combinations.

lated and anisotropic fractal-based features. Combination of these features does not improve performance.

Among the various models tested, model C demonstrates the highest performance when using the FF feature set in conjunction with VRF and CMR, employing anisotropic fractal-based features. This model has 0.672 score in balanced accuracy, the same value obtained by the other best model found in model B. This model is thoroughly analyzed in the section 6.3.

Results of VRF+CMR+FF with anisotropic FF slightly improve VRF+CMR results in balanced accuracy, precision and F1 score, but no statistically significant differences where found among their metrics.

Analogously to the previous models, interpolation of features does not result in statistically significant changes ($\alpha < 0.1$).

6.3 Explainability analysis

In this section SHAP (see section 3.4) analysis for model C using VRF+CMR+FF with anisotropic FF is exposed. We selected this model for further analysis as it is the model with best performance that contains fractal-based features.

For this analysis shapley values are calculated for all participants in test splits and classified using its corresponding classifier. Summary plots are created using mean shapley values of each participant independently to the classifier where shapley values come from.

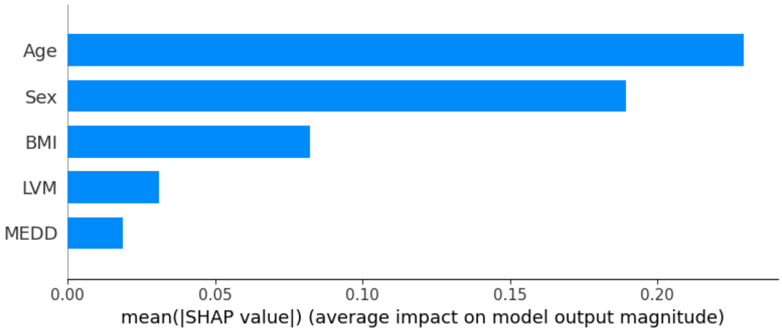


Figure 6.2: Top five feature contribution for model C trained with VRF+CMR+FF. Feature contribution is considered using mean of absolute value of shapley values.

Analysing shapley values for VRF+CMR+FF feature set we observe that both demographic features, age and sex, are the most important with huge difference (see Figure 6.2). The third feature in order of influence is BMI, from VRF. This feature has 37.8% more influence than the next feature in order of mean shapley values. These three feature are with huge difference the more influential features, and for this reason models including VRF have a good performance.

For CMR features, LVM has the fourth position in order of importance. The rankings in positions six, eight and ten pertain to LVMi, LVESVi and RVEDVi, respectively.

The fifth feature in order of importance is MEDD, form the fractal-based feature set. This fact suggests that myocardium tissue during end-diastole has an important impact in diagnose of IHD. We already know that myocardium geometry and tissue have an important role in IHD diagnose [36] and our results make sense with this previous observation.

Considering the influence of a feature during the prediction depending to its value, we can observe (Figure 6.3) that low complexity of myocardial end-diastolic tissue has a negative shapley value. This can be interpreted as low MEDD produces non-diseased classification. This fact seem coherent with exploratory analysis from section 5.3.2 where we observed that IHD group has significantly higher mean for this feature. Average and high values tend to have influence on positive class (i.e. classify as diseased of IHD).

Shapley values have been grouped by feature sets to see the impact of every set in the classification task compared with other feature groups. Results are presented in Figure 6.4. From observations of top five features, dominance of demographic features was expected because these features were the most important for the classifier. Similarly VRF has a huge impact because BMI is the third feature with more

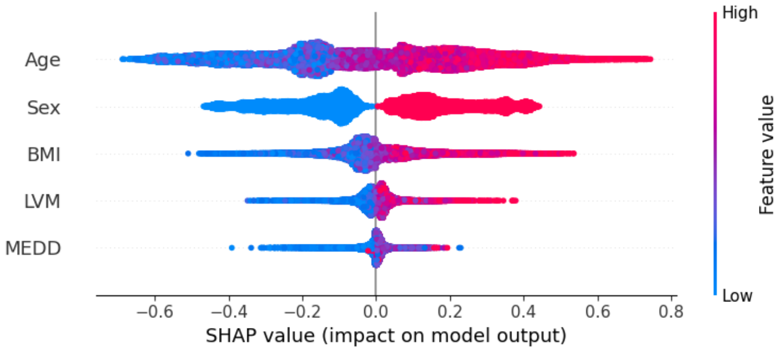


Figure 6.3: Top five feature contribution for model C trained with VRF+CMR+FF. Shap values are represented as a dot for each participant. Horizontal axis represent the shap value and colour represents the values of the feature for each observation.

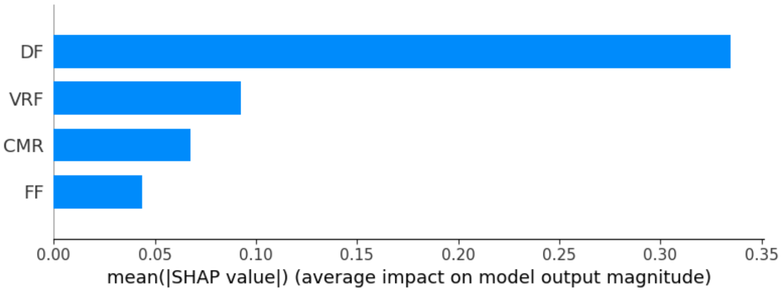


Figure 6.4: Overall contribution of demographic features (DF), vascular risk factors (VRF), cardio magnetic resonance indexes (CMR) and fractal-based features (FF).

influence. BMI importance has a huge difference with LVM and less important features.

Fractal-based features are the less important features. It represents a 35, 3% less influence than CMR features. Then we conclude that aggregated fractal-based features are not as influential as aggregated CMR features.

As suggested in Rauseo et. al [36] left ventricle and myocardium plays an important role in IHD classification. Similar behaviour has been observed in our experiment (Figure 6.5). We observed that myocardium-related fractal-based features have higher aggregated impact than LV or RV. We also observed that tissue complexity and homogeneity of LV has more impact than RV.

Tissue complexity was observed to be more influential than homogeneity. Box-counting dimension has, aggregated, 72.0% more impact in classification than lacunarity (see Figure 6.6).

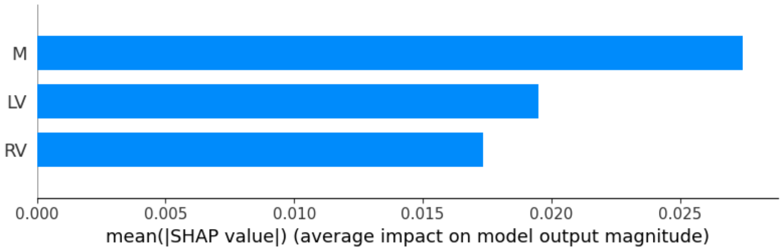


Figure 6.5: Overall contribution of fractal-based features grouped by cardiac structure. Abbreviations: Myocardial, M; left ventricle, LV; right ventricle, RV.

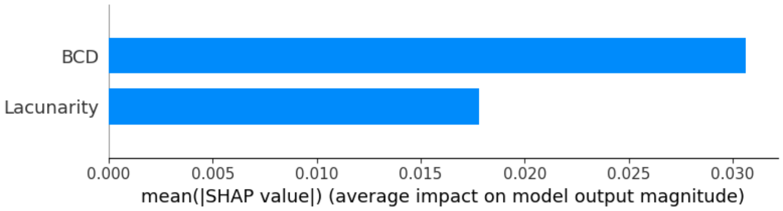


Figure 6.6: Overall contribution of fractal-based features grouped by fractal metric. Abbreviations: Box-counting dimension, BCD.

Chapter 7

Conclusions

This section is divided into two parts. First, the conclusions gathered during the process of elaborating the work are summarised. In the second part, possible extensions to the work are presented.

7.1 Project conclusions

The aims of this project included presenting a novel set of CMR texture features based on fractal theory that allow us to describe the cardiovascular tissue and implementing algorithms to extract this information from CMR, the reference modality in the evaluation of the cardiac function and structure. Furthermore, they also included the study, development and validation of machine learning models that allow us to assess the potential contribution of fractal analysis of CMR images to the diagnosis of IHD.

We successfully achieved these objectives by devising a two-step process. Firstly, we extracted the features from a given CMR image set. To this end, we implemented efficient algorithms (less than 2s to process a 3D CMR) that can be applied in large databases. On second step, these features were employed, along with other image parameters and clinical information of the patient, to train machine learning models for diagnostic purposes. Although for this thesis we focused in IHD, this two-step process provides a framework for further investigations, covering different CVDs.

Regarding to the feature extraction, we provided methods that quantify from texture of CMR images. Related studies use to work with shape related fractal-based features [10][18][20]. We contributed with texture information to complement traditional CMR index, which focus only on shape.

In addition, a comparative analysis between anisotropic and interpolated fractal-based features has been conducted. As detailed in section 4.4, it is evident that the computation of interpolated features needs significantly more time. However, the results presented in section 6 indicate no significant differences between these two approaches. Consequently, we conclude that interpolation is unnecessary, and the calculation of fractal-based features in anisotropic form is advantageous in terms of time efficiency during the feature extraction process.

As outlined in section 6, one of the models that yielded the best classification results incorporates fractal information. Statistical differences where not observed in the performance of models combining VRF, CMR and FF so we can not determine if there is a significantly better combination. Differences between models are very small and a more extended amount of experiments should be performed in order to find this differences, if they exist.

We observe that shape related features (CMR) and texture related features (FF) yield similar performances in classification with slight better outcomes for shape related features. All in all, results suggest that fractal-based features might have similar performance as traditional CMR indexes.

For our best fractal-based model, and through our analysis, we were able to assess the contribution of fractal analysis to the task of patient classification, and it was observed that the set of fractal parameters has lower influence on this task compared to CMR indexes.

A notable result from the study is that myocardial end-diastolic tissue complexity is on of the top five influential characteristics for IHD diagnose. We observed that low complexity tissues in this region of interest tends to indicate non-diseased patients. MEDD has more importance in classification than the other used CMR indexes (except form LVM). This suggests that could be a potential novel biomarker for IHD diagnose.

7.2 Further work

The present work leaves open several lines of research that are yet to be explored. Some are suggested for future work on this field:

- Exploring the Relationship between Fractal Features and Radiomics [47]. The term Radiomics include shape and texture-related features. Future work will involve a detailed study of Radiomics versus and in combination with fractal-based features for disease diagnosis.

- Investigate the potential of the propsoed pipeline based on machine learning and fractals in ischemic disease prognosis. In contast with the diagnostic scenario studied in this work, prognosis involves the identification of people potential at risk of developing the diasese

- Study if fractal features can provide more relevant information on tissues with different diseases (in diagnostic or prognostic scenario) such as atrial fibrillation, heart failure, myocardial infarction or stroke.

- Perform a more detailed analysis of distribution of heart fractal-based features among different samples of population clustering by sex, age, country, ethnic background, ... to obtain better insights into the characteristics of those sub-populations in terms of cardiac complexity and heterogeneity.

Appendix A

Fractal sets

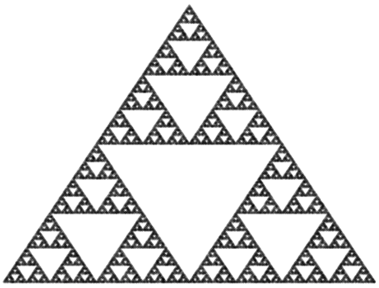


Figure A.1: Sierpinski triangle.

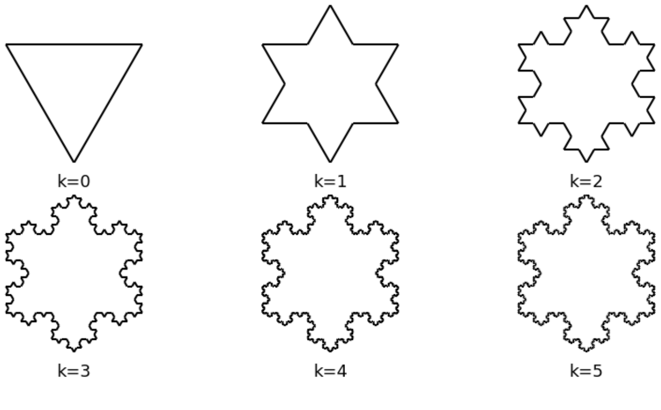


Figure A.2: Construction of von Koch's snowflake. k represents the recursive step of the iteration.

Appendix B

Lagrange multipliers

This section introduces the use of Lagrange multipliers. Concepts on this section are from [5] (Appendix E).

Lagrange multipliers are used to find the stationary points of a function of several variables subject to one or more constraints. Consider the problem of finding the maximum of a function $f(x_1, x_2)$ subject to a constraint relating x_1 and x_2, written in the form $g(x_1, x_2) = 0$.

To solve this problem we introduce the parameterλ, called Lagrange multiplier. Consider a n-dimensional variable x with components $x_1, ..., x_n$. The constraint equation $g(x) = 0$ then represents a $(n-1)$-dimensional surface in x-space.

Note that for any point on the constraint surface, the gradient $\nabla g(x)$ will be orthogonal to the surface. Consider x such that $g(x) = 0$, and consider a nearby point $x + \epsilon$ that also lies on the surface. If we make Taylor expansion around x, we have

$$g(x+\epsilon) \simeq g(x) + \epsilon^T \nabla g(x)$$

Because both x and $x + \epsilon$ lie on the constraint surface, we have $g(x) = g(x + \epsilon)$ and hence $\epsilon^T \nabla g(x) \simeq 0$. In the limit $|\epsilon| \to 0$ we have $\epsilon^T \nabla g(x) = 0$, and because ϵ is then parallel to the constraint surface, we see that the vector ∇g is normal to the surface.

We seek for a point x^* on the constraint surface such that $f(x)$ is maximised. Such point must have $\nabla f(x^*)$ orthogonal to the constraint surface, otherwise we could increase the value of $f(x)$ by moving a short distance along the surface. Thus ∇f and ∇g are parallel vector there must exist λ such that

$$\nabla f + \lambda \nabla g = 0$$

where $\lambda \neq 0$.

We introduce then the Lagrangian function $L(x, \lambda) = f(x) + \lambda g(x)$. (B) is obtained by setting $\nabla_x L = 0$ and $g(x) = 0$ is obtained by setting $\frac{\partial L}{\partial \lambda} = 0$. To find the maximum of $f(x)$ with $g(x) = 0$ we find the stationary point of $L(x, \lambda)$ with respect both x and λ. It leads to $n + 1$ equations.

Consider now the problem for inequality constraint $g(x) \geq 0$. There are to kind of solutions, whether the stationary points lie in the region $g(x) > 0$ or in $g(x) = 0$.

If $g(x) > 0$, then $\nabla f(x) = 0$, hence $\lambda = 0$. If $g(x) = 0$ we are in the previous case. Note that the sign of λ plays a crucial role as $\nabla f(x)$ must be oriented away from the region $g(x) > 0$ to be a maximum. Therefore, we have $\nabla f(X) = -\lambda \nabla g(x)$ for $\lambda > 0$. In both cases $\lambda g(x) \geq 0$ so the conditions for optimising are $g(x) \geq 0$, $\lambda \geq 0$ and $\lambda g(x) = 0$. These are knows as the Karush-Kuhn-Tucker (KKT) conditions.

Note that if we are seeking for minimisation of $f(x)$ we can apply the same method using Lagrange function $L(x, \lambda) = f(x) - \lambda g(x)$.

This method can be extended to multiple equality and inequality constraints. Consider we want to maximise $f(X)$ subject to $g_j(x) = 0, j = 1, ..., J$ and $h_k(x) \geq 0, k = 1, ..., K$. The we introduce Lagrange multipliers $\{\lambda_j\}$ and $\{\mu_k\}$ and optimise the function

$$L(x, \{\lambda_j\}, \{\mu_k\}) = f(x) + \sum_{j=1}^{J} \lambda_j g_j(x) + \sum_{k=1}^{K} \mu_k h_k(x)$$

subject to $\mu_k \geq 0$, $\mu_k h_k = 0$ for $k = 1, ..., K$.

Appendix C

Tables

C.1 Population's disease summary

In this appendix population's disease is summarised in a table. The number of patients classified for each ICD10 code are shown separated in two scenarios, prognostic and diagnostic. The patients are not unique, which means that if a patient has suffered a heart failure and myocardial infarction, which correspond to I50 and I21 respectively, he/she will be counted in both rows.

C.2 Subjects' characteristics

In this appendix subjects' characteristics are summarised in a table. Shown data correspond to demographic features, vascular risk factors (Table C.2) and CMR features (Table C.3). This table complements information given in Table 5.1.

Disease	ICD10 code	Count Diagnostic	Count Prognostic
Ischemic heart disease	I20	917	375
	I21	440	217
	I22	38	0
	I23	3	0
	I24	92	74
	I25	1197	663
Healthy	-	28146	28146

Table C.1: Counting patients table. Patient count is not unique among ICD10 codes. Healthy patients are considered to be patients without any of the listed ICD10 code in diagnostic or prognostic scenario.

Characteristics	Whole imaging set (n=32,096)	IHD (n=1592)
Demographic features		
Age, mean (std)	63.27 (±7.54)	67.62 (±6.44)*
Sex female, n (%)	16591 (52%)	440 (28%)*
VRF		
No smoker, n(%)	29977 (94%)	1458 (92%)*
Deprivation, mean (std)	-1.96 (±2.67)	-1.81 (±2.80)
BMI (kg/m^2), mean (std)	26.61 (±4.24)	28.09 (±4.30)*
BSA (m^2), mean (std)	1.87 (±0.20)	1.95 (±0.20)*
IPAQ score, mean (std)	2278.94 (±2357.40)	2278.83 (±2380.65)
Alcohol intake, n (%)		
Never	1521 (5%)	97 (6%)
Special occasions only	2633 (8%)	141 (9%)
One to three times a month	3439 (11%)	169 (11%)
Once or twice a week	8254 (26%)	403 (25%)
Three or four times a week	9058 (28%)	426 (27%)
Daily or almost daily	7080 (22%)	356 (22%)
IPAQ group, n (%)		
Group 1	7293 (23%)	396 (25%)*
Group 2	16539 (52%)	775 (49%)*
Group 3	8076 (25%)	415 (26%)*

"*" indicate statistical differences between the two populations using rank sum test for continuous values and chi-squared test for categorical variables ($\alpha < 0.05$).

Table C.2: Statistics of data comparing whole image set and population with Iscahemical Heart Disease for DF, VRF features.

Characteristics	Whole imaging set (n=32,096)	IHD (n=1592)
CMR		
LVEDV (*ml*), mean (std)	148.01 (±33.82)	158.62 (±38.85)*
LVESV (*ml*), mean (std)	60.58 (±19.41)	68.42 (±27.11)*
LVSV (*ml*), mean (std)	87.42 (±19.22)	90.20 (±19.41)*
LVEF (%), mean (std)	59.48 (±6.12)	57.72 (±7.61)*
LVM (*g*), mean (std)	86.39 (±22.47)	95.77 (±23.75)*
RVEDV (*ml*), mean (std)	156.49 (±37.23)	161.92 (±35.50)*
RVESV (*ml*), mean (std)	67.68 (±21.35)	70.95 (±20.73)*
RVSV (*ml*), mean (std)	88.81 (±20.21)	90.97 (±20.29)*
RVEF (*ml*), mean (std)	57.20 (±6.16)	56.52 (±6.54)*
LVEDVi (ml/m^2), mean (std)	78.63 (±13.89)	81.22 (±16.93)*
LVESVi (ml/m^2), mean (std)	32.08 (±8.68)	34.92 (±12.67)*
LVSVi (ml/m^2), mean (std)	46.55 (±8.34)	46.30 (±8.62)
LVMi (g/m^2), mean (std)	45.66 (±8.70)	48.85 (±9.53)*
RVEDVi (ml/m^2), mean (std)	83.04 (±15.20)	82.90 (±14.81)
RVESVi (ml/m^2), mean (std)	35.78 (±9.35)	36.22 (±9.17)*
RVSVi (ml/m^2), mean (std)	47.26 (±8.76)	46.67 (±8.98)*

"*" indicate statistical differences between the two populations using rank sum test for continuous values and chi-squared test for categorical variables ($\alpha < 0.05$).

Table C.3: Statistics of data comparing whole image set and population with Iscahemical Heart Disease for CRM features.

Appendix D

Statistical tests

D.1 Rank sum test

During the experiments rank sum test is used for determining if two samples are drawn from the same distribution. We used this test because it is a non parametric test, which means that we do not have to assume a fixed distribution for our data. During experiments and analysis, `scipy.stats.ranksum` feature of `scipy` library[1] was used as it provides a high level methods that implement this test.

In this section rank sum test will be introduced and its computation explained. Test definition comes from S. M. Ross' statistics book [37] chapter 12.

Let $X_1, ..., X_n$, $Y_1, ..., Y_m$ denote two independent samples. F and G, assumed to be continuous, are the distribution functions of the two samples respectively. Then the null hypothesis is $H_0 : F = G$.

The rank sum test (also known as the Mann-Whitney test or Wilcoxon test) consists of ordering the $n + m$ values. Since we are assuming that F and G are continuous, this ranking will be unique. Once ordered we will give a rank for every value, which mean that the rank for the fist values is 1, for the second value 2 and so on. For $i = 1, .., n$ we define R_i the rank of the value X_i.

The rank sum test used the test statistic T equal to the sum of the ranks from the first sample

$$T = \sum_{i=1}^{n} R_i$$

We consider a significance level of α for H_0. If we observed $T = t$, then H_0 is

[1] Official documentation can be found at `https://docs.scipy.org/doc/scipy/reference/stats.html`

rejected if

$$P_{H_0}(T \leq t) \leq \frac{\alpha}{2} \text{ or } P_{H_0}(T \geq t) \leq \frac{\alpha}{2}$$

This means that we reject the hypothesis if the sum of ranks from the first hypothesis is either too small or too large to be explained by chance.

This probabilities are equal to

$$P_{H_0}(T \leq t) \leq \frac{\alpha}{2} \text{ or } P_{H_0}(T \leq t-1) \geq 1 - \frac{\alpha}{2} \tag{D.1}$$

To compute these probabilities we use a recursive method. Let $P(N, M, K)$ be the probability that the sum of the ranks of the first sample will be less than or equal to K when samples have sizes N and M and H_0 is true. The probabilities in D.1 are $P(n, m, t)$ and $P(n, m, t-1)$.

Consider the larges of the $N + M$ data values belong to the first sample, then T is the $N + M$ plus the sum of ranks of the other $N - 1$. Then $T \leq K$ if the sum of ranks of the other $N - 1$ is less or equal to $K - (N + M)$. But since the remaining $N - 1 + M$ all come from the same distribution (H_0 is true) it follows that the sum of ranks of the $N - 1$ is less or equal to $K - M - N$ with probability $P(N - 1, M, K - N - M)$. With similar argument, if the larges values belongs to the second distribution we get that the sum of ranks of the first sample will be less than or equal to K with probability $P(N, M - 1, K)$. Also as the larges values is one of the $X_1, ..., X_n, Y_1, ..., Y_m$ it will belong to first sample with probability $N/(N + M)$. Putting this together we get

$$P(N, M, K) = \frac{N}{N+M} P(N-1, M, K-N-M) + \frac{M}{N+M} P(N, M-1, K)$$

Starting from the condition

$$P(1, 0, K) = \begin{cases} 0 & K \leq 0 \\ 1 & K > 0 \end{cases}, \; P(0, 1, K) = \begin{cases} 0 & K < 0 \\ 1 & K \geq 0 \end{cases}$$

The equation can be solved recursively to obtain $P(n, m, t-1)$ and $P(n, m, t)$.

As test rejects the hypothesis when

$$2P(n, m, t \leq \alpha) \text{ or } \alpha \geq 2[1 - P(n, m, t-1)]$$

it follows that the p-value of the test statistic when $T = t$ is

$$2 \min\{P(n, m, t), 1 - P(n, m, t-1)\}$$

Bibliography

[1] Ahmed M. Alaa, Thomas Bolton, Emanuele Di Angelantonio, James H. F. Rudd, and Mihaela van der Schaar, *Cardiovascular disease risk prediction using automated machine learning: A prospective study of 423,604 UK biobank participants*, PLOS ONE **14** (2019), no. 5, e0213653.

[2] C. Allain and M. Cloitre, *Characterizing the lacunarity of random and deterministic fractal sets*, Phys. Rev. A **44** (1991), 3552–3558.

[3] Rahman Attar, Marco Pereañez, Ali Gooya, Xènia Albà, Le Zhang, Milton Hoz de Vila, Aaron M. Lee, Nay Aung, Elena Lukaschuk, Mihir M. Sanghvi, Kenneth Fung, Jose Miguel Paiva, Stefan K. Piechnik, Stefan Neubauer, Steffen E. Petersen, and Alejandro F. Frangi, *Quantitative cmr population imaging on 20,000 subjects of the uk biobank imaging study: Lv/rv quantification pipeline and its evaluation*, Medical Image Analysis **56** (2019), 26–42.

[4] André Ricardo Backes, *A new approach to estimate lacunarity of texture images*, Pattern Recognition Letters **34** (2013), no. 13, 1455–1461.

[5] Christopher M. Bishop, *Pattern recognition and machine learning*, Springer, 2009.

[6] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik, *A training algorithm for optimal margin classifiers*, Proceedings of the Fifth Annual Workshop on Computational Learning Theory (New York, NY, USA), COLT '92, Association for Computing Machinery, 1992, p. 144â152.

[7] Leo Breiman, *Random forests*, Machine Learning **45** (2001), no. 1, 5–32.

[8] Leo Breiman, Jerome H Friedman, Richard A Olshen, and Charles J Stone, *Classification and regression trees*, CRC Press, 1984.

[9] Gabriella Captur, Audrey L. Karperien, Alun D. Hughes, Darrel P. Francis, and James C. Moon, *The fractal heart — embracing mathematics in the cardiology clinic*, Nature Reviews Cardiology **14** (2016), no. 1, 56–64.

[10] Gabriella Captur, Vivek Muthurangu, Christopher Cook, Andrew S Flett, Robert Wilson, Andrea Barison, Daniel M Sado, Sarah Anderson, William J McKenna, Timothy J Mohun, Perry M Elliott, and James C Moon, *Quantification of left ventricular trabeculae using fractal analysis*, J Cardiovasc Magn Reson **15** (2013), no. 1, 36 (en).

[11] Gabriella Captur, Filip Zemrak, Vivek Muthurangu, Steffen E Petersen, Chunming Li, Paul Bassett, Nadine Kawel-Boehm, William J McKenna, Perry M Elliott, João A C Lima, David A Bluemke, and James C Moon, *Fractal analysis of myocardial trabeculations in 2547 study participants: Multi-Ethnic study of atherosclerosis*, Radiology **277** (2015), no. 3, 707–715 (en).

[12] Tianqi Chen and Carlos Guestrin, *Xgboost: A scalable tree boosting system*, CoRR **abs/1603.02754** (2016).

[13] Corinna Cortes and Vladimir Vapnik, *Support-vector networks*, Machine Learning **20** (1995), no. 3, 273–297.

[14] Adele Cutler, David Cutler, and John Stevens, *Random forests*, vol. 45, pp. 157–176, 2011.

[15] P. Dong, *Test of a new lacunarity estimation method for image texture analysis*, International Journal of Remote Sensing **21** (2000), no. 17, 3369–3373.

[16] Kenneth Falconer, *Fractal geometry : mathematical foundations and applications*, 3rd ed., John Wiley & Sons, 2014.

[17] Barry Alan Fittes, *Gray-level transformations for interactive image enhancement*, Tech. report, 1975.

[18] Polyxeni Gkontra, Kerri-Ann Norton, Magdalena M. Żak, Cristina Clemente, Jaume Agüero, Borja Ibáñez, Andrés Santos, Aleksander S. Popel, and Alicia G. Arroyo, *Deciphering microvascular changes after myocardial infarction through 3d fully automated image analysis*, Scientific Reports **8** (2018), no. 1.

[19] Polyxeni Gkontra, Magdalena Żak, Kerri-Ann Norton, Andrés Santos, Aleksander Popel, and Alicia Arroyo, *A 3d fractal-based approach towards understanding changes in the infarcted heart microvasculature*, 10 2015, pp. 173–180.

[20] Daniel J. Gould, Tegy J. Vadakkan, Ross A. Poché, and Mary E. Dickinson, *Multifractal and lacunarity analysis of microvascular morphology and remodeling*, Microcirculation **18** (2011), no. 2, 136–151.

[21] J. E. Hutchinson, *Fractals and self similarity*, Indiana University Mathematics Journal **30** (1981), no. 5, 713–747.

[22] Karim Lekadir, Richard Osuala, Catherine Gallin, Noussair Lazrak, Kaisar Kushibar, Gianna Tsakou, Susanna Aussó, Leonor Cerdá Alberich, Kostas Marias, Manolis Tsiknakis, Sara Colantonio, Nickolas Papanikolaou, Zohaib Salahuddin, Henry C Woodruff, Philippe Lambin, and Luis Martí-Bonmatí, *Future-ai: Guiding principles and consensus recommendations for trustworthy artificial intelligence in medical imaging*, 2021.

[23] Jian Li, Qian Du, and Caixin Sun, *An improved box-counting method for image fractal dimension estimation*, Pattern Recognition **42** (2009), no. 11, 2460–2469.

[24] Yu Liu, Lingyu Chen, Heming Wang, Lanlan Jiang, Yi Zhang, Jiafei Zhao, Dayong Wang, Yuechao Zhao, and Yongchen Song, *An improved differential box-counting method to estimate fractal dimensions of gray-level images*, Journal of Visual Communication and Image Representation **25** (2014), no. 5, 1102–1111.

[25] Sylvie Lorthois and Francis Cassot, *Fractal analysis of vascular networks: insights from morphogenesis*, J Theor Biol **262** (2009), no. 4, 614–633 (en).

[26] Scott M Lundberg and Su-In Lee, *A unified approach to interpreting model predictions*, Advances in Neural Information Processing Systems 30 (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), Curran Associates, Inc., 2017, pp. 4765–4774.

[27] B. B. Mandelbrot, *The fractal geometry of nature*, 3 ed., W. H. Freeman and Comp., New York, 1983.

[28] Benoit B. Mandelbrot, *Measures of fractal lacunarity: Minkowski content and alternatives*, Fractal Geometry and Stochastics (Basel) (Christoph Bandt, Siegfried Graf, and Martina Zähle, eds.), Birkhäuser Basel, 1995, pp. 15–42.

[29] Pertti Mattila, *Geometry of sets and measures in euclidean spaces : fractals and rectifiability*, Cambridge studies in advanced mathematics ; 44, Cambridge University Press, Cambridge, 1995 (eng).

[30] Alexandra N. Nowbar, Mauro Gitto, James P. Howard, Darrel P. Francis, and Rasha Al-Lamee, *Mortality from ischemic heart disease*, Circulation: Cardiovascular Quality and Outcomes **12** (2019), no. 6.

[31] Steffen E Petersen, Nay Aung, Mihir M Sanghvi, Filip Zemrak, Kenneth Fung, Jose Miguel Paiva, Jane M Francis, Mohammed Y Khanji, Elena Lukaschuk, Aaron M Lee, et al., *Reference ranges for cardiac structure and function using cardiovascular magnetic resonance (cmr) in caucasians from the uk biobank population cohort*, Journal of Cardiovascular Magnetic Resonance **19** (2017), no. 1, 1–19.

[32] Gardner R.H. O'Neill R.V Plotnick R.E., *Lacunarity indices as measures of landscape texture*, Landscape Ecol **8** (1993), 201–211.

[33] Esmeralda Ruiz Pujadas, Zahra Raisi-Estabragh, Liliana Szabo, Celeste McCracken, Cristian Izquierdo Morcillo, Víctor M. Campello, Carlos Martín-Isla, Angelica M. Atehortua, Hajnalka Vago, Bela Merkely, Pal Maurovich-Horvat, Nicholas C. Harvey, Stefan Neubauer, Steffen E. Petersen, and Karim Lekadir, *Prediction of incident cardiovascular events using machine learning and CMR radiomics*, European Radiology **33** (2022), no. 5, 3488–3500.

[34] C.A. Rabbath and D. Corriveau, *A comparison of piecewise cubic hermite interpolating polynomials, cubic splines and piecewise linear functions for the approximation of projectile aerodynamics*, Defence Technology **15** (2019), no. 5, 741–757.

[35] Zahra Raisi-Estabragh, Nicholas C Harvey, Stefan Neubauer, and Steffen E Petersen, *Cardiovascular magnetic resonance imaging in the UK biobank: a major international health research resource*, Eur Heart J Cardiovasc Imaging **22** (2021), no. 3, 251–258 (en).

[36] Elisa Rauseo, Cristian Izquierdo Morcillo, Zahra Raisi-Estabragh, Polyxeni Gkontra, Nay Aung, Karim Lekadir, and Steffen E. Petersen, *New imaging signatures of cardiac alterations in ischaemic heart disease and cerebrovascular disease using cmr radiomics*, Frontiers in Cardiovascular Medicine **8** (2021).

[37] Sheldon M Ross, *Introduction to probability and statistics for engineers and scientists*, 5 ed., Academic Press, San Diego, CA, August 2014.

[38] Nirupam Sarkar and B.B. Chaudhuri, *An efficient approach to estimate fractal dimension of textural images*, Pattern Recognition **25** (1992), no. 9, 1035–1041.

[39] L. S. Shapley, *Contributions to the theory of games (am-28), volume ii*, pp. 307–318, Princeton University Press, Princeton, 1953.

[40] Arlene Sirajuddin, S. Mojdeh Mirmomen, Seth J. Kligerman, Daniel W. Groves, Allen P. Burke, Faraz Kureshi, Charles S. White, and Andrew E. Arai, *Ischemic heart disease: Noninvasive imaging techniques and findings*, RadioGraphics (2021).

[41] Elias M. Stein and Rami Shakarchi, *Real analysis: Measure theory, integration, and hilbert spaces*, pp. 323–388, Princeton University Press, 2005.

[42] Jack Stewart, Gavin Manmathan, and Peter Wilkinson, *Primary prevention of cardiovascular disease: A review of contemporary guidance and literature*, JRSM Cardiovascular Disease **6** (2017).

[43] Carolin Strobl, Anne-Laure Boulesteix, Achim Zeileis, and Torsten Hothorn, *Bias in random forest variable importance measures: Illustrations, sources and a solution*, **8** (2007), no. 1, 25.

[44] W. Sun, G. Xu, P. Gong, and S. Liang, *Fractal analysis of remotely sensed images: A review of methods and applications*, International Journal of Remote Sensing **27** (2006), no. 22, 4963–4990.

[45] Charles R. Tolle, Timothy R. McJunkin, and David J. Gorsich, *An efficient implementation of the gliding box lacunarity algorithm*, Physica D: Nonlinear Phenomena **237** (2008), no. 3, 306–315.

[46] Yu-Chang Tzeng, Kuo-Tai Fan, and Kun-Shan Chen, *A parallel differential box-counting algorithm applied to hyperspectral image classification*, ieee geoscience and remote sensing letters **9** (2011), no. 2, 272–276.

[47] Joost J.M. van Griethuysen, Andriy Fedorov, Chintan Parmar, Ahmed Hosny, Nicole Aucoin, Vivek Narayan, Regina G.H. Beets-Tan, Jean-Christophe Fillion-Robin, Steve Pieper, and Hugo J.W.L. Aerts, *Computational radiomics system to decode the radiographic phenotype*, Cancer Research **77** (2017), no. 21, 104–107.

[48] Yuxuan Xia, Jianchao Cai, Edmund Perfect, Wei Wei, Qi Zhang, and Qingbang Meng, *Fractal dimension, lacunarity and succolarity analyses on ct images of reservoir rocks for permeability prediction*, Journal of Hydrology **579** (2019), 124–198.