Random Regression Forests for Fully Automatic Multi-Organ Localization in CT Images

Prasad Samarakoon

▶ **To cite this version:**

Prasad Samarakoon. Random Regression Forests for Fully Automatic Multi-Organ Localization in CT Images. General Mathematics [math.GM]. Université Grenoble Alpes, 2016. English. NNT : 2016GREAM039 . tel-01449813v2

HAL Id: tel-01449813

https://theses.hal.science/tel-01449813v2

Submitted on 10 Jan 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Communauté
UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Mathématiques et Informatique**

Arrêté ministérial : du 7 août 2006

Présentée par

Prasad N. SAMARAKOON

Thèse dirigée par **Emmanuel PROMAYON**
et codirigée par **Céline FOUARD**

préparée au sein **laboratoire Techniques de l'Ingénierie Médicale et de la Complexité - Informatique, Mathématiques et Applications de Grenoble (TIMC-IMAG)**
et de **l'école doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique (MSTII)**

Random Regression Forests for Fully Automatic Multi–Organ Localization in CT Images

Thèse soutenue publiquement le **30 septembre 2016**,
devant le jury composé de :

M. Grégoire MALANDAIN
DR, INRIA Sophia Antipolis, Sophia Antipolis, Président
M. Ivan BRICAULT
PU-PH, Centre Hospitalier Universitaire Grenoble Alpes, Grenoble, Examinateur
Mme. Isabelle BLOCH
PR, Télécom ParisTech, CNRS LTCI, Paris, Rapporteur
M. Jean-Louis DILLENSEGER
MCF, Institut Universitaire de Technologie de Rennes, INSERM U1099, Rennes, Rapporteur
M. Emmanuel PROMAYON
MCF, Université Grenoble Alpes, TIMC-IMAG-CNRS, Grenoble, Directeur de thèse
Mme. Céline FOUARD
MCF, Université Grenoble Alpes, TIMC-IMAG-CNRS, Grenoble, Co-Directeur de thèse
M. Nabil ZEMITI
MCF, Université Montpellier 2, LIRMM UMR 5506, Montpellier, Invité

To my mother, the lady with the most kind and gentle soul who has shown me how to be true to yourself amidst constant hardships.

To my father, who has spoken to me through his actions rather than words that it's never too late to try or to change.

To my better half, whose unconditional love, support, understanding and motivation have shaped me into the person who I am today and making each day of my life, a bliss to live in.

Abstract

Computer assisted medical intervention has become an integral part of present day's medicine where medical image analysis plays an indispensable role. With the advancements of the modern day computing resources, machine learning techniques have emerged as a vital component in this field. The use of the supervised machine learning technique called random forests has shown very encouraging results in medical image analysis. More specifically, Random Regression Forests (RRFs), a specialization of random forests for regression, have produced the state of the art results for fully automatic multi–organ localization. Despite the very encouraging results, the relative novelty of the method in this field still raises numerous questions about how to optimize its parameters for consistent and efficient usage. Additionally, the RRF method has many parameters that require heuristic tuning which reduces its ability to be used in a more general setting.

In this context, the goal of this dissertation is to carry out a detailed study on the use of the Random Regression Forest methodology for multi–organ localization. First, we perform a thorough analysis of decision trees and of RRFs in the context of multi–organ localization in order to present and understand the inner workings of RRFs. From this, three directions are explored. The first direction investigates whether the localization performance of RRFs can be further improved by adding more spatially consistent information. We then propose to use the random model variables to approximate the random process. This results in a newer type of RRF, faster and more efficient in terms of memory usage: the Light Random Regression Forest. Finally, we propose an automatic and consistent approach to find the forest leaf nodes that participate in the final localization prediction. Furthermore, this proposal leads to the elimination of two other arbitrarily tuned parameters increasing the generality of RRFs for multi–organ localization, without reducing their localization performances.

Résumé

Les Gestes Médico-Chirurgicaux Assistés par Ordinateur sont devenus une partie intégrante de la médecine d'aujourd'hui où l'analyse d'images médicales joue un rôle indispensable. Avec les progrès des ressources informatiques, les techniques de l'apprentissage automatique ont émergé comme un élément essentiel dans ce domaine. L'utilisation de la technique de l'apprentissage supervisé appelée "forêts aléatoires" a montré des résultats très encourageants dans l'analyse de l'imagerie médicale. Plus précisément, les "Random Regression Forests" (RRFs), une spécialisation des forêts aléatoires pour la régression, ont produit d'excellents résultats pour la localisation automatique multi-organes. Malgré ces résultats impressionnants, la relative nouveauté de cette méthode soulève encore de nombreuses questions d'optimisation pour une utilisation cohérente et efficace. En outre, les RRFs ont de nombreux paramètres qui nécessitent une optimisation heuristique, réduisant ainsi leur capacité à être utilisés dans un contexte plus général.

L'objectif de cette thèse est de réaliser une étude détaillée sur l'utilisation de la méthodologie des RRFs dans la localisation automatique de plusieurs organes. Tout d'abord, nous procédons à une analyse approfondie des arbres de décision et des RRFs dans le contexte de localisation de plusieurs organes afin de présenter et de comprendre leurs fonctionnements internes. De là, trois axes sont explorés. Le premier axe examine si les performances de localisation des RRFs peut encore être améliorées en ajoutant de l'information spatiale plus cohérente. Nous proposons ensuite d'utiliser les variables du modèle aléatoires pour approcher le processus aléatoire. Cela se traduit par un nouveau type de RRFs, plus rapide et plus efficace en termes d'utilisation de la mémoire : les Light Random Regression Forests. Enfin, nous proposons une approche automatique et cohérente pour trouver les nœuds feuilles qui participent à la prédiction finale de la localisation. En outre, cette proposition conduit à l'élimination de deux autres paramètres arbitrairement ajustés augmentant la généralité de RRF sans réduire leurs performances de localisation.

Acknowledgment

This work would not have been a reality had it not being for Céline. I believe, the mutual understanding that grew between Céline and me during the time she was my lecturer of Signals and Image Processing module in 2011 and then as my Masters 2 research supervisor in 2012 paved way for her to entrust me with the responsibility of this research work. During the 42 months that we have been working together, not even once she had discouraged me or made me question my decision of wanting to do research. On the contrary, she has always encouraged me to follow my intuition without forgetting to put me under "reality-checks" every now and then. Among the many inspirational stories you have shared with me in rough moments, I would never forget the "pitbull" story. I make this an opportunity to pay my highest regards to you Céline for supporting and encouraging me throughout the journey. It would be utterly incomplete in my mind if I missed to appreciate the gigantic effort you put into preparing the lecture notes and the course work of the module of Signals and Image Processing (and without a doubt all the other modules too). Thank you very much for the best 10 sessions of Signals and Image Processing lectures that I have ever experienced.

It has been nothing less than a delight to work under Mahnu. I consider myself really lucky to have had the chance to work closely with such a great human who is also a walking encyclopedia too. I have never seized to admire his vivid imagination that leads to fascinating hypotheses while remaining in the realm of reality. It is amazing how Mahnu finds time not only for me but for all his students amidst his extremely busy schedule. His guidance and constant encouragement have always given me that extra energy I needed to course-correct myself in times of self doubt and uncertainty. I am ever so thankful to the wonderful mentor, the role model you have been for me during the last three years.

I am very grateful to Prof. Isabelle Bloch and Dr. Jean–Louis Dillenseger for accepting to review this dissertation amidst their busy schedules and

during the summer. Their constructive feedback has helped us immensely to have an augmented view of the entire project. I am also thankful to Prof. Ivan Bricault, Dr. Grégoire Malandain, and Dr. Nabil Zemiti for agreeing to be the examiners at the thesis defense.

I would make this an opportunity to show my gratitude to two other mentors from the university of Moratuwa, Mrs. Nanayakkara and Dr. Gamage. The unbelievable investment of Mrs. Nanayakkara on the well-being and soft-skill development of her students never seized to amaze me. Being the best possible complement of her, the passion and drive that Dr. Gamage brings to the research-skill development in an institution that doesn't possess a lot of resources is exemplary. I am also very grateful to Ylies Falcon for rekindling my desire of research and being one of the most dedicated lectures that I have had the good fortune of meeting.

I have had the good fortune of getting into a very dynamic and welcoming research team at the laboratory. I have had nothing but pleasant memories with each member of GMCAO. Among everyone a special note of thank you goes out to Pierre-Loup, Ben, Nicolas, Arnaud, Paul, Johan, Mathieu, Anthony, Sonia, Baptist, Giao, Antoine, Anna, Fanny, Guillaume, Raef, Jérôme and Pierre-Yves for being my lunch buddies and coinche partners. Though our weather tolerance capacities were at the opposite extremes, Cecilia and I shared the office for two years without any conflicts; weather related or otherwise. She has been a good friend who always motivated me giving moral support, food treats in addition to the numerous philosophical discussions. Thank you for making our office a fun and interesting place to be. I have had the good fortune of sharing an office with Jérôme, Anna and Ahmad too. They all have been wonderful to me and I have enjoyed our times together, both serious and fun. I want to thank Ivonne for all the tasty and spicy Mexican food and all the philosophical conversations we have had. Starting our PhDs at the same time, Ahmad and I discovered the different phases of the PhD together. Those discoveries and numerous discussions we have had paved the way to our friendship. Ahmad, thank you for all the kind and thoughtful gestures of yours.

Vincent Luboz was my supervisor of the research project carried out during the first year of the Masters. His humility does not seem to have limits. I have always looked up to him during the years that I have got to know him. His guidance, constant support and simply his way of being have

had a great impact on my life. My mentor, my friend, thank you very much for all what you have done.

We had our doubts moving into la Tour du Pin as that was the first time we moved away from a big city in France. But our ex-neighbor, Laurent made our doubts vanish in a matter of days. His generous ways and appetite for surprises not only made us become bolder but also ignited the spark of adventurer side of our souls. Thank you for being our friend and for all the nice memories.

Bastien and Séverine have been our best friends in France. They made us discover many French treasures that were hidden to us before. They have always come to our rescue in times of need. Thank you for your thoughtfulness, helping hands, numerous spend the days and many fun trips together.

I would like to thank the Agence Nationale de la Recherche (ANR) for providing the funding through the "Technologies pour la santé et l'autonomie" (TecSan) project Robacus (ANR-11-TECS-020-01).

It is not an easy task to find words to thank the ones who have made me who I am today. Yet, the least I could do, is to try to show my gratitude to them although trying that is as unattainable as measuring how big the universe is with a measuring tape.

They are the ones who are responsible for my motto of life: "Je suis qui je suis!".

Since a long time, my elder brother carried a lot of weight of the family by helping my parents in numerous ways. I would want to make this an opportunity to pay my utmost respect and acknowledgment for all what you have done for me. My sister has been my solace with her unique way of showing love and care. Thank you for all the hand holding and kind gestures which added a lot of meaning to my life.

Growing up in an era where competition is not only promoted but also thought and taught to be essential, I am forever in debt to my parents who constantly kept me away from competition with others but myself. As a kid, I asked for a lot of materialistic things from my parents. Though they were not rich by any means of the word, I never recall a single instant where I wasn't given what I had asked for. Leading my own life with my better half and having the first hand experience of living on a tight budget, I can do nothing but wonder how they were able to provide me all what I asked for. As an adult, there were instances that they stood by my decisions

although what they believed were the total opposite. It does not mean that there were neither arguments nor explanations; there were plenty! But the selfless manner in which they conducted themselves were only known to me in stories but not in real life. Ma and Tha, I am forever in debt to you. And I am a tad sad that one day as a parent, I will never be able to raise the bar that you always kept.

I have been blessed with the rare opportunity to live my life with my best friend. And she has been my best friend much longer than she has been my wife. Life has been fun, adventurous and rewarding since the first day that we have met. How she gets up every time she falls down and finds more courage than what she started with to get to her humble targets have mesmerized not only me but everyone around her. I never thought that traversing a country by bike would be a feast that we would taste. But with the vision of my lion hearted little warrior, we did make it happen. In the same manner, among many other things, there would not have been a thesis to write, had it not been for her. She has literally been the guiding force behind me. With her unconditional love, undying faith in me, unfailing encouragements and exemplary life traits have carried me through yet another journey of my life. Nuts, my love, I am ever so thankful to you for being with me and being there for me through thick and thin. BHW!

Contents

List of Figures

List of Tables

List of Abbreviations

AID Automatic Interaction Detector

BWPE Bounding Wall Prediction Error

CAMI Computer Assisted Medical Intervention

CART Classification And Regression Tree

CE Centroid Error

CHM Centroid–Hit Measure

CT Computed Tomography

HbRRF Hough–based Random Regression Forest

HU Hounsfield Unit

ID3 Iterative Dichotomiser 3

IQR Inter Quartile Range

LPR Light Puncture Robot

LRRF Light Random Regression Forest

MAP Maximum *A–Posteriori*

MR Magnetic Resonance

MRI Magnetic Resonance Imaging

NDT Non-linear Decision Tree

PFDT Polynomial-Fuzzy Decision Tree

RAM Random Access Memory

RCF Random Classification Forest

RF Random Forest

ROI Region of Interest

RRF Random Regression Forest

RRT Random Regression Tree

RSMDT Rough Set-based Multivariate Decision Tree

Every new beginning comes from some other beginning's end.

- Seneca

1 Introduction

1.1 Context

Medicine[1] is defined as the science or practice of the diagnosis, treatment, and prevention of disease. The pioneering work carried out by Ledley and Lusted [1959] in late 1950's on reasoning foundations of medical diagnosis laid the stepping stones in harnessing the power of computers in medicine. Soon after, the theoretical advances of Ledley and Lusted were put to good use not only for diagnosis but also for treatment by many researchers [Warner et al., 1964; Gorry and Barnett, 1968; Weiss et al., 1978].

Surgery[2], the treatment of injuries or disorders of the body by incision or manipulation, especially with instruments is often a combination of complex procedures. With the rapid advancements in the semiconductor industry the computers shrank in size but grew drastically in capabilities and performance enabling the computers to be used in surgery. Computer assisted medical intervention was the inevitability [Lavallee and Cinquin, 1990; Cinquin et al., 1995].

[1] "Medicine, n.1." OED Online. Oxford University Press. Retrieved March 17, 2016.
[2] "Surgery, n.1." OED Online. Oxford University Press. Retrieved March 17, 2016.

Computer Assisted Medical Intervention (CAMI) is defined as the discipline that aims at providing tools allowing the clinician to use multi-modal data in a rational and quantitative manner in order to plan, simulate, and execute minimum invasive medical interventions accurately and safely [Troccaz, 2009]. Presently, CAMI plays a major role in all main phases of medicine, namely, prevention, diagnosis, treatment, and treatment follow-up.

The general methodology of CAMI can be summarized as a three-fold loop of perception, decision/simulation, and action [Cinquin et al., 1995] as illustrated in Fig. 1.1. The perception phase consists of multi-modal data acquisition and processing through standard medical imaging techniques such as Computed Tomography (CT), Magnetic Resonance Imaging (MRI), ultra sound, photon emission tomography, endoscopy, *etc.* The perception phase is only complete when the data acquired by various sensors are taken into consideration as well. The decision phase is two fold. First, a digital patient is created by correctly modeling the acquired patient data. Then, a plan of action is derived based upon the information obtained through the digital patient while maximizing the efficiency and minimizing the risk of the therapeutic procedure. In the action phase, the planned intervention is carried out by the clinician, for instance, with the assistance of a navigational medical robot.

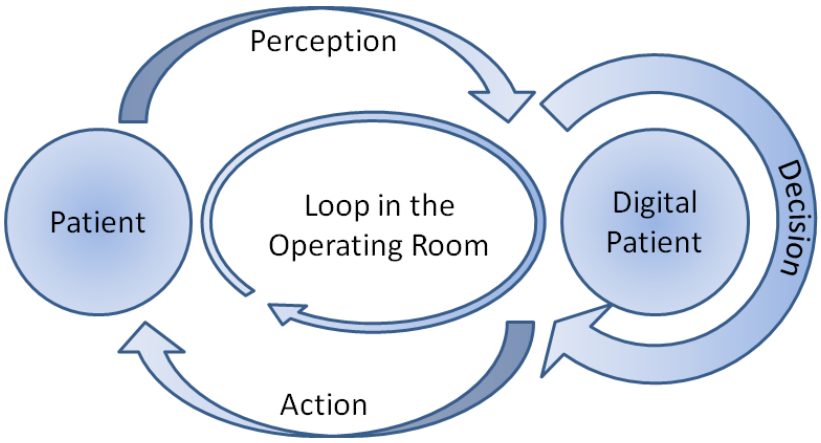


Figure 1.1: General methodology of CAMI consists of a three–fold loop. Perception: data acquisition through medical imaging techniques coupled with other sensory input. Decision: creation of a digital patient and devising a plan of action. Action: carrying out the planned intervention with the probable use of a navigational medical robot.

As an illustration, in this setting, the laboratory of TIMC-IMAG (Techniques for biomedical engineering and complexity management – informatics, mathematics and applications – Grenoble) has developed a navigational robot called the Light Puncture Robot (LPR)[3] [Taillant et al., 2004; Bricault et al., 2008; Zemiti et al., 2008]. LPR is an active (*i.e.*, the robot carries out certain tasks autonomously with the supervision of the clinician) percutaneous[4] robot which is both CT and MRI compatible. It is used to enable complex needle trajectories that require high accuracy and would otherwise be very challenging to do manually.

An indispensable step of the decision phase in building the digital patient is the extraction of high level information of the patient data through means of segmentation. Segmentation is the process of partitioning the image into some non-intersecting regions in such a way that each region is homogeneous with respect to the intended application [Pal and Pal, 1993]. In CAMI, segmentation is the process of partitioning the image pixels/voxels[5] into biologically meaningful non-overlapping regions. The stage of segmentation is of utmost importance as the decisions and actions taken are dependent upon the segmented output.

When organ segmentation is the goal of medical image segmentation, the aim is to partition the image voxels into non-overlapping anatomical structures. A tremendous number of research work has been carried out on segmentation of medical images. The various proposed segmentation methods can be broadly summarized into four major categories although different researchers have proposed different categories [Sharma et al., 2010; Yan and Wang, 2010; Gu and Peters, 2006]. They are:

1. methods based on gray level features,
2. methods based on geometric models,
3. methods based on statistical atlases, and
4. knowledge based methods.

One method from one category is seldom sufficient to carry out any practical organ segmentation. A collection of methods belonging to different categories are often required depending on the intended segmentation purpose.

[3]This work was funded by the French ANR within the TecSan project Robacus (ANR-11-TECS-020-01)

[4]Needle insertion through the skin

[5]Here after the term 'voxel' will be employed in place of the term 'pixel/voxel'.

Organ localization is the antecedent step of organ segmentation. Locating an organ in a medical image by bounding that particular organ with respect to an entity such as a bounding box or sphere is termed organ localization. Multi-organ localization takes place when multiple organs are localized simultaneously.

The Random Forest (RF) [Breiman, 2001] method is a famous knowledge based method that is utilized in many fields. A known variant of random forests, called Random Regression Forests (RRFs) was first used to solve the multi-organ localization problem in CT images by Criminisi et al. in 2010 [Criminisi et al., 2010]. The studies carried out by Criminisi et al. showed a lot of promise by outperforming the state of the art results reported at the time [Criminisi et al., 2010, 2013].

1.2 Motivation

An important step of the usage of LPR in an actual percutaneous operation is segmenting the target organ and the other organs to avoid during the procedure. Currently, the organ segmentation step is carried out manually by the clinician. Ultimately, our goal is to make LPR as autonomous as possible in order to provide the clinician with seamless assistance in performing percutaneous procedures. One axis towards the above mentioned goal is the fully automatic segmentation of the relevant organs. **As fully automatic multi–organ segmentation is a very challenging scientific task, as an initial step, we first concentrated on fully automatic multi–organ localization.**

We opted to use the RRF method for automatic multi–organ localization as the state of the art results were produced using the same method. Despite the very encouraging results, the RRF method has many parameters that require heuristic tuning that inhibit its ability to be used in a more general setting. According to Louppe [2014], *"... the theoretical properties and statistical mechanisms that drive the algorithm are still not clearly and entirely understood. Random forests indeed evolved from empirical successes rather than from a sound theory ... "*. Louppe also states that although the construction of a basic building block of RFs, *i.e.*, a single decision tree, can be easily described, proper and efficient implementation of the algorithm remains a challenging task. Consequently, scientific literature often omits

the implementation details.

The same inhibitions identified by Louppe hold true for the more specialized form of RFs, *i.e.*, the Random Regression Forests. Despite its superior performance over the state of the art, we assume that the slow rate of adaptability of the RRF algorithm for multi-organ localization among the medical image processing community should be due to these inhibitions.

1.3 Objective and Scope

In this backdrop, **the objective of this dissertation is to provide a detailed study of using Random Regression Forest methodology for multi-organ localization, emphasizing on the understanding of the influence of various parameters and making the RRF method as generic as possible by eliminating the heuristic tuning of certain parameters.**

The dissertation evolves exclusively around the problem of multi-organ localization. Among the many medical imaging modalities present, such as magnetic resonance, ultra-sound, x-ray, *etc.*, we limit our study to computed tomography as it is the first image modality used in the final LPR application.

1.4 Significance

The main intended outcome of the study, on a theoretical level, is to advance knowledge in the use of Random Regression Forest method for multi-organ localization in CT images that promotes more generic usage across different organ localization applications thanks to the reduction of ad hoc parameters and the development of a rigorous methodology.

On a more practically applicable level, another intention of the study, is to make certain implementation details available that would either enhance localization or result in more generalization or lead to more computational and resource efficient solutions.

Though our main focus is on random regression forests, the theoretical and practical advancements discussed during the study is generic enough so that we hope that they will be applied in other forms of random forests too.

1.5 Overview

The remainder of this dissertation contains 7 chapters.

In Chap. 2, we present decision trees, the fundamental building block of RRF. It includes a concise introduction to machine learning, a short bibliographical review and a detailed analysis of the key concepts of decision trees. Then, the Random Regression Forests are presented in Chap. 3. This chapter constitutes a thorough study of the related literature, an in–depth analysis of the key concepts, and a digest of the multi–organ localization process using RRFs. We tried to present both Chap. 2 and Chap. 3 in a didactic manner providing simple and easy to understand examples where possible.

The scientific approach adopted to carry out the various studies of the dissertation are described in Chap. 4 along with the information on our CT image database and the benchmarking technique. In Chap. 5, we study the effect on the RRF algorithm when some spatial consistent information are incorporated. We present Light Random Regression Forests in Chap. 6, a new type of RRF that is faster, uses much less memory than classic RRFs while maintaining the same localization capabilities. We take the first steps towards a more generalized RRF framework in Chap. 7 by proposing a consistent and automatic method to choose the number of forest leaf nodes that participate in the final organ localization prediction.

Finally, Chap. 8 presents a discussion on organ detection using RRFs followed by general concluding remarks and future perspectives.

A good decision is based on knowledge and not on numbers.

- Plato

2

Decision Trees and Their Analysis

Contents

In this chapter we describe decision trees, the fundamental building block of random forests. The introductory section (Sect. 2.1) comprises a concise description pertaining to machine learning along with a handful of core definitions. Then, the decision trees are presented in a more detailed manner in Sect. 2.2. The main components of a decision tree are introduced in Sect. 2.2.1 before concisely presenting its evolution in Sect. 2.2.2. Details on the node split function and related parameters are introduced in Sect. 2.2.3 whereas Sect. 2.2.4 introduces how to evaluate

the *peformance* of a split. In Sect. 2.2.5, we present how to make a prediction using the leaf nodes.

Our analysis of various aspects of the decision tree concept is presented in Sect. 2.3. First, the main components of supervised learning are analyzed in Sect. 2.3.1 before the main steps of decision tree induction are analyzed in Sect. 2.3.2. Finally, Sect. 2.3.3 presents how the best split can be chosen among many splits.

Throughout the chapter, whenever possible, we have used a simple example to illustrate the concept being described. The same example is developed as the chapter progresses in order to enhance the didactic nature of the dissertation.

2.1 Introduction to Machine Learning

Machine learning can generally be described as the science concerned with creating computer systems and algorithms that enables machines to "learn" from previous experience [Izenman, 2008].

Definition: Machine Learning

In Murphy [2012], machine learning is defined as "a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision making under uncertainty".

According to Murphy, machine learning tasks can be broadly categorized into three main categories.

1. *Predictive or supervised learning*: given a labeled set of input-output pairs (called the training set), learning a mapping from inputs to outputs in such a manner that when a previously unseen input is given, the learned mapping is capable of "predicting" the correct output. Depending on the characteristics of the output, the supervised learning is described differently. When the output is categorical or nominal, it is termed as classification. When the output is real-valued, it is termed as regression.

   An email spam filter is a real-life classification example. The goal of the filter is to classify the incoming email as spam or not depending on

its training set. Predicting how long a patient would be hospitalized using the previous data collected such as type of disease, age, gender, income, *etc.*, is an example of regression.

2. *Descriptive or unsupervised learning*: given a set of inputs, the ultimate goal is to find "interesting patterns" or to describe the input. Learning is unsupervised as no training set containing the input-output relationship is provided similar to the supervised learning. Thus the problem is not well defined compared to supervised learning.

   Finding clusters in a group when no prior information is available about how many clusters are actually present, is an example of unsupervised learning. By analyzing the purchasing and web-surfing behavior, clustering buyers into groups in order to carry out targeted promotional campaigns is a common practice in present day e-commerce.

3. *Reinforcement learning*: is situated between supervised and unsupervised learning. Learning by interacting with the environment and measuring the effects of these interactions through reward or punishment is the core of reinforcement learning. This reflects one of the fundamental ways that humans learn.

   Reinforcement learning is extensively employed in game playing. As each move of a game depends on many factors, covering all the possibilities exclusively is impossible even for a simple board game. Hence, the best way to learn is by playing, i.e., by applying reinforcement learning.

The following section introduces the common terminology used in the field of machine learning in order to move onto more in-depth discussions pertaining to decision trees.

2.1.1 A Spoonful of Machine Learning Jargon

Similar to all fields, the field of machine learning has its own terminology. It is important to have a clear understanding of this jargon in order to submerge in the world of machine learning. An effort is made to familiarize a few of the main terms and the mathematical notation used, through one of the introductory classification examples retrieved from the pioneering work of Breiman et al. in 1984.

The study consisted of 215 heart attack patients who survived at least 24 hours after being hospitalized due to a heart attack. The goal of the study was to identify the patients who would survive more than 30 days (termed "low risk") and the patients who would not (termed "high risk"). More technically, the goal was to come up with a classification algorithm using the data provided by the 215 patients. 19 variables, including age, blood pressure, previous heart pathology, *etc.*, were used for the study.

Definition: Features

Any property or measurement of an entity or any calculation that can be performed on it are called a feature of that entity.

Hence, each patient of the study can be described using 19 features. Features are two fold.

1. Numerical: the quantitative variables (features) whose values are integers of real numbers. (*e.g.*, age, blood pressure, heart rate, *etc.*)
2. Categorical: the qualitative variables (features) whose values are symbolic. (*e.g.*, previous myocardial infraction, presence of an elevated rate of impulse, *etc.*)

Mathematically, each patient is described as a vector having 19 dimensions (19D).

Note: Mathematical Notation

Throughout the dissertation, boldface lowercase symbols (e.g., $\mathbf{x}$), typewriter uppercase symbols (e.g., $\mathtt{M}$), and calligraphic uppercase symbols (e.g., $\mathcal{S}$) are employed to denote vectors, matrices, and sets respectively. A conscious effort is made to adhere to the above mentioned mathematical notation in order to maintain the consistency throughout the text.

Then, the i^{th} patient (also termed the i^{th} *instance* or *sample* or *data point*) is described as:

$$\mathbf{x}_i = (x_{i_1}, x_{i_2}, \ldots, x_{i_{19}}) \quad .$$

The set of input instances makes up the input $\mathcal{X}$, where

$$\mathcal{X} = \{(\mathbf{x}_i)\}_{i=1}^{215} \quad .$$

Similarly, the output $\mathcal{Y}$ is comprised of one/multi dimensional numerical/ categorical variables. In this particular example, the output consists of one dimensional (1D) categorical variables:

$$\mathcal{Y} = \{(y_i)\}_{i=1}^{215} \ ,$$

where $y_i = \{\text{low risk or high risk}\}$. The 215 patients in this study consisted of the training set.

Definition: Training Set

A labeled set of input–output pairs $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$ is defined as the training set where N is the number of training instances.

Although all available patient data were used as the training set in this study, later studies divided them into two groups called the training set and testing set.

Definition: Testing Set

Another labeled set of input–output pairs that is used to fine tune the parameters and assess the strength and utility of the predictive relationship found by the training process. The testing set does not include the data pairs used in the training set.

While the training set is used to build the learning algorithm, the testing set is used to fine tune the parameters and estimate the performance of the algorithm. When the amount of training data available is limited, often, cross validation is performed.

Definition: Cross Validation

First, the available training data is split into certain number of groups (say K). Then the algorithm will be trained K times, using all groups but the k^{th} group where $k \in \{1, 2, \ldots, K\}$. Each time, the left out portion of the data is used as the testing set. This is defined as the K *cross validation* strategy.

To answer the initial problem, Breiman et al. [1984] proposed the following tree–like structure (see Fig. 2.1) as their classification algorithm which only used 3 features out of available 19 features. The process of building

such a structure is explained in Sect. 2.3.2. This simple and easily understood proposal had a correct classification rate of 89% and 75% for low risk and high risk patients respectively.

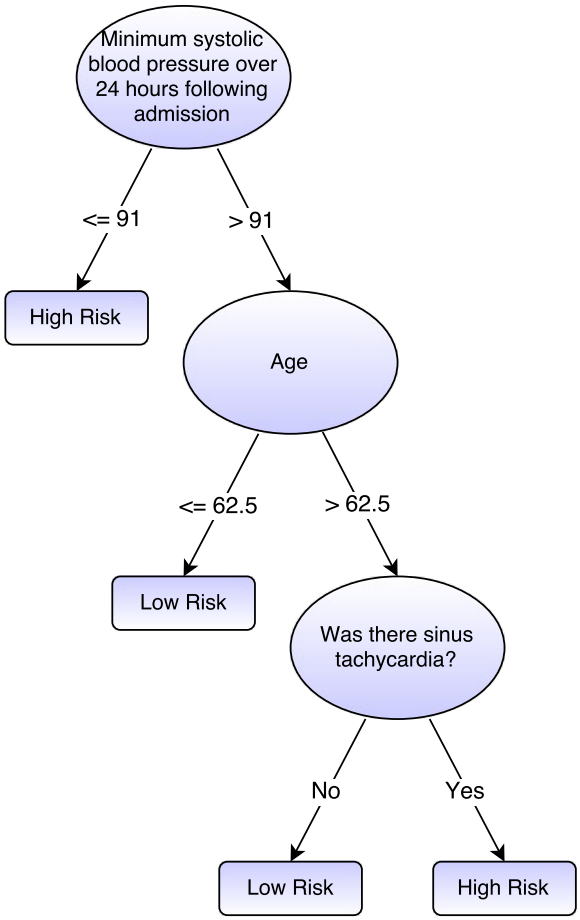


Figure 2.1: Tree like structure proposed by Breiman et al. in order to classify the 215 patients as low risk and high risk.

Enough terminology is presented in order to define the concept of supervised learning mathematically. Following is the Murphy's definition of supervised learning [Murphy, 2012].

Definition: Supervised Learning

Given a training set $\mathcal{D}$, learning a mapping $\mathcal{F}$ from the input $\mathcal{X}$ to output $\mathcal{Y}$ ($\mathcal{F} : \mathcal{X} \rightarrow \mathcal{Y}$) is the goal of supervised learning. When a previously unseen input $\mathbf{x}$ is present, the learned mapping ($\mathcal{F}(\mathbf{x})$) is

used to estimate (predict) the corresponding output $\hat{\mathbf{y}}$. ($\hat{\mathbf{y}}$ is used to denote the prediction as this is an estimation of the real output $\mathbf{y}$.)

Using the mapping proposed through the tree like structure of the example, the risk of a new patient who suffered a heart attack can easily be classified as high or low. Three ordered questions; namely,

1. minimum systolic blood pressure over 24 hours following admission,
2. age of the patient, and
3. whether the patient has elevated rate of impulse (sinus tachycardia)

lead to the final classification.

The next section introduces the concept of a decision tree and its main components without going into deeper discussions about its inner workings.

2.2 Decision Trees

The fundamental building block of random forests is the concept of decision trees. We introduce the main components of a decision tree in Sect. 2.2.1 before concisely addressing the humble beginning and the rapid evolution of the concept in Sect. 2.2.2. We present the split nodes and leaf nodes in more detail in the final three sub sections. In Sect. 2.2.3, we introduce the node split function and various parameters involved with it. Details on how to evaluate the *strength* of a split are presented in Sect. 2.2.4 before presenting how to make a prediction using the leaf nodes in Sect. 2.2.5.

2.2.1 Introduction

This section provides a simple portrait of the main components of a decision tree.

Definition: Decision Trees

In Chikalov [2013], a decision tree is defined as a well structured hierarchical entity that recursively partitions a set of objects into subgroups of objects that are more similar within the subgroups, using some features of the objects.

The above definition highlights three main characteristics of decision trees.

1. *A decision tree is hierarchical.* A decision tree is composed of two types of nodes. Namely, *split* (*internal*) nodes, and *leaf* (*terminal*)

nodes (see Fig. 2.2). Each split node has 2 or more children depending on whether the tree is binary or n-ary. The first split node is called the *root*. Except for the root node, all the nodes have only one parent. Leaf nodes do not have any children. Decision trees are hierarchical as they adhere to the strict hierarchy mentioned above. Additionally, every path that starts from the root is always terminated at a leaf node without forming any cycles. During this path traversal, it may traverse through one of more split nodes but only always terminated at a unique leaf node.

2. *A decision tree recursively partitions the input space into groups of objects that are more similar.* Attached with every split node, is a decision function who's sole responsibility is to partition the incoming objects into 2 or more splits depending on the n-arity of the decision tree. These decision functions try to maximize the node purity with respect to some measurement or statistic. As every path that starts from the root to a leaf possesses one or more split nodes, the incoming objects (hence, the input space) will be recursively partitioned.

   The most suitable decision function for each split node will be found at the training phase in such a manner that the similarity between the partitioned groups of objects is higher than the similarity of all objects that come into the split node. Finally, the objects accumulated at a leaf node will have the highest similarity of the input features so that this information will be used to do the prediction at the testing phase.

3. *A decision tree uses some features to do the partitioning.* The similarity of objects are measured using some of the features of the objects (*i.e.*, any property or measurement of the object or any calculation that can be performed on it). Although an object may possess extremely large number of features (*e.g.*, for a voxel of an image, one can use its position, intensity, result of any image filter, *etc.*) only a handful of those are usually used for the partitioning. The feature responses will be compared against one or more thresholds in order to decide to which partition an object belongs to.

Definition: Feature Response

Given a data point and a specific feature, the value of that feature at that data point is called the feature response.

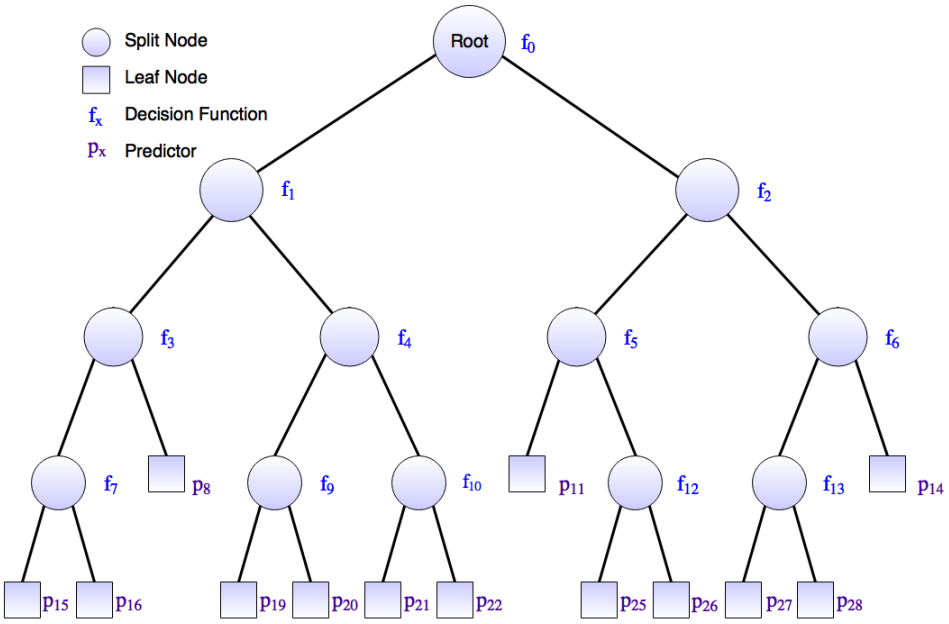


Figure 2.2: Main components of a decision tree. A decision tree consists of split nodes and leaf nodes. Each split node has a decision function that splits the incoming objects into two or more groups depending on the n-arity of the tree. As each split node results in a binary split in this example, it represents a binary tree. The objects of the new splits are more similar within the child node than the objects that arrived at the parent split node. Finally, leaf nodes store the similarity information that are later used for prediction.

In the next section, we concisely describe the evolution of decision trees in order to demonstrate the rapid growth of the concept within a time period of half a century.

2.2.2 Evolution of Decision Trees

To the best of our knowledge, the earliest documented instance resembling decision trees was found in a journal article by Belson in 1959 [Belson, 1959]. Until then, correlation procedures were used for classification, which were

complex and tedious to carry out by hand as the use of computers was not common during that era. The *Wherry-Doolittle* technique is a prime example for such a procedure [Garrett, 1947].

Belson termed his proposal "biological classification" where the aim of the study was to match two groups using certain characteristics as controls in order to compare the original two groups. Selecting a consumer panel to be representative of the public is an example of such a classification. The input objects were non-symmetrically divided using binary questions posed on the relevance of the control characteristics leading to a binary tree.

Belson's proposal was so much simpler to carry out without complex or tedious mathematical calculations, that it would be fitting to quote his concluding remarks: "*The method as I have described it is, it is true, a movement towards a more empirical way of doing things; but it is just as much a movement away from a sophistication which is too often either baffling or misleading*". Even to date, the simplicity and understandability of decision trees are unmatched to the machine learning "black box" models such as neural networks.

The idea proposed by Belson was first introduced as a tree based computer program called Automatic Interaction Detector (AID) by Morgan and Sonquist in 1963 [Morgan and Sonquist, 1963]. AID was proposed to handle predictions in survey data and the proposal resulted in a binary decision tree that was used exclusively for regression. During the years that followed, many improvements, adjustments, and changes were proposed to the existing decision tree algorithms [Morgan and Messenger, 1973; Friedman, 1977, 1979; Quinlan, 1979, 1986, 1993; Kass, 1980; Breiman and Stone, 1978; Breiman et al., 1984; Loh and Vanichsetakul, 1988; De Ville, 1990; Buntine, 1992; Clarke, 1992; Loh and Shih, 1997; Geurts et al., 2006; Özuysal et al., 2007]. However, the current norms of decision trees largely take shape from the contributions made by Breiman et al. [1984] and Quinlan [1986].

Many informative and thorough reviews have been published on various aspects of decision trees over the years, undoubtedly, owing to their popularity and usefulness [Safavian and Landgrebe, 1991; Murthy, 1998; Lim et al., 1998, 2000; Kothari and Dong, 2000; Loh, 2011; Kotsiantis, 2013]. Although we do not attempt to convey all the details covered in those reviews, we make a conscious effort to present the most important details in a concise manner. For further details, please refer to the reviews mentioned

above.

2.2.3 Node: Split Function

The sole responsibility of a split node (j) of an n-ary tree is to split the set of incoming objects ($\mathcal{X}_j$) into n disjoint subsets:

$$\mathcal{X}_j = \{\mathcal{X}_{j,1} \cup \mathcal{X}_{j,2} \cup \cdots \cup \mathcal{X}_{j,n}\} \quad , \tag{2.1}$$

where $\mathcal{X}_{j,i} \cap \mathcal{X}_{j,k} = \varnothing$, $\forall\, i \neq k$. If the split is made into two disjoint subsets, then it is termed *binary* splitting (see Fig. 2.3a). It results in *left* and *right* child nodes. If the split results in more than two disjoint subsets, then it is a *n-ary* split (see Fig. 2.3b) which creates n child nodes.

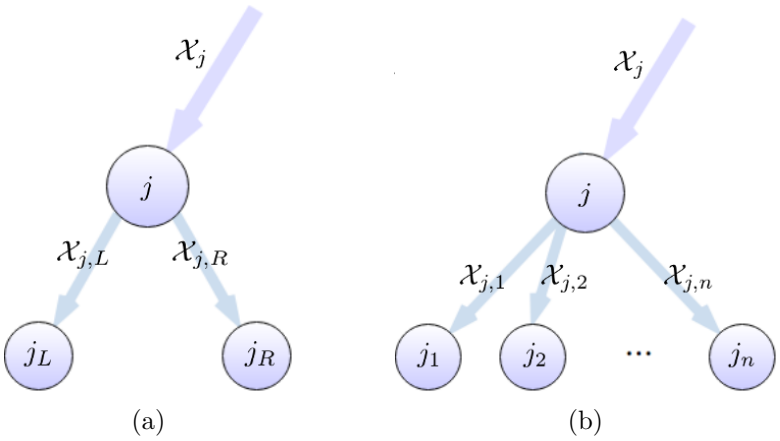


Figure 2.3: Types of splits. (a) A binary split. This results in two child nodes, named *left* and *right* child. (b) An n-ary split. This results in n child nodes.

As mentioned previously, an input object $\mathbf{x}$ (also termed *data point*) is described using its features in the following manner:

$$\mathbf{x} = (x_1, x_2, \ldots, x_D) \in \mathbb{R}^D \quad . \tag{2.2}$$

Although the dimensionality of the feature space (D) can be extremely large, practically, only a small number of features (d) are used for the tree induction process ($d \ll D$). Let the feature selection per node be denoted by:

$$\boldsymbol{\phi}(\mathbf{x}) = (x_{\phi_1}, \ldots, x_{\phi_d}) \in \mathbb{R}^d \quad . \tag{2.3}$$

Then, depending on the feature selection per node $\boldsymbol{\phi}(\mathbf{x})$, two types of split functions exist:

1. Univariate split functions: for a given split node, only a single feature is selected by $\boldsymbol{\phi}(\mathbf{x})$ to participate in the split function (see Sect. 2.2.3.1).

2. Multivariate split functions: for a given split node, more than one feature is selected by $\boldsymbol{\phi}(\mathbf{x})$ to participate in the split function (see Sect. 2.2.3.2).

The split function at the split node j with an incoming data point $\mathbf{x}$ (see Fig. 2.4) can be formulated in the following manner:

$$f\left(\mathbf{x}, \boldsymbol{\theta}_j\right) : \mathbb{R}^d \times \mathcal{T} \rightarrow \text{split} \ , \tag{2.4}$$

where $\boldsymbol{\theta}_j$ denotes the split parameters associated with split node j, $\boldsymbol{\theta}_j = (\boldsymbol{\phi}_j, \boldsymbol{\psi}_j, \boldsymbol{\tau}_j)$, and $\boldsymbol{\theta}_j \in \mathcal{T}$. $\mathcal{T}$ is the space of all split parameters. $\boldsymbol{\phi}_j(\mathbf{x})$ selects which features are used for the split function. The geometric primitive $\boldsymbol{\psi}_j$ is used to separate the data points in the feature space. Furthermore $\boldsymbol{\psi}$ is defined as a geometric surface in the feature space (see Fig. 2.5). The interaction between $\boldsymbol{\phi}_j(\mathbf{x})$ and $\boldsymbol{\psi}_j$ produces a scalar value s ($s \in \mathbb{R}^1$). Finally, s is compared to the threshold values given in the threshold vector $\boldsymbol{\tau}_j$ to find the corresponding partition of the data point. The n-arity of the decision tree determines how many threshold values ($|\boldsymbol{\tau}_j|$) are compared to s (see Fig. 2.6).

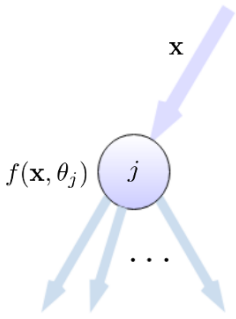


Figure 2.4: The split function at the split node j is responsible for sending an incoming data point $\mathbf{x}$ down the correct branch of the node.

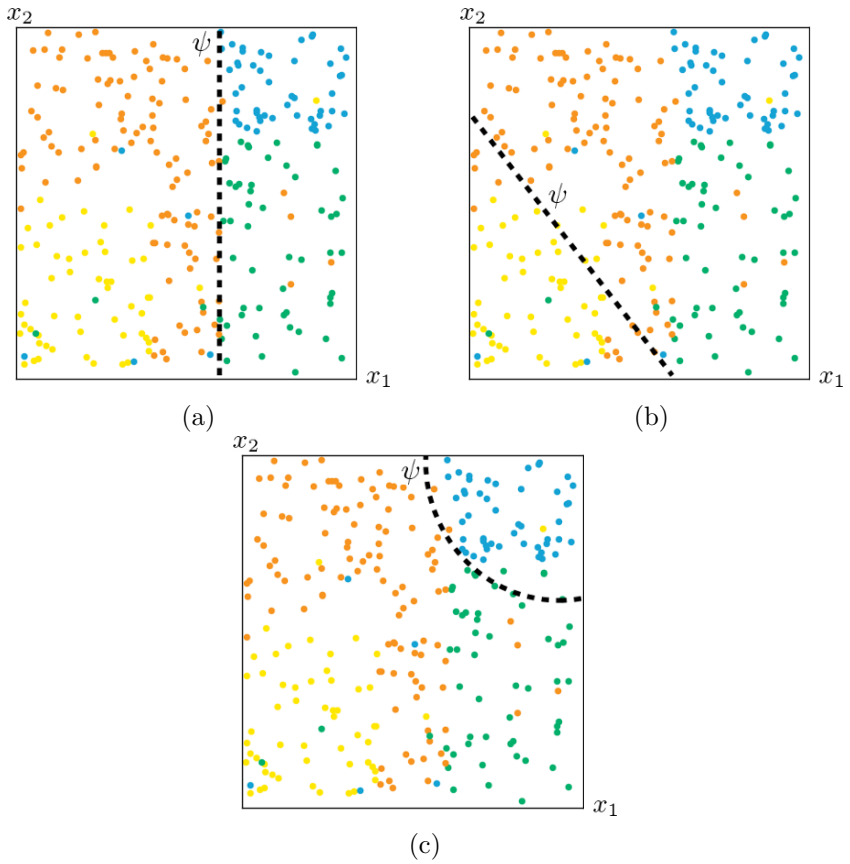


Figure 2.5: Different split functions. Different colors of the points indicate different classes. The distribution of the data points are given in the 2D feature space. (a) An axis–parallel univariate split function. (b) A multivariate split function that defines an oblique hyperplane in the feature space. (c) Another multivariate split function that defines a non–linear hyper surface. (The figure is inspired by Criminisi and Shotton [2013].)

In order to illustrate the split function of a decision tree, here is a simple 1D example:

Assume the following 1D feature distribution $\mathcal{X} = x$ and its corresponding output $\mathcal{Y} = y$ in Fig. *Ex.* 2.1.

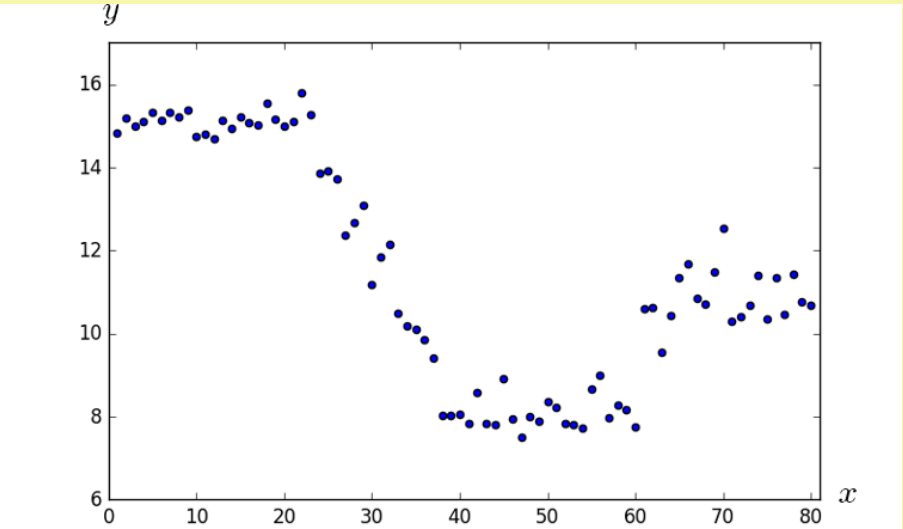


Figure *Ex.* 2.1: The feature distribution and its corresponding output.

Next, assume that the above illustrated distribution is estimated by the following partitioning of the input feature space as given by Fig. *Ex.* 2.2.

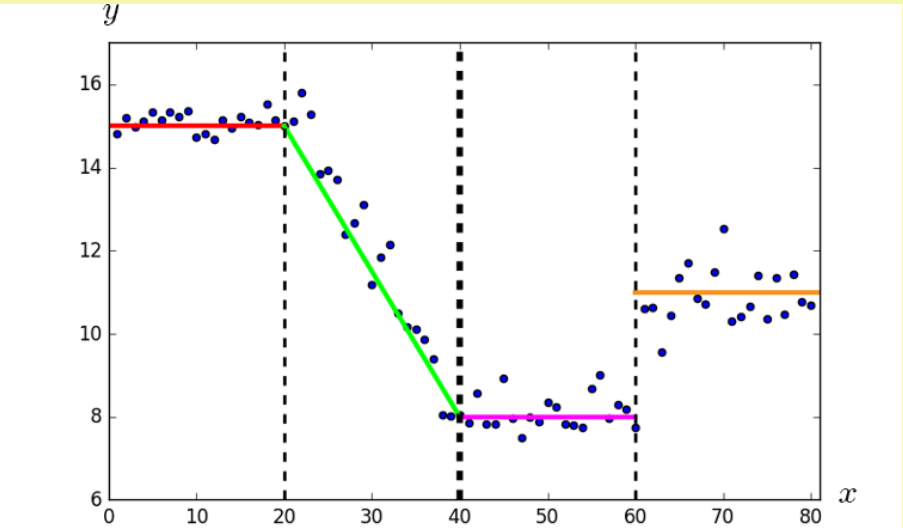


Figure *Ex.* 2.2: The piecewise continuous distribution estimation.

This is equivalent to the regression tree provided below in Fig. *Ex.* 2.3.

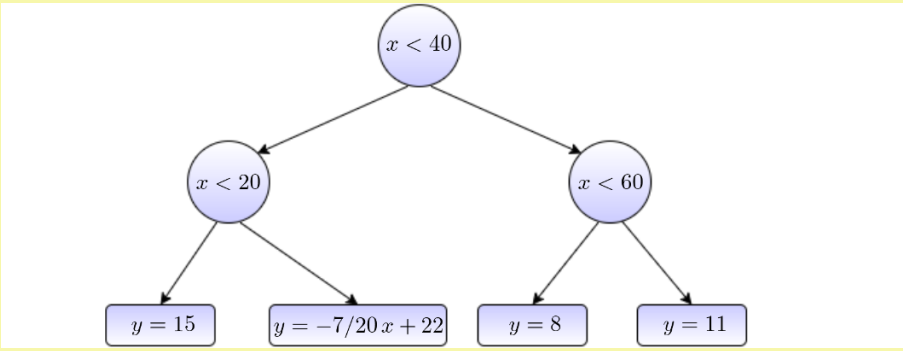


Figure *Ex.* 2.3: The corresponding regression tree that partitions the whole input feature space.

For demonstration purposes, let us evaluate the split function by using an unseen data point $x = 52$. At the root node, the comparison $x < 40$ is false. Hence, the data point goes to the right child node. Then, at the new split node, the comparison $x < 60$ is true. Consequently, the data point moves to the left child node (*i.e.,* the 3$^{\text{rd}}$ leaf node). The path taken by the data point $x = 52$ is presented in Fig. *Ex.* 2.4.

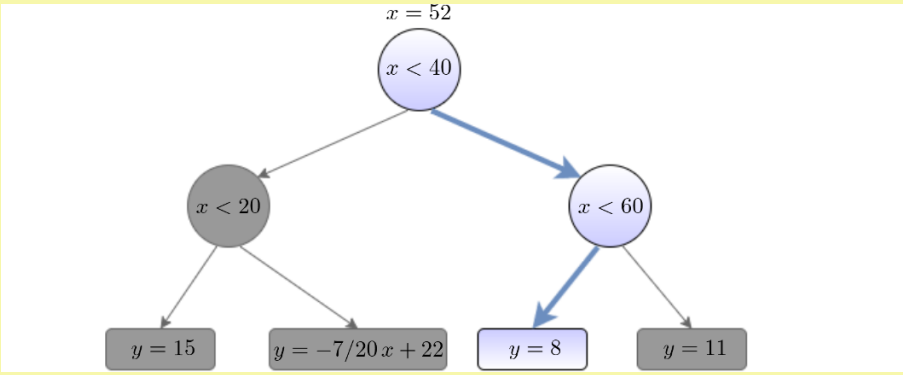


Figure *Ex.* 2.4: The path taken by the unseen data point $x = 52$.

2.2.3.1 Univariate Split Functions

The most popular splitting mechanism found in the literature is univariate splitting. Using only one feature for the split function out of all available

features, *i.e.*, only one non–zero component of $\boldsymbol{\phi}_j(\mathbf{x})$, results in a univariate split function. In this scenario, the interaction between $\boldsymbol{\phi}_j(\mathbf{x})$ and $\boldsymbol{\psi}_j$ can be described by the dot product $(\boldsymbol{\phi}_j(\mathbf{x}) \cdot \boldsymbol{\psi}_j)$ where $\boldsymbol{\psi}_j = (0\,0\,\ldots\,1\,\ldots\,0\,\lambda)$ using the homogeneous coordinates. The non–zero component of $\boldsymbol{\psi}_j$ corresponds to the non–zero feature and λ is the scaling factor of the homogeneous coordinates. Univariate split functions produce axis–parallel splitting of the multi–dimensional feature space (see Fig. 2.5a).

Univariate axis–parallel splitting leads to a binary split if the feature response s of $\boldsymbol{\phi}_j(\mathbf{x}) \cdot \boldsymbol{\psi}_j$ is compared in such a manner that produces a *true* or *false* result (see Fig. 2.6a). This is achieved by comparing s to one threshold value ($s < \tau$ or $s > \tau$) or one range of values ($\tau_1 < s < \tau_2$). Univariate axis–parallel splitting may also result in an n–ary split if s is compared to many threshold values (see Fig. 2.6b) in the following manner:

$$\text{split of } j = \begin{cases} j_1, & \text{if } s < \tau_1 \\ j_2, & \text{if } \tau_1 \leq s < \tau_2 \\ \ldots & \\ j_n, & \text{if } s \geq \tau_n \end{cases} . \tag{2.5}$$

A few of the examples for univariate binary splits can be found in [Morgan and Sonquist, 1963; Breiman et al., 1984; Buntine, 1992; Loh and Shih, 1997] whereas examples for univariate multi splits can be found in [Quinlan, 1979, 1986; Loh and Vanichsetakul, 1988; De Ville, 1990].

Univariate split functions are the most popular due to their ease of implementation and interpretability. Since only one feature is used at a time for splitting, univariate split functions result in deep decision trees which may not be efficient. It may also lead to weaker classifications than trees where multivariate split functions were employed for the same tree depths.

2.2.3.2 Multivariate Split Functions

Using more than one feature for the split function out of all available features, *i.e.*, multiple non–zero components of $\boldsymbol{\phi}_j(\mathbf{x})$ results in a multivariate split function. Multivariate split functions may result in a linear combination of features or non–linear combination of features. Similarly to the univariate split functions, the multivariate split functions may result in bi-

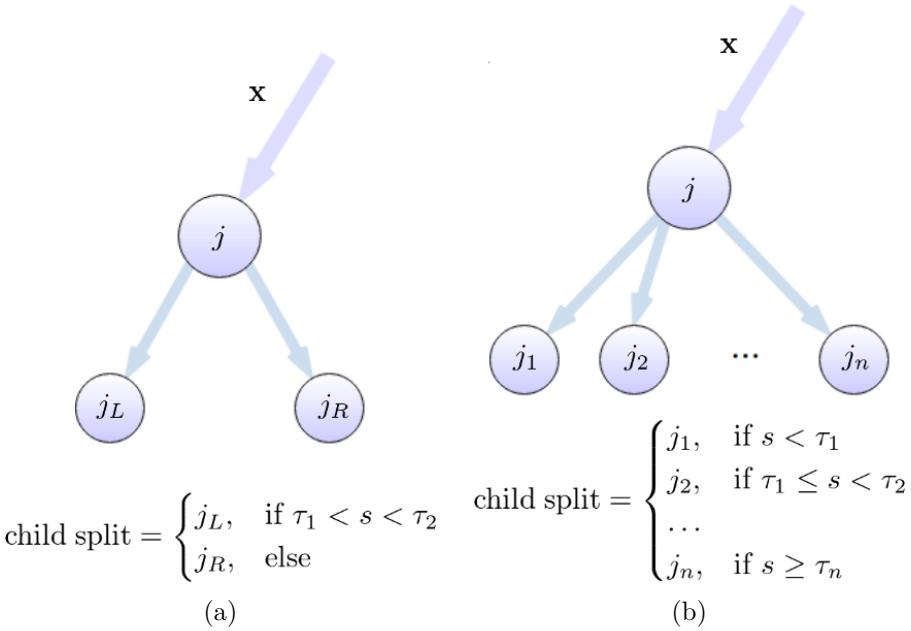


Figure 2.6: The relationship between the number of thresholds and the number of splits. (a) A binary split is generated by a single *true* or *false* comparison. $s < \tau$ or $s > \tau$ can also be used instead of $\tau_1 < s < \tau_2$. (b) An n–ary split is generated by a range of comparisons. s is the result of the interaction between $\boldsymbol{\phi}_j(\mathbf{x})$ and $\boldsymbol{\psi}_j$.

nary or n–ary splits depending on the number of thresholds used.

When a linear combination of features is used for splitting, the resulting decision trees are termed *oblique trees* as they result in *oblique* (slanted) hyperplanes of the feature space (see Fig. 2.5b). The interaction between $\boldsymbol{\phi}_j(\mathbf{x})$ and $\boldsymbol{\psi}_j$ can be described by $\boldsymbol{\phi}_j(\mathbf{x}) \cdot \boldsymbol{\psi}_j$ in this scenario too, where $\boldsymbol{\psi}_j = (0\, 0\, \ldots\, \boldsymbol{\psi}_{j,i}\, \ldots\, \boldsymbol{\psi}_{j,k}\, 0\, \lambda)$ using the homogeneous coordinates and the non–zero components of $\boldsymbol{\psi}_j$ correspond to the features selected for the splitting by $\boldsymbol{\phi}_j(\mathbf{x})$. A few examples of oblique trees are present in the studies of Breiman et al. [1984]; Loh and Vanichsetakul [1988]; Utgoff and Brodley [1991]; Murthy et al. [1993, 1994].

When a non–linear combination of features is used for splitting, the interaction between $\boldsymbol{\phi}_j(\mathbf{x})$ and $\boldsymbol{\psi}_j$ is modeled in the following manner:

$$\left(\boldsymbol{\phi}_j\left(\mathbf{x}\right)\right)^{\mathsf{T}} \boldsymbol{\psi}_j\, \boldsymbol{\phi}_j\left(\mathbf{x}\right) = s \ , \tag{2.6}$$

where $\boldsymbol{\psi}_j$ denotes a $\mathbb{R}^{(d+1)\times(d+1)}$ matrix in homogeneous coordinates. A

hyper surface in the feature space is the result of the split (see Fig. 2.5c).

New features were generated using the original primitive ones by Ittner and Schlosser in their work on Non-linear Decision Trees (NDTs) [Ittner and Schlosser, 1996]. The authors argued that using not only linear combinations of features (as oblique trees would), but also products of certain features could increase the discrimination capabilities of the newly *augmented* features. As an example, they argued and provided empirical evidence that the augmented feature *Petal Area* generated from the product of two features (*Petal Length* and *Petal Width*) of the famous *Iris* dataset[1] was better at classification than a linear combination of those features. Many other mechanisms such as Polynomial-Fuzzy Decision Trees (PFDTs) [Mugambi et al., 2004], model ensemble-based nodes [Altınçay, 2007], and Rough Set-based Multivariate Decision Trees (RSMDTs) [Wang et al., 2012] were based on generating multivariate splits.

In the next section, we discuss how the *strength* of a split can be evaluated.

2.2.4 Split Node Evaluation

At the training phase, at the split node j, depending on the selected split parameters ($\boldsymbol{\theta}_j = (\boldsymbol{\phi}_j, \boldsymbol{\psi}_j, \boldsymbol{\tau}_j)$) many split configurations are generated. Selecting the best split among the possible splits is a key characteristic of the decision tree algorithms. Mathematically, this can be interpreted as the maximization of an objective function $\mathcal{I}$ at the split node j in the following manner:

$$\boldsymbol{\theta}_j = \underset{\boldsymbol{\theta} \in \mathcal{T}}{\arg\max}\ \mathcal{I}(\mathcal{X}_j, \boldsymbol{\theta}) \ , \tag{2.7}$$

or as the minimization of an energy $\mathcal{W}$ at the split node j in the following manner:

$$\boldsymbol{\theta}_j = \underset{\boldsymbol{\theta} \in \mathcal{T}}{\arg\min}\, \mathcal{W}\left(\mathcal{X}_j, \boldsymbol{\theta}\right) \ . \tag{2.8}$$

In classification, the child nodes are expected to reduce the impurity of the parent node. In the same manner, the child nodes are expected to reduce the divergence of the parent node in regression. Consequently, any metric of impurity (or divergence) can generally be used to measure the

[1] The *Iris* dataset is a famous classification dataset of three different types of iris plants. It possessess four features. Namely, sepal length, sepal width, petal length, and petal width. The dataset can be accessed at https://archive.ics.uci.edu/ml/datasets/Iris

goodness of a split. Ultimately, the split that has the maximum impurity (or maximum divergence) reduction should be selected in classification (or regression). More generally the selection of best split can also be interpreted as a cost reduction scenario:

$$\begin{aligned} \Delta\text{cost} &= \text{cost}(\text{parent}) - \text{cost}(\text{children}) \ , \\ \boldsymbol{\theta}_j &= \arg\max_{\boldsymbol{\theta} \in \mathcal{T}} \ \Delta\text{cost} \ . \end{aligned} \tag{2.9}$$

The outputs (y_i's) of the training set are used in the calculation of these objective or energy or cost functions.

2.2.5 Leaf Node Decision

The final step in inducing a single decision tree, is to come up with the estimation or prediction model for the whole tree. Reducing the global generalization error of the entire decision tree is strictly equivalent to reducing the local generalization error of each leaf node [Louppe, 2014]. Consequently, the emphasis is on finding the best possible prediction models for the leaf nodes.

The leaf prediction models are built using the outputs (categorical or numerical) of the data points accumulated at the training phase. More precisely, this is done by employing conditional distributions. Given a data point $\mathbf{v}$ the conditional distribution for classification is given by:

$$p(y \mid \mathbf{v}) \ , \tag{2.10}$$

where y is the categorical output. Similarly, given a data point $\mathbf{v}$ the conditional distribution for regression is given by:

$$p(\mathbf{y} \mid \mathbf{v}) \ , \tag{2.11}$$

where $\mathbf{y}$ is the continuous output.

The most common prediction model is the Maximum *A–Posteriori* (MAP) model. For classification, MAP is defined in the following manner:

$$\hat{y} = \arg\max_{y} \ p\left(y \mid \mathbf{v}\right) \ . \tag{2.12}$$

For regression, MAP is defined as:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} p(\mathbf{y} \mid \mathbf{v}) \quad . \tag{2.13}$$

2.3 Analysis of Decision Trees

The following sections present our analysis of various aspects of decision trees. First, main components of supervised learning are analyzed in Sect. 2.3.1. The Sect. 2.3.2 presents the main steps of decision tree induction. Finally, Sect. 2.3.3 analyzes how the best split can be chosen among many splits.

2.3.1 Main Components of Supervised Learning

Irrespective of the algorithms used for the learning process and irrespective of the type of the output (classification or regression), a general supervised learning method consists of a workflow of 5 main components (see Fig. 2.7). They are as follows:

1. Training set: the labeled dataset used for training.
2. Dataset preprocessor: often, the data in the training set is not used as is. Certain processing steps are carried out on the training set in order to make the data suitable for the inducer to perform the estimation induction. Normalizing or resizing images are two examples of such manipulations.
3. Dataset: the output of the dataset manipulator is the exact data that is fed into the inducer. Hence, this is the real data used by the supervised learning algorithm.
4. Inducer: the algorithmic procedure that produces an estimation from the provided dataset. Consequently, decision trees are induction algorithms and the process of building a decision tree is called *tree induction*.
5. Estimator: the learned generalized relationship between the input and output. Given an unseen data point, the estimator is able to predict the output. An estimator can either be a classifier or a regressor.

The following section presents the main steps related to building decision trees.

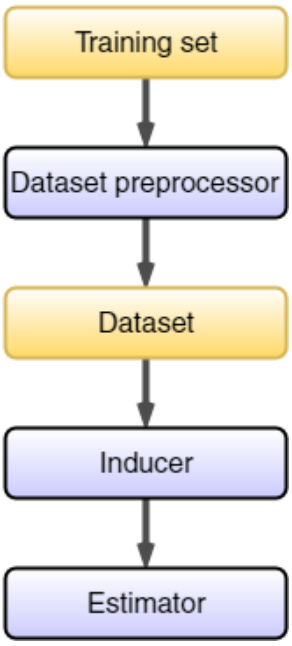


Figure 2.7: Main components of supervised learning. Data components are shown in yellow whereas algorithmic components are shown in blue. (The figure is inspired by Rokach [2010].)

2.3.2 Main Steps of Decision Tree Induction

The process of building a decision tree is called *tree induction* as mentioned in Sec. 2.3.1. Following are the major choices that should be made in order to build a decision tree structure using the provided training data:

- *The n–arity of the decision tree.*
  The n–arity of the decision tree is decided by how many threshold ranges are compared to the feature response of a data point. The choice of the n–arity is generally made for the whole decision tree although it is possible to decide the n–arity at each split node.

- *Which split function to be used for the node splitting?*
  For a given node, split functions may involve either only one feature, a linear combination of features or a non–linear combination of features. Although each object may possess a large number of features, only a handful of features are used to build a decision tree as the use of all features may not be necessary.

  In a linear or a non–linear combination of multiple features there are

a myriad of possibilities. Though many wise schemes have been proposed, finding the best possible combinations of features is still a hard task. Additionally, once some combinations are found, finding the corresponding thresholds adds to the difficulty of the multivariate split functions. However, if the correct feature combinations (linear or non-linear) and threshold values are found, it leads to shallower trees compared to univariate trees. For the same depth, multivariate trees result in better classification or regression comparatively to univariate trees.

Hence, the usual practice is to select a few pertaining features and use a limited number of predefined values per each selected feature as thresholds to generate split configurations.

- *How to select the best splitting configuration among many configurations?*
  As mentioned previously, the selected metric should be able to measure the decrease of the impurity (or the increase of the purity) resulting from the splitting of the node. Functions based on Shanon's entropy [Shannon and Weaver, 1949], Gini index [Lerman and Yitzhaki, 1984], and squared error loss [Barbieri and Berger, 2004] are some of the measures used to select the split configuration that maximizes the decrease of the impurity among the available split configurations. This will be analyzed in more detail in Sect. 2.3.3.

- *When to stop the node splitting?*
  While tree structures grow exponentially, computer resources such as memory are not unlimited. This often leads to a compulsory stopping criterion by limiting the maximum depth of the tree depending on the physical limitations and capabilities of the hardware used. If splitting a node does not substantially decrease the impurity (or increase the purity) of the splits, it may also indicate that further splitting is not necessary. Additionally, leaf nodes are often expected to accumulate a collection of training objects rather than a single object. Hence, a limitation on a minimum number of objects accumulated at a node may also serve as a stopping criterion.

- *How to model the similarity of the objects collected at a leaf node?*
  If the decision tree is used for classification, a leaf node often has a

representation of all possible classes and percentage of votes per each class. In the case of regression, leaf node may simply have a real valued vector or a collection of vectors depending on whether the regression operation is univariate or multivariate. Additionally, some notion of the confidence of the proposed similarity is expected to be saved along with the similarity measures.

- *Should the tree be pruned?*
  Once a decision tree is built, a post processing step called pruning can be carried out to improve the strength of the tree.

  Definition: Overfitting
  Overfitting occurs when the prediction model models the training data *too* well. It models the noise of the training data and fails to generalize to unseen data.

  Definition: Tree Pruning
  Tree pruning is a mechanism that reduces the size and complexity of a decision tree by removing certain nodes that provide weak classification/regression outputs or that result in overfitting the training data.

2.3.3 Selecting The Best Split

Best split selection measures are considered separately for classification and regression since classification is concerned with categorical outputs whereas regression is concerned with real valued outputs.

2.3.3.1 Best Split Selection in Classification

Details of a few of the famous best split selection criteria for classification are presented in this section. We make use of the Fig. 2.8 depicting a splitting scenario of a split node j for the rest of this section. Let us assume the following for a given split node j:

- the total number of categories denoted by $\mathcal{C} = \{1, 2, \ldots, c\}$.
- The total training data points at node j is N.
- The number of data points that belong to class i is N_i such that $\sum_{i=1}^{c} N_i = N$.

- The number of partitions at node j is s.
- The number of data points that each partition k possesses is N^k such that $\sum_{i=1}^{c} N_i^k = N^k$.
- And finally, the following $\sum_{k=1}^{s} N_i^k = N_i$ and $\sum_{k=1}^{s} \sum_{i=1}^{c} N_i^k = N$ hold.

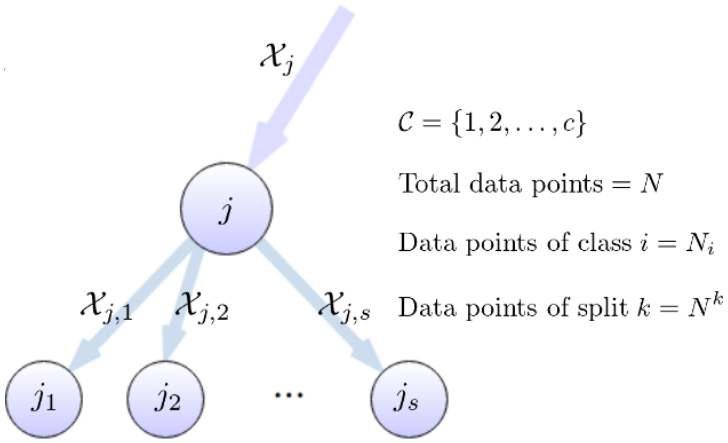


Figure 2.8: Best split selection in classification.

Using the Eq. (2.9) page 25, the impurity reduction at split node j with s resulting splits can be described as a weighted cost of child nodes:

$$\Delta\text{cost} = \text{cost}(\mathcal{X}_j) - \sum_{k=1}^{s} p_k \, \text{cost}(\mathcal{X}_{j,k}) \ , \tag{2.14}$$

where $p_k = \frac{N^k}{N}$ and $\mathcal{X}_{j,k}$ is the dataset that accumulates at the partition k (see Fig. 2.8).

In the next sections we present a few of the main measures used to select the best split configuration in classification, namely, the misclassification rate, Gini index, information gain, and information gain ratio. Finally, we carry out a comparison study of these methods.

Misclassification rate: given a split k, the class label of the split can be defined as the most probable label inside it in the following manner:

$$\hat{y}_k = \arg\max_{i \in \mathcal{C}} \ p_{i,k} \ , \tag{2.15}$$

where $p_{i,k} = \frac{N_i^k}{N^k}$. Then, the misclassification rate *(MR)* or the error rate for the split k is defined as mentioned below:

$$MR_k = (1 - p_{\hat{y}_k}) \quad . \tag{2.16}$$

Then, the split configuration that minimizes the misclassification rate across all the splits can be selected as the best split. That is:

$$\boldsymbol{\theta}_j = \underset{\boldsymbol{\theta} \in \mathcal{T}}{\arg\min} \left((1 - p_{\hat{y}_j}) - \sum_{k=1}^{s} (1 - p_{\hat{y}_k}) \right) \quad . \tag{2.17}$$

Gini index: misclassification error rate had the shortcoming of not choosing the pure splits over the impure splits. Gini index was proposed by Breiman et al. in the famous Classification And Regression Tree (CART) induction algorithm to overcome the above mentioned weakness [Breiman et al., 1984]. Given a split k, the Gini index (or the expected error rate) (GI) is written in the following manner.

$$GI_k = \sum_{i=1}^{c} p_{i,k} \left(1 - p_{i,k}\right) = \sum_{i=1}^{c} p_{i,k} - \sum_{i=1}^{c} p_{i,k}^2 = 1 - \sum_{i=1}^{c} p_{i,k}^2 \quad . \tag{2.18}$$

The split configuration that produces the smallest Gini index reduction will be selected as the best split. That is:

$$\boldsymbol{\theta}_j = \underset{\boldsymbol{\theta} \in \mathcal{T}}{\arg\min} \left(1 - \sum_{i=1}^{c} p_j^2 - \sum_{k=1}^{s} \left(1 - \sum_{i=1}^{c} p_{i,k}^2 \right) \right) \quad . \tag{2.19}$$

Information gain: it is one of the most common measure employed in finding the best split [Bolón-Canedo et al., 2013]. Shanon's entropy is often used to measure the information gain. The reduction in entropy (or the gain of information - IG) due to the reduction of impurity caused by the splitting is measured in the following manner:

$$IG = H\left(\mathcal{X}_j\right) - \sum_{k=1}^{s} \left(\frac{N^k}{N} \right) H\left(\mathcal{X}_{j,k}\right) \quad , \tag{2.20}$$

where $H\left(\mathcal{X}_j\right)$ is the Shanon's entropy. And Shanon's entropy is defined as follows:

$$H\left(\mathcal{X}_j\right) = -\sum_{i=1}^{c} p\left(\mathcal{X}_i\right) \log\left(p\left(\mathcal{X}_i\right)\right) \quad , \tag{2.21}$$

where $p(\mathcal{X}_i) = \frac{N_i}{N}$. Then, Eq. (2.20) can be written in the following manner:

$$\begin{aligned} IG = &- \left[\sum_{i=1}^{c} \left(\frac{N_i}{N}\right) \log\left(\frac{N_i}{N}\right)\right] \\ &- \left[\sum_{k=1}^{s} \left(\frac{N^k}{N}\right) \left(-\sum_{i=1}^{c} \left(\frac{N_i^k}{N^k}\right) \log\left(\frac{N_i^k}{N^k}\right)\right)\right] \end{aligned} \tag{2.22}$$

Shanon's entropy based information gain criterion was first used in the Iterative Dichotomiser 3 (ID3) algorithm by Quinlan [1986].

Information gain ratio: Quinlan demonstrated that information gain criterion had a strong bias in favor of tests that resulted in many splits. In order to reduce this tendency, a normalizing factor (g) based on entropy was proposed in Quinlan [1993]. That is:

$$g = -\sum_{k=1}^{s} \left(\frac{N^k}{N}\right) \log\left(\frac{N^k}{N}\right) \tag{2.23}$$

Then, he defined the information gain ration as the *normalized* information gain:

$$IG \text{ ratio} = \frac{IG}{g} \tag{2.24}$$

Behavior of Impurity Measures

In order to study the different behavior of the above mentioned impurity measures let us consider the following example. Assume a two object class (c_1 and c_2) classification example with equal number of data points of each class at the parent node ($N_{c_1} = N_{c_2}$). At one leaf node caused by the split, if $p_{c_1} = p$ then $p_{c_2} = 1 - p$. Then, the misclassification rate, Gini index, and entropy for that leaf node (l) can be written in the following manner:

$$\begin{aligned} MR(l) &= 1 - \max(p, 1-p) \ , \\ GI(l) &= 2p(1-p) \ , \text{ and} \\ H(l) &= -p\log(p) - (1-p)\log(1-p) \ . \end{aligned} \tag{2.25}$$

The corresponding graph of the three impurity measures is presented in Fig. 2.9. All three impurity measures are maximum when the *a posteriori* probabilities of the classes are the same (i.e., $p = (1-p) = 0.5$). And they

are minimum when the nodes are pure (i.e. only when one type of objects gets accumulated at the node or when $p = 0$ or $p = 1$).

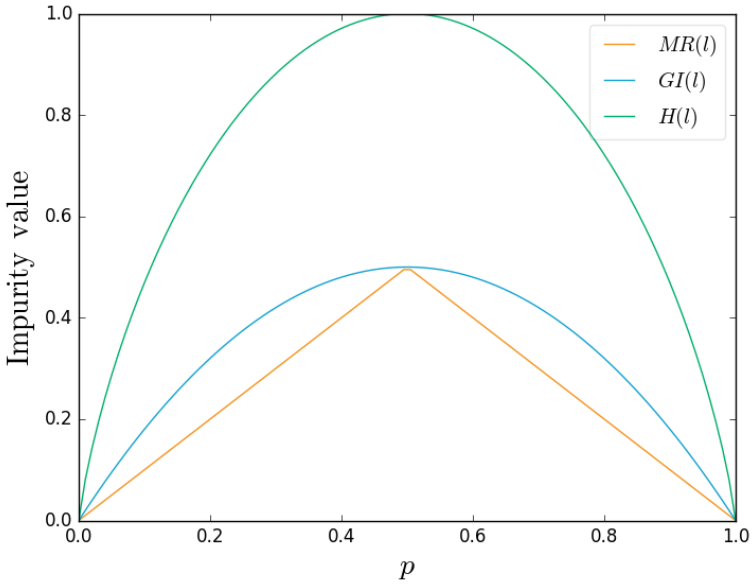


Figure 2.9: Node impurity measure variation for two class binary classification. Horizontal axis corresponds to the posteriori probability of class $c_1 = p$. All three measures are minimum when the node is pure (i.e., when $p = 0$ or $p = 1$). Maximum measure values are obtained when the class probabilities are equal (i.e., when $p = (1 - p) = 0.5$). (The figure is inspired by [Murphy, 2012].)

2.3.3.2 Best Split Selection in Regression

In regression, since the output variable is quantitative, the available impurity measures change accordingly.

Squared error loss: the simplest of impurity measure in regression is the squared error loss. At a split k, the squared error loss is defined in the following manner:

$$cost_k = \frac{1}{N^k} \sum_{\mathbf{x} \in \mathcal{X}_{j,k}} (\mathbf{y}_\mathbf{x} - \bar{\mathbf{y}})^2 \ , \tag{2.26}$$

where $\bar{\mathbf{y}} = \frac{1}{N^k} \sum_{\mathbf{x} \in \mathcal{X}_{j,k}} \mathbf{y}_{\mathbf{x}}$. Then, the best split can be selected as the maximum squared error loss reduction split:

$$\boldsymbol{\theta}_j = \arg\max_{\boldsymbol{\theta} \in \mathcal{T}} \left(\frac{1}{N} \sum_{\mathbf{x} \in \mathcal{X}_j} (\mathbf{y}_{\mathbf{x}} - \bar{\mathbf{y}})^2 - \sum_{k=1}^{s} \left(p_k \frac{1}{N^k} \sum_{\mathbf{x} \in \mathcal{X}_{j,k}} (\mathbf{y}_{\mathbf{x}} - \bar{\mathbf{y}})^2 \right) \right) \ , \tag{2.27}$$

where $p_k = \frac{N^k}{N}$.

Information gain: the information gain criterion is defined in the same manner as in Eq. (2.20) but instead of Shanon entropy, the differential entropy is used owing to the fact that the output variable is continuous. The differential entropy is defined as follows:

$$H(\mathcal{X}_j) = -\int_{\mathbf{y} \in \mathcal{Y}} p(\mathbf{y}) \log \left(p(\mathbf{y}) \right) d\mathbf{y} \ , \tag{2.28}$$

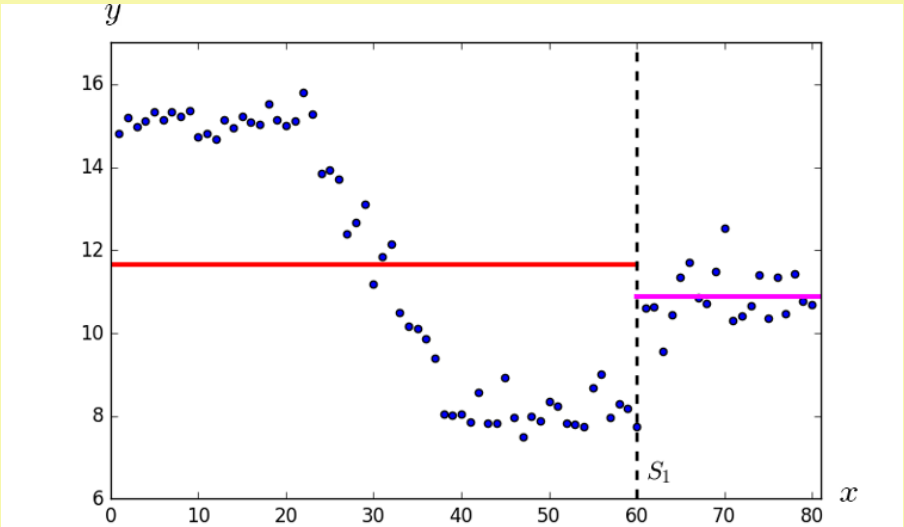
where $p(\cdot)$ is the probability density function. It needs to be estimated from the available data points of $\mathcal{X}_j$. Generally, $p(\cdot)$ is approximated by Gaussian–based models in the following manner:

$$H(\mathcal{X}_j) = \frac{1}{2} \log \left((2\pi e)^d \left| \Lambda(\mathcal{X}_j) \right| \right) \ , \tag{2.29}$$

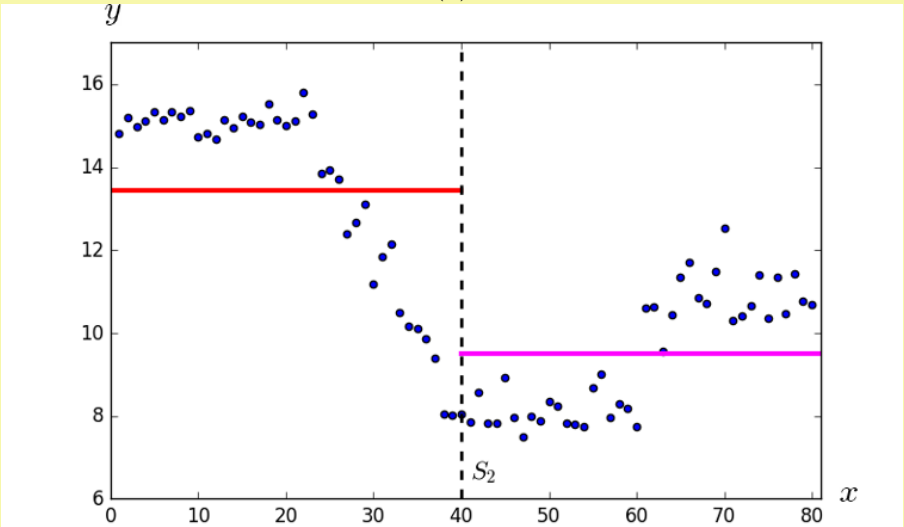
where $\Lambda(\mathcal{X}_j)$ is the $d \times d$ dimensional covariance matrix.

The best split selection using the squared error loss is illustrated using the same example presented in Sect. 2.2.3.

Assume that we want to choose the best split for the root node. Two possible splits (S_1 and S_2) are presented in Fig. *Ex.* 2.5a. and Fig. *Ex.* 2.5b. respectively.



(a)



(b)

Figure *Ex.* 2.5: Best split selection between S_1 (a) and S_2 (b) using the squared error loss as presented in Eq. (2.27). The squared error losses are $S_1 = 0.11$ and $S_2 = 3.86$. Consequently, S_2 is chosen as the best split as it produces the maximum squared error loss reduction.

The information required to calculate the best split between S_1 and S_2 are presented in Table *Ex.* 2.1. The squared error loss reduction (Er) is calculated below for both S_1 and S_2 using Eq. (2.27):

$$\mathrm{Er}_{S_1} = 7.79 - (0.75 \times 10.10 + 0.25 \times 0.40) = 0.11 \ ,$$

and

$$\mathrm{Er}_{S_2} = 7.79 - (0.50 \times 5.65 + 0.50 \times 2.21) = 3.86 \ .$$

S_2 is selected as the best split since it results in a greater reduction of the squared error than S_1 does.

	S	S_1		S_2	
		$S_{1,L}$	$S_{1,R}$	$S_{2,L}$	$S_{2,R}$
N	80	60	20	40	40
$\bar{y}$	11.46	11.65	10.88	13.42	9.50
cost	7.79	10.10	0.40	5.65	2.21
p	1.00	0.75	0.25	0.50	0.50

Table *Ex.* 2.1: Information required to calculate the squared error loss using Eq. (2.27). L and R denote the left split and the right split respectively, S is the root node without any splits, N is the number of data points, $\bar{y}$ is the mean of all the data points of the partition, p is the fraction of data points of the partition and *cost* is the squared error loss calculated using Eq. (2.26).

Although a lot of literature can be found on best split selection measures [Ben-Bassat, 1982; Shih, 1999; Kothari and Dong, 2000; Arauzo-Azofra et al., 2011; Bolón-Canedo et al., 2013], they generally agree that none of the methods are superior than the others in all situations. According to those studies the optimal suitability of a split selection measure depends on the problem at hand. For a more detailed description refer the studies mentioned above.

In this chapter we first introduced the fundamental concepts of machine learning before introducing decision trees. Then, we gave a detailed description of the main components of a decision tree along with a concise description of the relevant literature. The node split function, split node evaluation, and prediction using leaf nodes were detailed. Then, we carried out an in–depth analysis of the main steps of the decision tree induction process as well as how to select the best split among many splits.

In the next chapter, we will follow the same approach where the Random Regression Forest concept and the related literature will be presented first. Then, the key areas of RRF concept will be analyzed in detail. Finally, we will conclude the chapter with a summary of the algorithms used for training a Random Regression Forest (RRF) and the algorithms used to predict the localization of multiple organs in Computed Tomography (CT) images using an already trained RRF.

An approximate answer to the right problem is worth a good deal more than an exact answer to an approximate problem.

- John Tukey

3

Random Regression Forests and Their Analysis

Contents

This chapter is dedicated to the analysis of the Random Regression Forest methodology in the context of multi–organ localization.

First, we present the two main types of ensemble frameworks and a number of ensemble combination methods in Sect. 3.1. Then, in Sect. 3.2, we give a concise description of the evolution of Random Forests (RFs), different types of RFs and their various applications in the field of medical image analysis. Decision jungles and random ferns are described in Sect. 3.3.

In Sect. 3.4, we present an in–depth analysis of Random Regression Forests (RRFs). As we are interested in multi–organ localization using RRFs, in Sect. 3.4.1, we introduce the concept of multi–organ localization. The intrinsic parameters of a RRF are presented in Sect. 3.4.2. Then, we detail the training set preparation and image preprocessing steps in Sect. 3.4.3 and Sect. 3.4.5 respectively. The training phase and the concepts related to the training phase are described in detail in Sect. 3.4.6. Similarly, we present the prediction phase and the related concepts in Sect. 3.4.7. Finally, in Sect. 3.5.1 and Sect. 3.5.2, we summarize the algorithmic steps of forest training and prediction with the use of a few illustrations.

A random forest is an ensemble (collection) of random decision trees. A Random Regression Forest (RRF) is a special type of random forest. Ensemble methods are a very powerful concept in machine learning. We present an overview of ensemble methods in the next section.

3.1 Ensemble Methods

The main purpose of an ensemble method is to combine multiple base estimators (classifiers or regressors) in order to arrive at a combined estimator which improves the prediction than any single base estimator alone would have performed [Rokach, 2010]. Rokach identifies two families of ensemble methods:

- dependent ensemble frameworks, and
- independent ensemble frameworks.

3.1.1 Dependent Ensemble Frameworks

In a dependent ensemble framework, the base estimators are built in cascade and each new estimator is built using the output of the previous estimator (see Fig. 3.1). Hence, the knowledge from the previous iteration is employed to guide the learning of the current iteration. Finally, the output of all iterations are combined in some manner.

A few examples on this family of ensemble methods are AdaBoost [Freund and Schapire, 1996], variations of AdaBoost such as Real AdaBoost [Friedman et al., 2000] or Ivoting [Breiman, 1999], and gradient tree boosting [Friedman, 2001]. For further details please refer the corresponding articles.

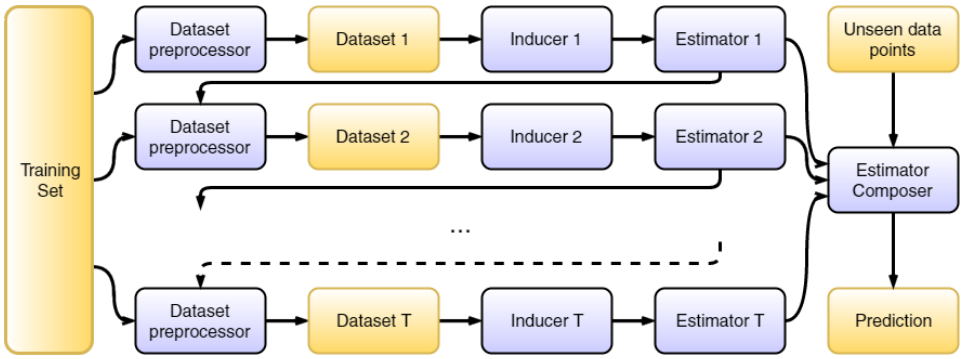


Figure 3.1: A dependent ensemble framework. The output of the previous estimator is input to the next where the process of estimation is further refined. Finally, at the estimator composer, a combined output is generated from the output of each estimator. The data components are shown in yellow and the algorithmic components are shown in blue. (The figure is inspired by Rokach [2010])

For most of the above mentioned algorithms, the entire training set should be loaded to the main memory. In addition to that, if the first base estimator is of poor quality, then, the remaining cascade of estimators produce a low quality ensemble too. Since the output of the first base estimator is propagated throughout the following cascade of estimators, overfitting can easily occur [Quinlan, 1993]. These are the main criticisms of the dependent ensemble frameworks.

3.1.2 Independent Ensemble Frameworks

The main feature of an independent ensemble framework is to build several estimators independently and combine the output of each estimator in some manner to obtain the final estimation. The entire training set is divided into disjoint or overlapping datasets which are in turn used to learn separate estimators (see Fig. 3.2).

One advantage of this process is the ability to use independent induction algorithms if desired. Since each estimator is learned independently, the learning process can easily be parallelized too. Additionally, the independent ensemble frameworks do not possess the weaknesses of the dependent ensemble frameworks that were due to the dependency among the estimators.

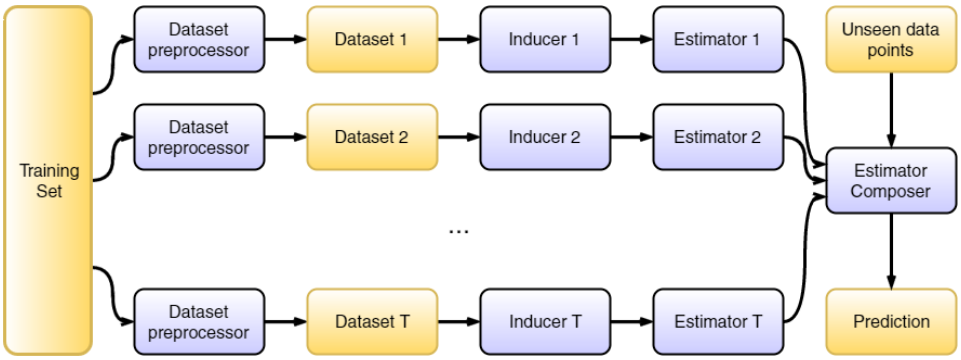


Figure 3.2: An independent ensemble framework. Estimators are learned independently of each other either by using disjoint or overlapping datasets from the training set. Finally, a combined output is produced at the estimator composer. The data components are shown in yellow and the algorithmic components are shown in blue. (The figure is inspired by Rokach [2010])

Breiman proposed the concept of *bagging* (bootstrap aggregating) which is one of the most famous independent ensemble frameworks. Each estimator is trained using a sub sample of some size, taken from the training set with replacement. Bagging methods are suitable for unstable inducers as bagging reduces this instability and improves the estimator accuracy [Breiman, 1996].

Another famous sibling of the family of independent ensemble framework is the random forest ensemble. Instead of training each split node of a decision tree with all available features, a relatively small subsamples of features are chosen randomly for the training. This helps in reducing the correlation among the individual trees which in turn leads to better estima-

tors as a composed ensemble [Breiman, 2001]. Since the entire feature space is divided into small subsamples, it is relatively easy to work with a very high dimensional feature spaces.

Although the concept of random ensembles were originated with decision trees, it can be applied to any supervised learning technique. A more detailed overview of the concept of random forests is presented in Sect. 3.2.

3.1.3 Ensemble Combination Methods

Irrespective of whether the ensemble method is dependent or independent, combining the outcome of each estimator is a very important step. Rokach [2009] identifies two main families of combination methods. Namely, weighting combination methods, and meta-learning combination methods. We describe the weighting combination methods and provide a concise analysis of them in Sect. 3.1.3.1 and Sect. 3.1.3.2 respectively.

3.1.3.1 Weighting Combination Methods

As the name suggests, each learned estimator is given a fixed or dynamic weight that contributes to the final ensemble estimation. Some of the most popular weighting combination methods are presented below.

Simple averaging: This is the simplest combination method available. Each estimation is given a similar weight as all the estimations from each inducer is simply averaged in the following manner:

$$p(\mathbf{y}|\mathbf{v}) = \frac{1}{T}\sum_{t=1}^{T} p_t(\mathbf{y}|\mathbf{v}) \ , \tag{3.1}$$

where $p_t(\mathbf{y}|\mathbf{v})$ is the posterior distribution estimated by the t^{th} inducer and T is the number of inducers.

Simple multiplication: If independence can be assumed among the different inducers, then instead of averaging all the estimations from each inducer, one can multiply each estimation to arrive at the final estimation:

$$p(\mathbf{y}|\mathbf{v}) = \frac{1}{Z}\prod_{t=1}^{T} p_t(\mathbf{y}|\mathbf{v}) \ , \tag{3.2}$$

where $p_t(\mathbf{y}|\mathbf{v})$ is the posterior distribution estimated by the t^{th} inducer, T is the number of inducers and Z is the normalizing constant ensuring unit probability.

Performance weighting: The idea behind performance weighting is to give more weight to the estimators that are more accurate. The accuracy is measured using a validation dataset and the weights are calculated in the following manner:

$$\omega_t = \frac{1 - E_t}{\sum_{t=1}^{T} (1 - E_t)} \ , \tag{3.3}$$

where E_t is the error of the t^{th} inducer and accuracy is defined as $1 - E_t$. Then, the combined estimation is simply:

$$p(\mathbf{y}|\mathbf{v}) = \sum_{t=1}^{T} \omega_t p_t(\mathbf{y}|\mathbf{v}) \ , \tag{3.4}$$

where $p_t(\mathbf{y}|\mathbf{v})$ is the posterior distribution estimated by the t^{th} inducer and T is the number of inducers.

In meta-learning combination methods, the already learned estimators and their estimation of the training data are used to learn new estimators and combined at a higher level. Presenting more information on meta-learning combination methods is beyond the scope of this dissertation. For further information, refer to Rokach [2009, 2010].

3.1.3.2 Weighting Combination Methods – Analysis

The effect of the ensemble combination method on the final estimation is illustrated in Fig. 3.3. Each inducer produces a posterior estimation $p_t(\mathbf{y}|\mathbf{v})$ modeled as a Gaussian. More confident estimations result in taller and thinner Gaussian distributions whereas shorter and wider curves correspond to less confident estimations. When individual estimations are combined using simple averaging, the influence of less confident estimations can be observed too (see Fig. 3.3a). On the contrary, when simple multiplication is used, the influence of less confident estimations becomes almost invisible (see Fig. 3.3b). But in both cases, the final estimation is highly influenced by the most confident estimation.

We present random forests and different types of random forests in the

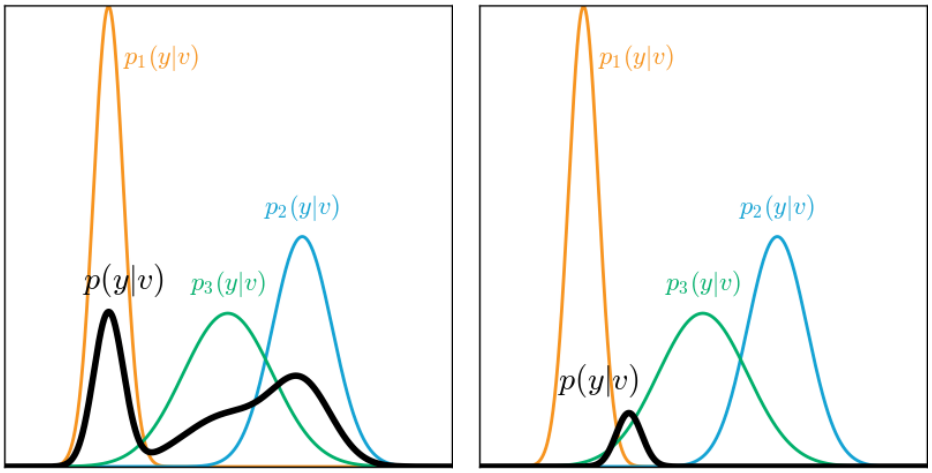


Figure 3.3: The effect of the ensemble combination method on the final estimation. The posterior distributions given by the three inducers are shown in color ($p_1(\mathbf{y}|\mathbf{v})$, $p_2(\mathbf{y}|\mathbf{v})$, and $p_3(\mathbf{y}|\mathbf{v})$). The combined esimation is presented by a thick black curve ($p(\mathbf{y}|\mathbf{v})$). (a) Estimations are combined using simple averaging. Final distribution displays evidence of the underlying individual distributions. (b) Estimations are combined as products of individual estimations ($Z = 4.7 \times 10^{-7}$). Final distribution loses any evidence of the underlying distributions. In both scenarios, the final distribution is highly influenced by the most confident individual estimation.

next section.

3.2 Random Forests

In this section, first, we provide a concise description of the evolution of random forests. Then, descriptions of types of random forests and their uses in the context of medical image analysis are presented.

As previously mentioned in Sect. 3.1.2, the concept of random forest is an independent ensemble method. The fundamental idea is to train each split node of a decision tree with a relatively small subsample of features chosen randomly instead of using all available features.

3.2.1 Evolution of Random Forests

The first reported use of randomization in literature is found in the work of Amit and Geman on hand written digit recognition using decision trees

[Amit and Geman, 1994]. They stated that using random sub-sampling of features reduces the dependence among the trees and that this led to powerful ensembles only using simple averaging as the combination method. Ho in a parallel study proposed a similar strategy of building multiple trees using randomly selected subspaces of the entire feature space also on hand written digit recognition [Ho, 1995]. Both Amit and Geman and Ho built a single decision tree using the entire feature space and compared it with an ensemble of decision trees built using random sub-samples of features to empirically prove the superiority of the random ensemble. This idea is the foundation of this dissertation.

As previously shown in the Sect. 3.1.2, bagging was proposed by Breiman as another concept to reduce the correlation between the decision trees in order to increase the estimation capacity. The concepts of bagging and randomized sub-samples of features were first employed together by Ho [1998] who proposed the *random subspace* method.

In 2001, Breiman proposed the famous study titled *Random Forests* where the concepts of bagging [Breiman, 1996] and randomization [Amit and Geman, 1994; Ho, 1995] were employed in combination to carry out classification and regression tasks using decision trees.

Since the establishment of random forests in 2001 by Breiman many improvements and/or variations have been proposed by various researchers [Robnik-Šikonja, 2004; Rodriguez et al., 2006; Geurts et al., 2006; Bernard et al., 2009; Baumann et al., 2014].

In the proposal of Robnik-Šikonja [2004], the author employed five split selection measures to build different trees that reduced the correlation between the trees. They also proposed a voting weight mechanism where they discarded certain trees entirely for certain classes.

The idea of *rotation forest* was proposed by Rodriguez et al. [2006] where they generated new features from the original features. They split the feature space into a number of subsets and carried out principal component analysis to obtain the new features. This is interpreted as a rotation over the number of axis similar to the number of feature space splits, hence the name *rotation forest*.

Extra trees proposed by Geurts et al. [2006] randomly chooses not only the feature sub-space but also the thresholds of the split function at each split node. In contrast to other methods described so far, the algorithm does

not use bagging, instead, it uses the whole training set for each tree. In the most extreme case, only one feature and one threshold is selected resulting in extremely randomized trees. They are extremely fast to build as no split node optimization needs to be carried out. The authors provided empirical results that showed that extremely randomized trees performed as well as random forest algorithms.

Bernard et al. provided empirical evidence of the existence of a subset of random trees that lead to reduced generalization error compared to using the full set of random trees built by the classical induction algorithms [Bernard et al., 2009]. Hence they proposed to do a selection of the induced random trees that are added to the final ensemble using techniques such as sequential forward selection or sequential backward selection [Hao et al., 2003] instead of adding all the induced trees arbitrarily.

Inspired by the AdaBoost technique, Baumann et al. [2014] advocated assigning a class specific vote to each leaf node of the forest. Votes were based on the depth of the tree rather than on an error measure. A linear combination of weighted leaf nodes were used to obtain the final estimation rather than using a majority voting technique.

Although different categorizations exist in the literature, we find there are four main types of random forests. They are:

- classification forests,
- Hough forests,
- clustering forests and
- regression forests.

The following sections present brief introductions about these four different types of random forests. Additionally, a concise summary of their applications in the field of medical image analysis is also included for each type of random forest.

3.2.2 Classification Forests

Undoubtedly, Random Classification Forests are the most famous members of the family of random forests. A Random Classification Forest (RCF) is a pure ensemble of randomly trained classification trees. The output of each classification tree would be purely categorical.

RCFs have been employed to solve classification problems in numerous application fields. In medical image analysis, RCFs have been used to detect

and localize multiple organs in Computed Tomography (CT) images [Criminisi et al., 2009], delineate the myocardium in echocardiographic images [Lempitsky et al., 2009], segment multiple sclerosis lesion in Magnetic Resonance (MR) images [Geremia et al., 2011] among many other applications.

Although Criminisi et al. [2009] have used RCFs to detect and localize multiple organs in CT images, the same authors claimed that the localization results using RRFs were twice as better than the results obtained using RCFs on identical training and prediction sets [Criminisi et al., 2010]. Consequently, we also used RRFs for multi–organ localization.

3.2.3 Hough Forests

Duda and Hart [1972] demonstrated how to use the Hough transform not only to detect straight lines in images but curves as well. The image points were transformed into its quantized parameter space. Each potential line in the image space would then cast a vote in the parameter space. The maximum votes in the parameter space would correspond to lines in the image space. A generalization of the Hough transform was proposed to detect generic parametric shapes by Ballard [1981]. Presently, *Hough transform* generally describes any detection procedure that is based on aggregation of *Hough* votes originated from an image or image sequences [Gall et al., 2011].

According to [Gall and Lempitsky, 2009], a Hough forest can be described as a classification forest augmented with spatial consistent information about the centroid of the bounding box of the object[1]. Similarly to a classification forest, the split nodes try to minimize the impurity of the categorical labels (classes). Additionally, they also try to minimize the variance of the spatial information regarding the centroid of the bounding boxes of the respective classes. Generally, only one of these two uncertainty measures is considered during the selection of the best split. Finally, the leaf nodes contain not only the proportions of training objects that belong to each class but also the spatial information about the centroids.

At the testing phase, the testing data points that get accumulated at leaf nodes cast a probabilistic vote (generally, a Gaussian vote) about the

[1] In that sense, it is quite counter intuitive to call them *Hough* forests as in this scenario, spatial information are added in the image space but not in a parametric space as the original Hough transform proposes to. The casting of votes by the forest leaves is similar to the original Hough transform and it is the only link with the original method.

centroid of the detected object weighted by the class proportion saved during the training phase. Although each vote casted by a leaf may not be coherent and point to the same location, all votes casted by the the leaves of the forest taken together usually lead to the correct location.

Hough forests have been used in general machine learning contexts of object detection [Gall and Lempitsky, 2009], object detection and tracking [Gall et al., 2011], and pose estimation [Tejani et al., 2014]. In the context of medical image analysis, Hough forests have been employed to segment the left ventricle [Milletari et al., 2014], prostate [Zettinig et al., 2015], and multiple organs (left ventricle, prostate, and mid brain) [Milletari et al., 2015] in ultra sound images.

3.2.4 Clustering Forests

Decision forests recursively partition the input feature space. Even if no supervision is provided (*i.e.*, no output labels for classification or no output vectors for regression, hence, no training data), a decision forest still partitions the input feature space. Blockeel et al. [1998] describes this behavior as the inherent behavior of clustering. Hence, a collection of decision trees without any supervision can be interpreted as a clustering forest where every clustering tree of the forest yields a clustering of the input feature space.

Since no training data is provided, the objective function has to be different from the known classification or regression counter parts. If the data points in the node are assumed to be distributed according to a Gaussian distribution, the information gain criteria of the input features can be used as the objective function similar to Eq. (2.20) in page 31. The entropy (H) of a node (j) is described in the following manner.

$$H\left(\mathcal{X}_j\right) = \frac{1}{2}\log\left((2\pi e)^d|\Lambda(\mathcal{X}_j)|\right) \quad , \tag{3.5}$$

where $\mathcal{X}_j$ is the input data points at the node, d is the dimension of the input features, and $\Lambda(\mathcal{X}_j)$ is the covariance matrix. Similarly, each leaf node would be described using a Gaussian distribution which describes the cluster.

During the testing phase, one would be interested to know to which cluster a previously unseen data point belongs to. The data point would be collected at a leaf per tree of the forest. And finally the ensemble model will average the estimation from each tree to define the final cluster of the data

point.

The idea of clustering forests have been used in generating codebooks[2] that were eventually used for image classification tasks [Moosmann et al., 2007, 2008] including object classification in CT images [Mouton et al., 2014].

3.2.5 Regression Forests

The second most famous type of random forests is Random Regression Forests (RRFs). They differ from their famous sibling, RCFs as the output of RRFs is continuous whereas the output of RCFs is categorical. They have been used to analyze medical images (from Sect. 3.2.5.1 to Sect. 3.2.5.3) and other types of images (Sect. 3.2.5.4).

3.2.5.1 Medical Images Related Uses

The pioneering work of Criminisi et al. in 2010 lead to the first application of RRFs for multi-organ localization in medical images (CT images) [Criminisi et al., 2010]. The ingenuity of the proposal was the manner in which the multi-organ localization problem was transformed into a multivariate regression problem. It comprised the regression of a 6D displacement vector from the bounding box walls of the organs to any given voxel. In this pioneering work, displaced appearance features (mean intensities of displaced cuboids) were employed to describe the given CT images. In 2013, the same team of researchers proposed a modified implementation of RRFs that enhanced the previously achieved results by modifying the split node optimization method, the description of the random process, and the eventual usage of this description for prediction [Criminisi et al., 2013].

Pauly et al. proposed a similar solution containing RRFs to detect and localize multiple organs in MR images [Pauly et al., 2011]. In contrast to the CT images, the appearance values (gray level values) of MR images are not directly related to the measured physical entity[3]. Consequently, they proposed a new feature set called local binary pattern descriptors in order to describe the MR images. These descriptors result is a binary feature vector that is built using the water and fat channels of the MR image sequence.

[2] A *codebook* is essentially a collection of *visual words* where a visual word is a vector of dense or sparse local image descriptors.

[3] The graylevel values of CT images are directly related to the Hounsfield Unit (HU). HU scale is the quantitative scale of radiodensity and is obtained from a linear transformation of the measured attenuation coefficients.

The above mentioned studies of RRFs in the process of organ localization in CT and MR images have paved way to many interesting solutions to numerous pertinent problems in medical image analysis. These contributions can mainly be described in two categories: 1.) localization and/or segmentation of bone structures (see Sect. 3.2.5.2) and 2.) localization and/or segmentation of soft-tissue organs (see Sect. 3.2.5.3).

3.2.5.2 Bone Structure Localization and/or Segmentation

RRFs have been used for the following research studies related to shoulders [Tschannen et al., 2016], vertebrae [Glocker et al., 2012; Roberts et al., 2012; Bromiley et al., 2015], hand [Donner et al., 2013; Ebner et al., 2014], and pelvis [Chen and Zheng, 2013, 2014; Chu et al., 2014, 2015; Lindner et al., 2012].

Recently, Tschannen et al. [2016] used RRFs to find the articular margin plane in shoulder arthroplasty[4] in CT images. First, the parameters of the articular margin plane (the center of the plane and the two angles of the normal of the plane relative to the CT image plane) were roughly estimated by a RRF using the displaced mean intensity cuboid features. The first coarse estimation was then refined using a cascade of 2 RRFs using a new feature type introduced in the study called *sheetness-based ray features*.

RRFs have been used for the automatic identification and localization of vertebrae in medical images. Glocker et al. [2012] used RRFs to roughly detect and localize all vertebrae visible within a given CT image as the first step of the two step procedure. They used displaced cuboid appearance features. In a 3 phase procedure to localize vertebrae in DXA[5] images, Roberts et al. [2012] employed RRFs with Haar features in the first phase to locate vertebral endplates. Bromiley et al. [2015] exploited RRFs to define a constrained local model to localize vertebrae in DXA images.

In localizing landmarks in hand X-ray and full body CT images, Donner et al. [2013] used a RCF first to get a rough localization before refining the localization with a RRF. In the final stage the authors used a graph theory technique to obtain the final predictions. Ebner et al. [2014] proposed a

[4]Shoulder arthroplasty is the orthopedic surgical procedure that replaces the shoulder joint by an artificial prosthesis.

[5]Dual-energy X-ray Absorptiometry (DXA) is an enhanced X-ray technology that measures bone mineral density and bone loss.

two step landmark localization in hand CTs images which involved RRFs in both steps. First coarse landmark localization took into consideration bigger and long range displaced cuboid appearance features. The second cascade localization step used smaller and short range features with training and testing voxels used only around the vicinity of the first estimations.

The femur and pelvis bones of 2D anteroposterior pelvis X-ray images were fully automatically segmented by Chen and Zheng [2013] in a two step procedure. In the first step, an image normalization was carried out by localizing 22 global landmarks involving left pelvis, right pelvis, left femur, and right femur bones using a RRF per landmark. Estimated landmarks were then fitted to the global statistical shape model learned using the 22 landmarks from the training phase. In the second step, a shape optimization was carried out similarly. The local landmarks of left pelvis and left femur consisting of 59 and 97 points respectively were estimated using another set of RRFs, one per landmark, before fitting the learned local shape models. For this study, image patches generated by multi-level histogram of oriented gradients were used as features. The same authors proposed a novel version of the method where they used features generated by flexible-level histograms of oriented gradients [Chen and Zheng, 2014]. The other improvement of the novel proposal was the pre-selection of the efficient features for training based on the trace ratio optimization mechanism.

RRFs were employed as the first landmark localization step of fully automatic hip joint segmentation in CT images [Chu et al., 2014, 2015]. The next steps of the procedure involved registering atlases over the first localization result before finally fitting an articulated statistical shape model to obtain the final segmentation. Mean and variance of voxel intensities of a displaced cuboid were used as the features for the study.

To segment the femur bones in anteroposterior pelvis X-ray images, Lindner et al. [2012] proposed a two step procedure using RRFs. First, a RRF was used to find the center of a reference frame, relative to which, the contours of the femur were segmented using another RRF.

3.2.5.3 Soft–Tissue Organ Localization and/or Segmentation

The following studies have employed RRFs in soft–tissue organ localization and/or segmentation. They include organs such as the brain [Kim et al., 2015; Han et al., 2014], heart [Zhen et al., 2014, 2016; Kelm et al., 2011], left

and right kidneys [Cuingnet et al., 2012], liver [Gauriau et al., 2013], and multiple abdominal organs comprising the stomach, liver, spleen, kidneys, and gallbladder [Gauriau et al., 2014].

Deep brain simulation treatments are crucial part of neuro-degenerative diseases such as Parkinson's disease. Only MR machines that produce very high magnetics fields, such as 7 T fields[6] are capable of direct visualization of these deep brain simulation structures. But 7 T MR machines are not commonly available whereas 1.5 T MR machines are. Kim et al. [2015] successfully proposed a method that learned a mapping of shape and pose parameters from 7 T MR machines to 1.5 T MR machines using RRFs.

Random Regression Forests were used to estimate the volume of left and right ventricles of the heart in MR images [Zhen et al., 2014, 2016]. The regression forest was used to map multivariate input (the feature vector) directly to bi-ventricular areas through which they calculated the volumes. Multi-scale 2D image patches were used as features. In the former study [Zhen et al., 2014], features were generated using pyramidal Gabor features and histogram of oriented gradients in addition to the appearance features. In the latter study [Zhen et al., 2016], an unsupervised 3 layer deep learning network was used to learn the features automatically. In a different study, Han et al. [2014] used RRFs to detect some landmarks in brain MR images that are used for registration.

In a study that was aimed at detecting, grading, and classifying coronary stenoses[7] in CT angiography, RRFs were used to estimate the lumen[8] of the coronary [Kelm et al., 2011].

Cuingnet et al. used a RRF to localize the left and right kidneys as the first step in their automatic kidney segmentation proposal [Cuingnet et al., 2012]. Then, they used the first localization estimations and their neighborhoods to train another RRFs to refine the location of the centroid of each kidney separately. The centroid prediction step was constrained to small displacements and was carried out multiple times until convergence.

In the generic and automatic work flow proposed for liver segmentation in CT images by Gauriau et al. [2013], RRFs were employed for the first step of liver localization. The same team of researchers proposed a modified

[6]Tesla (T) is the SI unit for measuring the strength of a magnetic field.

[7]The coronary stenosis is the abnormal narrowing of the coronary blood vessel.

[8]The lumen is the aperture within a tubular structure, in this case, the inside space of the coronary blood vessel.

approach where they used RRFs in a two step procedure to localize multiple organs in the abdomen cavity [Gauriau et al., 2014]. In the first step, a RRF was used to localize all organs simultaneously. In the second step, another RRF per organ was used to refine the localization using a pre–built probabilistic atlas and confidence maps.

3.2.5.4 Non–Medical Image Related Uses

There are a vast number of application domains that have acquired the services of RRFs to propose novel and ingenious solutions. We make an attempt to provide a few examples that are related to the domain of image analysis as it is the enclosing domain of the medical image analysis.

RRFs have been extensively used in *articulated pose estimation*, i.e., recovering configurations (poses) of people from images and/or image sequences. The 2D human poses are estimated from the color images. Among the many studies that have addressed the above mentioned problem, the studies by [Dantone et al., 2013, 2014] have proposed solutions using RRFs. Depth images are used to estimate the 3D human poses. Although a very famous algorithm on 3D human pose estimation by Shotton et al. [2011] employs only a RCF, the extension proposed by Girshick et al. [2011] instead employs a RRF. A very interesting proposal from Kostrikov and Gall [2014] details how to estimate the 3D humans poses from simple 2D color images using RRFs.

A special case of pose estimation is the case of head pose estimation. The pose of the head is generally expressed with respect to the 3D position of the nose and the angles of rotation of the head. Li et al. [2010]; Fanelli et al. [2011]; Tang et al. [2011] have proposed head pose estimation procedures incorporating RRFs.

Estimation of age from the 2D color images is another challenging problem to which solutions have been proposed with the application of RRFs [Montillo and Ling, 2009].

Though we have introduced 4 different types of random forests in this section 3.2, other types of categorizations exist in the literature. For a different and more informative categorization of random forests, please see Criminisi and Shotton [2013].

3.3 Beyond Random Forests

As mentioned previously, a random forest is an ensemble method where its base predictors are randomly trained decision trees. When the base predictor type diverges away from a tree (e.g., by a directed acyclic graph), the resulting ensemble is an entity that resembles a forest though not exactly a forest. In the sections that follow, we introduce two types of such ensemble methods. More precisely, decision jungles and random ferns.

3.3.1 Decision Jungles

The growth of a decision tree is exponential. Hence given enough data, in order to obtain acceptable estimation capabilities, the trees in a forest would need to be grown extremely large. Shotton et al. [2013] claimed that it may become impossible to grow or use random forests having decision trees with desired depths in a memory intensive environments such mobile or embedded applications. In order to address this shortcoming, they proposed decision jungles.

A decision jungle consists of directional acyclic graphs instead of binary trees. The memory consumption is decreased by reducing the number of nodes of a tree by introducing a node merging technique. Consequently, in decision jungles the path to a leaf node from the root is not unique (see Fig. 3.4).

The only difference in training a decision jungle from a decision forest are found in the best split selection and node merging steps. For further information, refer to Shotton et al. [2013].

3.3.2 Random Ferns

Random ferns were first proposed by Özuysal et al. [2007] in order to cater for the requirements of fast learning, fast estimation and reduced memory consumptions [Pauly, 2012]. Özuysal et al. assumed independence between features in order to propose this naive Bayesian method of random ferns.

Instead of hierarchically partitioning the input feature space like a decision tree (see Fig. 3.5a), a random fern partitions the whole feature space with each level of the fern, called a decision stump (see Fig. 3.5b). This is identical to building a decision tree with the same split function across the split nodes of the same level [Özuysal et al., 2010] (see Fig. 3.6). Each final

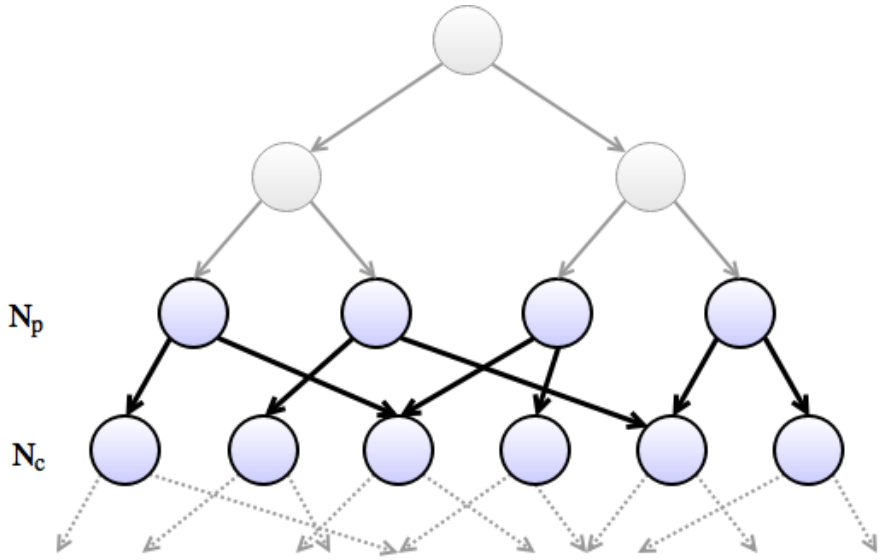
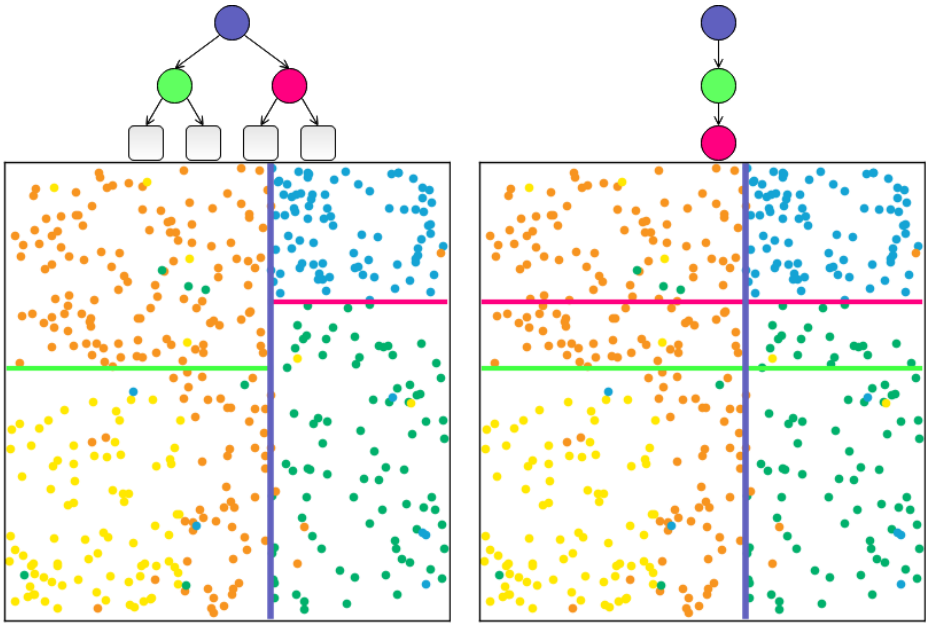


Figure 3.4: A directional acyclic graph. Each split node has two children whereas each node except the root node may have one or more parent nodes. The two colored levels are the parent level (N_p) and child level (N_c). The merge criterion in this scenario is $N_c = \lceil 1.5N_p \rceil$. This merging behavior reduces the exponential growth of the decision tree.

partition represents a multinomial distribution for different classes learned at the training phase. At the testing phase, the output of each fern was combined in a semi-naive Bayesian fashion to obtain the final estimation. For a more detailed description on random ferns, refer to Özuysal et al. [2007, 2010].

Pauly et al. [2011] have used random ferns to automatically localize multiple organs in MR Dixon sequences. Random ferns have been used in key point recognition Özuysal et al. [2007, 2010], image classification [Bosch et al., 2007], action recognition [Oshin et al., 2009], and capsule endoscopy[9] image classification [Li et al., 2014] among other applications.

[9]Capsule endoscopy is a mechanism that enables capturing the digestive tract through images, specially the small intestines. The system comprises a pill shaped miniature camera that the patient swallows, which in turn takes images of the digestive system until it passes with the fecal matter.



(a) (b)

Figure 3.5: The 2D feature space consists of data points belonging to four classes each presented in a different color. The resulting partitions are shown by lines having the same color as the corresponding nodes. (a) A tree partitions the input feature space hierarchically. (b) A fern partitions the entire input feature space by each decision stump. Consequently, ferns are not hierarchichal.

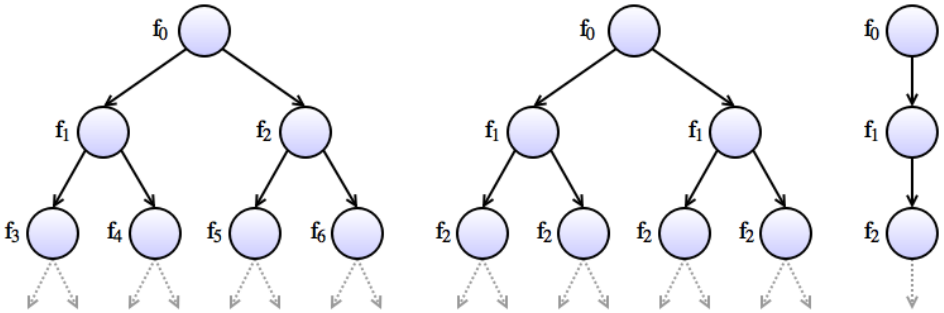


Figure 3.6: Each split node of a decision tree possesses a different split function (Left). A random fern (Right) can be built if all the decision functions in one level of the tree are the same (Middle). Hence, a fern is a specialization of a tree.

3.4 Analysis of Random Regression Forests

In the context of this dissertation, the ultimate goal of using a RRF is to fully automatically localize multiple organs in CT images. Hence, an in–depth analysis of RRFs in the context of multi–organ localization is presented in the following sections.

First and foremost, a concise analysis is carried out on the multi–organ localization in medical images in Sect. 3.4.1. The details on the intrinsic parameters of a RRF are presented in Sect. 3.4.2. In Sect. 3.4.3, we present the Random Regression Forest ensemble model. Then, we carry out a short analysis on the training set preparation and on the image preprocessing steps in Sect. 3.4.4 and Sect. 3.4.5 respectively. In Sect. 3.4.6 and Sect. 3.4.7, we thoroughly analyze the training phase and the prediction phase respectively.

To the best of our knowledge there are 6 studies that use random regression forests to automatically predict the localization of bounding boxes of multiple organs in medical images [Criminisi et al., 2010; Pauly et al., 2011; Cuingnet et al., 2012; Criminisi et al., 2013; Gauriau et al., 2013, 2014]. Among them, 3 of the studies focus solely on multi–organ localization only using Random Regression Forest without any other concept [Criminisi et al., 2010; Pauly et al., 2011; Criminisi et al., 2013]. The other three use RRFs either as a first localization step in a broader segmentation approach [Cuingnet et al., 2012] or use additional concepts to complement the use of RRF [Gauriau et al., 2013, 2014]. Although there are numerous other studies that use RRFs for many medical image analysis tasks (see Sect. 3.2.5.1), not much information is available on the RRFs. This may be due to the fact that RRFs are only used as a tool in a single step of multi step procedure where the main focus is not on RRFs or localization. Hence, our analysis of RRFs for multi–organ localization revolves around the 6 studies mentioned above. These studies are presented in Table 3.1.

All the random regression forests mentioned in Table 3.1 are composed of binary Random Regression Trees (RRTs). Hence our analysis only focuses on binary random regression trees. The analysis of n-ary RRTs is out of the scope of this dissertation. We often refer to or do comparisons with these 6 studies in Chap. 5, Chap. 6, and Chap. 7.

Study Num.	Study	Title	Pure RRF
(1)	Criminisi et al. [2010]	Regression forests for efficient anatomy detection and localization in CT studies	Yes
(2)	Pauly et al. [2011]	Fast multiple organ detection and localization in whole–body MR dixon sequences	Yes
(3)	Cuingnet et al. [2012]	Automatic detection and segmentation of kidneys in 3D CT images using random forests	No
(4)	Criminisi et al. [2013]	Regression forests for efficient anatomy detection and localization in computed tomography scans	Yes
(5)	Gauriau et al. [2013]	A generic, robust and fully automatic workflow for 3D CT liver segmentation	No
(6)	Gauriau et al. [2014]	Multi–organ localization combining global–to–local regression and confidence maps	No

Table 3.1: Main studies found in the literature that focus on multi–organ localization using RRFs. The studies are presented in the chronological order. The final column of the table signifies whether any additional concept is used. These studies are often referred or compared in Chap. 5, Chap. 6, and Chap. 7.

3.4.1 Multi–Organ Localization

Human brain–eye combination is extremely good at guiding the attention and eyes to the regions of interest in natural scenes [Oliva and Torralba, 2007]. Although localization of objects of interest in a scene is a natural process that often occurs involuntary for humans, it is very difficult for computers to have the same vision capabilities [Mibulumukini et al., 2013]. The innate ability of humans to model and understand context, and the inability of computers to do so have made the context modeling in computer vision a very pertinent research topic [Oliva and Torralba, 2007; Mibulumukini et al., 2013].

Analyzing a medical image in order to localize an organ can be accomplished by two contrasting approaches.

1. **By looking at the organ itself.** First, the organ is described in a

manner that a computer can interpret. The appearance of the organ (how its gray values are spatially spread), the contours of the organ, or the discriminating features of any image filter can be used for this description. Then, given an image, one can search which parts of the image is most similar to the description of the organ.

When multiple organs need to be localized simultaneously, this approach translates into repeating the process as many times as the numbers of organs. Consequently, this may not be the best way forward in multi–organ localization.

2. **By looking at the context of the organ.** Instead of describing the organ itself, in this approach an attempt is made to describe the context of the organ. One way of achieving this is by defining how the organ is spatially situated with respect to some land marks. An example of this would be describing the localization of the right kidney as below the liver, right to the spine, and above the right pelvis, *etc.* (see Fig. 3.7).

   Explicitly describing these contexts would mean localizing each organ explicitly (accordingly to the previous example, in order to localize the right kidney, the localization of the liver, spine, and right pelvis should be known in advance). Implicit description of these contexts would make it possible to use these context information without having to localize any of the organs explicitly. If one can recursively divide the input space into more coherent clusters, the end result would be an implicit description of the context of the whole image. This is exactly what a random regression forest does.

   Unlike the previous instance, a single pass of the image is sufficient in order to localize multiple organs simultaneously. One may argue that humans localize (or detect) objects in a manner similar to this. But if humans use the context information, they also make use of the information related to the object to obtain their conclusions with a higher confidence.

[10] Unless otherwise specified all the CT images are presented with the same view setting (window width = 350 HU and window level = 40 HU) throughout the dissertation.

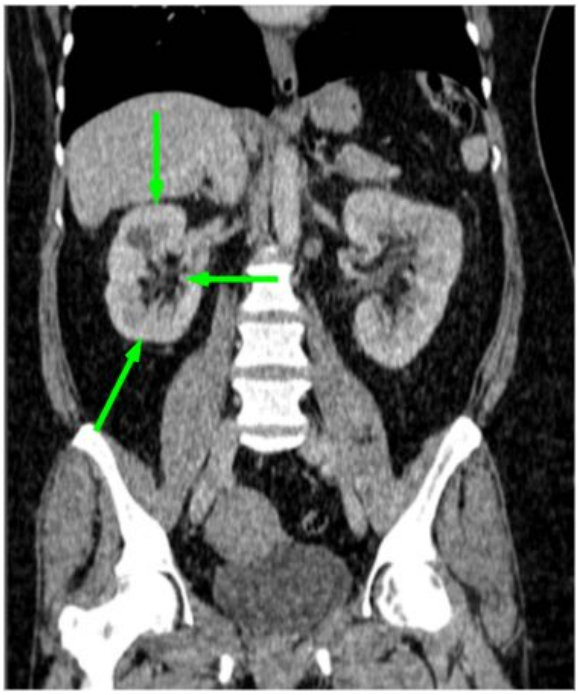


Figure 3.7: The right kidney is found below the liver, right to the spine and above the right pelvis. Hence, if one looks at below the liver, right of the spine and above the right pelvis, one should be able to localize the right kidney. The contrast of the image is adjusted for the best viewing purposes of the abdominal soft tissue organs (window width = 350 HU and window level = 40 HU [Johnson et al., 2007])[10].

3.4.2 Intrinsic Parameters

The number of regression trees and the maximum depth of a regression tree are the intrinsic parameters of a random regression forest. The extensive empirical study carried out by Oshiro et al. [2012] stated that there is no mechanism in figuring out the number of trees in a random forest. They also provided empirical evidence to demonstrate that adding more trees would not necessarily result in a forest with higher performance.

The standard practice is to fix a number of trees *a priori* at the start of the induction process. Bernard et al. [2009] proposed a method where they argued about the existence of a subset of tree from a bigger ensemble. After creating a random forest with a large number of decision trees, they proposed to add or subtract a tree from a set initially selected after an *a posteriori* evaluation of the performance of the forest. However, all the

studies mentioned in the Table 3.1 follow the standard procedure of setting the number of regression trees *a priori*. Concerning the maximum number of decision levels of a tree, the standard practice is to tune it depending on the application. The values of the intrinsic parameters of the main studies are presented in Table 3.2.

Study Num.	Num. of Trees	Max Decision Levels
(1)	12	7
(2)	6	8
(3)	7	15
(4)	4	12
(5)	7	12
(6)	3	14

Table 3.2: Intrinsic parameters of the studies presented in the Table 3.1.

Analyzing the values presented in Table 3.2 and considering that the studies are ordered from the oldest to the newest, the reduction of the number of trees and the increase of the maximum decision levels are clearly observed. It may suggest that the use of 3 to 5 random regression trees with maximum decision levels around 12 to 14 is the most popular choice for the multi–organ localization problem. This is probably due to the physical constraints imposed by the availability of Random Access Memory (RAM).

3.4.3 Random Regression Forest Ensemble

As described in Sect. 3.1.2, a RRF is an independent ensemble method. The ensemble method can be divided into four main parts (see Fig. 3.8). Namely, training set preparation phase, image preprocessing phase, training phase, and prediction phase.

In Sect. 3.4.4, we present how to localize organs manually and discuss the differences between localization and segmentation. Next, the image preprocessing steps commonly used in organ localization and their importance are briefly presented in Sect. 3.4.5. We describe the regression problem formulation using offset vectors, the features that capture the spatial context, how split nodes and leaf nodes are trained in Sect. 3.4.6. Finally, in Sect. 3.4.7, another analysis is carried out on the prediction phase of a RRF.

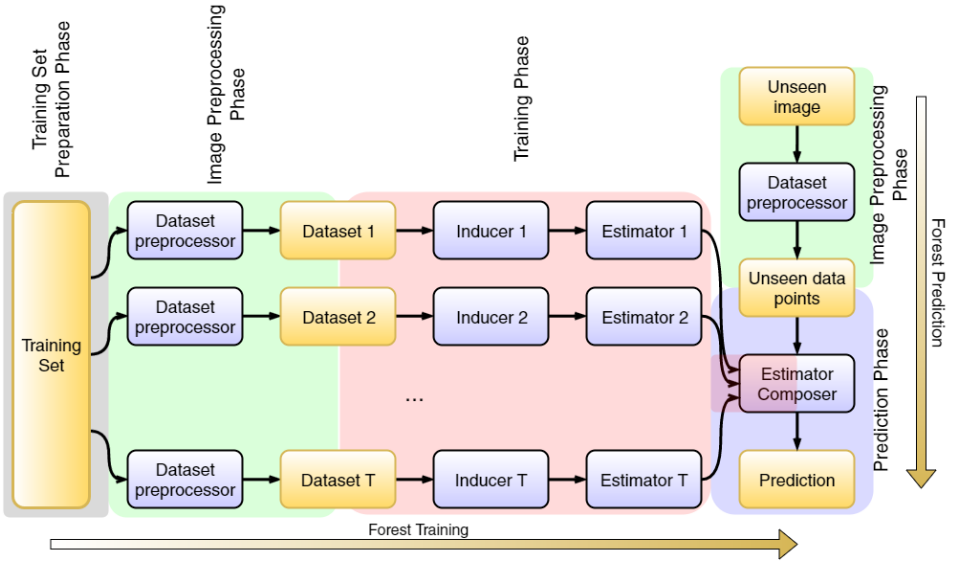


Figure 3.8: The RRF ensemble can be divided into the training set preparation phase, image preprocessing phase, training phase and prediction phase. Data components and algorithmic components are shown in yellow and blue respectively.

3.4.4 Training Set Preparation Phase

Although training set preparation phase is not to be found in the literature as a main phase of RRF ensemble method, we believe in its inclusion as a main step of the process.

Given a medical image, the **localization** of an organ can be achieved, for instance, by defining a bounding box tightly around that organ. In order to delineate a bounding box around an organ, one has only to find the two extreme points containing the organ. In our implementation of bounding box delineation using CamiTK, the user only has to click 6 points (the two extreme points in each direction) in order to mark the bounding box of an organ.

In order to manually **segment** an organ, the user has to label the regions belonging to the organ, usually by drawing all its boundaries (or contours). Even if semi–automatic guidance can be provided (*e.g.*, snakes or active contours), this is generally a very tedious and long process compared to delineating a bounding box and should be done by a trained expert. Unfortunately, expert time is rare and expensive. Hence, the ability to provide training data relatively easily and quickly for localization is a big advantage.

3.4.4.1 Bounding Box Vector Definition

The bounding box is denoted by an axis aligned 6 Dimensional (6D) displacement vector (in mm) to the 6 walls of the bounding box (right, left, anterior, posterior, inferior, and superior) of the organ (c) from the origin of the image (see Fig. 3.9). This vector is called the *bounding box* vector ($\mathbf{b}(c)$):

$$\mathbf{b}(c) = \left(\mathbf{b}_c^r,\ \mathbf{b}_c^l,\ \mathbf{b}_c^a,\ \mathbf{b}_c^p,\ \mathbf{b}_c^i,\ \mathbf{b}_c^s\right) \quad , \tag{3.6}$$

where c is the organ, r = right, l = left, a = anterior, p = posterior, i = inferior, and s = superior. These directions ($r, l, a, p, i,$ and s) correspond to how a radiologist interprets the image directions in synchronization with the human body (see Fig. 3.10).

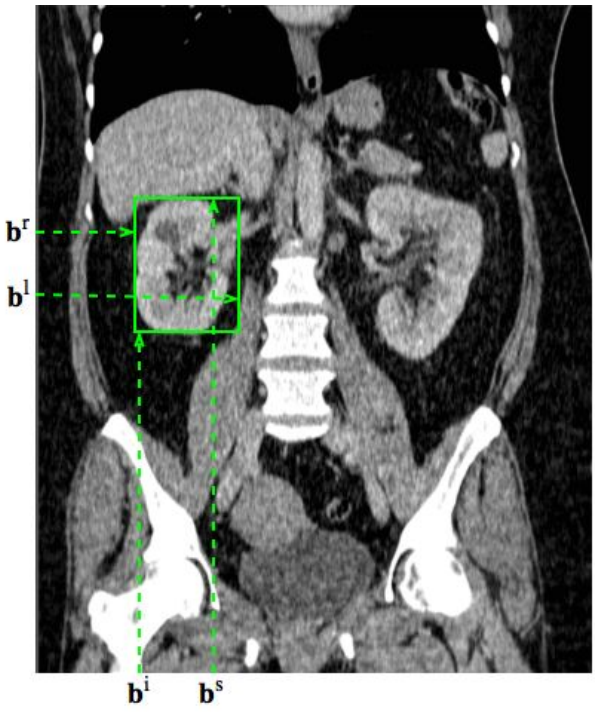


Figure 3.9: A coronal slice of a CT image and the 4 visible components of the bounding box vector of the right kidney. The bounding box vector is composed of 6 displacements from the origin of the image to each bounding box wall. (Since only one bounding box of an organ is shown in the figure the organ identifier (c) is omitted in the bounding box vector components for visibility. *e.g.,* $\mathbf{b}_c^l$ is shortened as $\mathbf{b}^l$.)[11]

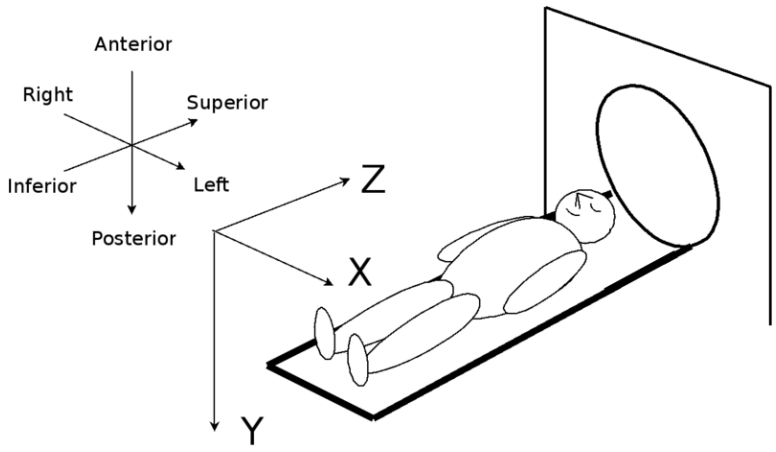


Figure 3.10: A radiologist observes the patient from the patient's feet to head. The image direction interpretation is based on this observation. The X, Y, and Z image axises progress from the right to left ($r \rightarrow l$), anterior to posterior ($a \rightarrow p$), and inferior to superior ($i \rightarrow s$) directions of the patient respectively. (Image retrieved from http://www.vtk.org/Wiki/File:DICOM-OrientationDiagram-LPS.png.)

3.4.5 Image Preprocessing Phase

Selection of image voxels that participate in the forest training phase as well as the prediction phase is the main responsibility of this phase. The use of a too small number of voxels will decrease the localization capabilities of the RRF algorithm as overfitting will occur. Although the use of too many voxels may not decrease the localization capabilities it will also affect the training and prediction times. In addition to that, when the offset vector distributions of leaf nodes (see Sect. 3.4.6) are saved to the disk, it would require a very large amount of disk space. Consequently, using too many voxels in RRFs impacts the usability aspects of the program. Hence, the choice of a *correct* number of voxels is an important criterion for the balance between the localization capabilities and usability measures of the method. The literature reports different ways of choosing the number of voxels: all voxels, a random subset of voxels, voxels on a regular grid, voxels belonging to a particular region, and voxels on a regular grid of a particular region (see Table 3.3).

[11] Unless otherwise specified, we have adopted the same notation in all the figures that follow.

In some studies, image preprocessing steps are carried out in order to reduce the noise of the images or carry out some normalization procedures. In the general context of multi–organ localization, the use of noise reduction methodologies may reduce the generality of the RRF capabilities. But in certain specific tasks (*e.g.*, the full segmentation of a specific organ) it may be advisable. The literature reports two main ways of preprocessing: down–sampling to reduce the number of voxels used and Gaussian filtering to smoothen the image (see Table 3.3).

Study	Voxels used in		Image Processing
	Training	Prediction	
(1)	all	all	none
(2)	all	all	none
(3)	n/a	n/a	n/a
(4)	in a regular grid of ±10 cm from center of each axial slice	in a regular grid of ±10 cm from center of each axial slice	down–sampling to 3 mm spacing per voxel
(5)	random subset of 40000 voxels	random subset	none
(6)	random subset of 30000 voxels	random subset of 30000 voxels	Gaussian smoothing

Table 3.3: Information on various voxel selection criteria and image preprocessing techniques used by the main studies. The studies are identified with the number assigned in the Table 3.1. (n/a: information not available)

3.4.6 Training Phase

As mentioned in Sect. 3.4.1, the solution proposed by random regression forests to the problem of multi–organ localization is based on the comprehension of the context. This context has to be independent of the size of the image as well as the origin of the image. Additionally, the problem should be formed as a regression problem.

If one considers a voxel of an image and the displacement from each bounding wall of the bounding box to the voxel, then, not only these displacements are independent of the size and origin of the image but also inherently possess a regression formation as a displacement is a continuous

entity. These displacements are only dependent on the position of the voxel and the bounding box of the organ.

3.4.6.1 Offset Vector Definition

The ingenious proposal of *offset vectors* ($\mathbf{d}$) first introduced by Criminisi et al. [2010] satisfied all these criteria and paved way for the utilization of random regression forests for multi–organ localization. Given a training voxel ($\mathbf{v}$) and an organ (c), the 6D displacement offset vector ($\mathbf{d}(\mathbf{v}, c)$) from the walls of the bounding box to the voxel is defined in the following manner (see Fig. 3.11).

$$\mathbf{d}(\mathbf{v}, c) = \hat{\mathbf{v}} - \mathbf{b}(c) \ , \tag{3.7}$$

where $\hat{\mathbf{v}} = (\mathbf{v}_x, \mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_y, \mathbf{v}_z, \mathbf{v}_z)$ made from the voxel position $(\mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_z)$ and $\mathbf{b}(c)$ as defined in Eq. (3.6) in page 64.

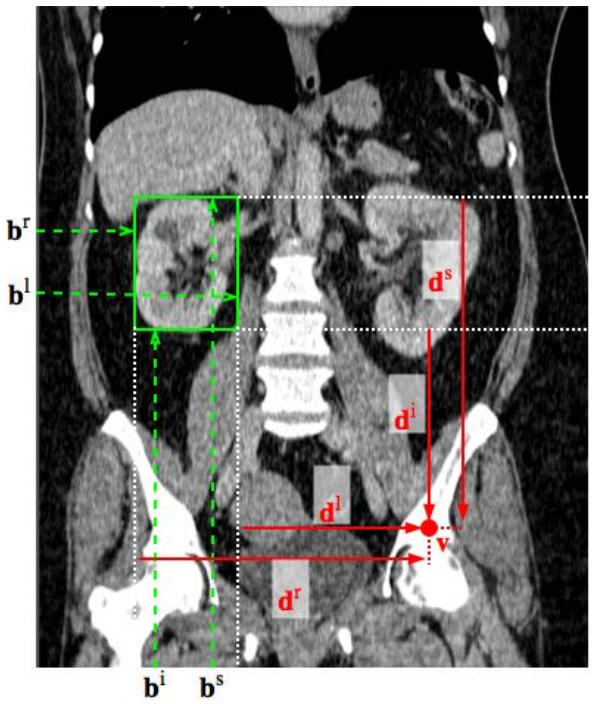


Figure 3.11: The offset vector is composed of 6 displacements from each bounding box wall to the voxel ($\mathbf{v}$). A coronal slice of a CT image and the 4 visible components of the offset vector (in red) of the right kidney of the voxel $\mathbf{v}$ and its bounding box vector (in green). The relationship of $\mathbf{d}(\mathbf{v}, c) = \hat{\mathbf{v}} - \mathbf{b}(c)$ (or $\hat{\mathbf{v}} = \mathbf{b}(c) + \mathbf{d}(\mathbf{v}, c)$) can clearly be observed.

Then, the multi–organ localization problem can be formulated as a multivariate regression problem that regresses the location of the bounding box walls of each organ given some set of image voxels.

As random regression forest is an ensemble of random regression trees, the next task is to induce the regression trees as mentioned in Sect. 2.3.2. This is done by:

- defining the features (Sect. 3.4.6.2),
- training the split nodes (Sect. 3.4.6.3),
- determining the best split among the many split configurations (Sect. 3.4.6.4), and
- training the leaf nodes (Sect. 3.4.6.5).

3.4.6.2 Feature Definition

The features used for the induction is of utmost importance as the selected features should be able to describe the context of the content but not the content per se.

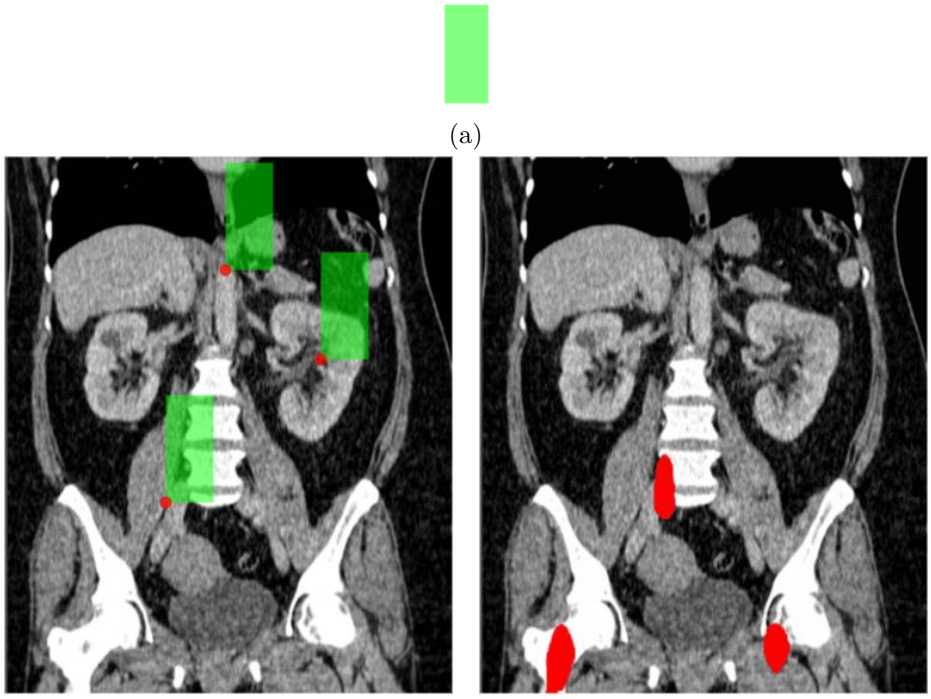
An image patch is a *local* portion of an image defined by either a small volume of a volumetric image or a small area of a 2D image. These patches can be used to describe the local portions of an image as is (*i.e.*, using the intensity values of the patch), or by transforming the patch into another entity (i.e., contours of the patch or any other image processing filter response). This is called a patch response. If the patches are of different sizes, the patch response is normalized with respect to the volume (or the area) of the patch. This facilitates the unbiased comparison of patch responses.

These patches can be used as features. But they would not capture the spatial context of the image but only capture the characteristics of the overlapping regions. Figure 3.12 provides a 2D example of using an image patch as a feature for regression tree induction. The 2D image patch used is presented in Fig. 3.12a. The feature response $\left(h\left(\mathbf{v},\theta\right)\right)$ is calculated as the mean intensity of the voxels overlapped by the patch (see 3.12b):

$$h\left(\mathbf{v},\theta\right) = \frac{1}{|\mathcal{V}|}\sum_{\mathbf{q}\in\mathcal{V}} I(\mathbf{q}) \ , \tag{3.8}$$

where $\mathbf{v}$ is the voxel, $\mathcal{V}$ is the volume of the patch, and $I(\mathbf{q})$ is the intensity

at the voxel $\mathbf{q}$[12]. Then, the feature response is subjected to the split function in order to find out whether the voxel goes to the left child or the right child (see Sect. 2.2.3.1 and the example in page 20). Figure 3.12c illustrates the binary separation of regions of the whole image when a high threshold value is used for the split function. The voxels that satisfy the test are shown in red.



(a) (b) (c)

Figure 3.12: A 2D example where an image patch is used as a feature. (a) The image patch. (b) Calculating the feature response at three voxels. (c) The binary separation using the a high threshold. The voxels that satisfy the test are shown in red. Since a high threshold is chosen, the high intensity regions that overlap the patch are separated from the rest.

Depending on the selected threshold, in this example we observe that regions belonging to bone structures are separated from the rest. This provides evidence that patches alone is useful at capturing the characteristics of image regions but not any information on the spatial context of those regions.

The use of displaced 2D patch differences as features (called Haar fea-

[12]The equations are presented in a 3D setting although the examples are in a 2D setting.

tures) for object detection has been reported in many studies [Oren et al., 1997; Papageorgiou et al., 1998; Viola and Jones, 2001]. But the displacement of one feature with respect to another was restricted as all features constructed a bigger enclosing rectangle. The innate ability of such features to describe the spatial context information was quickly understood by the scientific community. Consequently, the displacement restrictions that were present in the Haar features were dropped in the following studies [Gall and Lempitsky, 2009; Criminisi et al., 2009; Shotton et al., 2009; Criminisi et al., 2010, 2013]. And they defined the displaced binary patch features in the following manner:

$$h\left(\mathbf{v},\theta\right) = \frac{1}{|\mathcal{V}_1|}\sum_{\mathbf{q}\in\mathcal{V}_1} I(\mathbf{q}) - \frac{1}{|\mathcal{V}_2|}\sum_{\mathbf{q}\in\mathcal{V}_2} I(\mathbf{q}) \ , \tag{3.9}$$

where $\mathbf{v}$ is the voxel, $\mathcal{V}$ is the volume of the patch, and $I(\mathbf{q})$ is the intensity at the voxel $\mathbf{q}$.

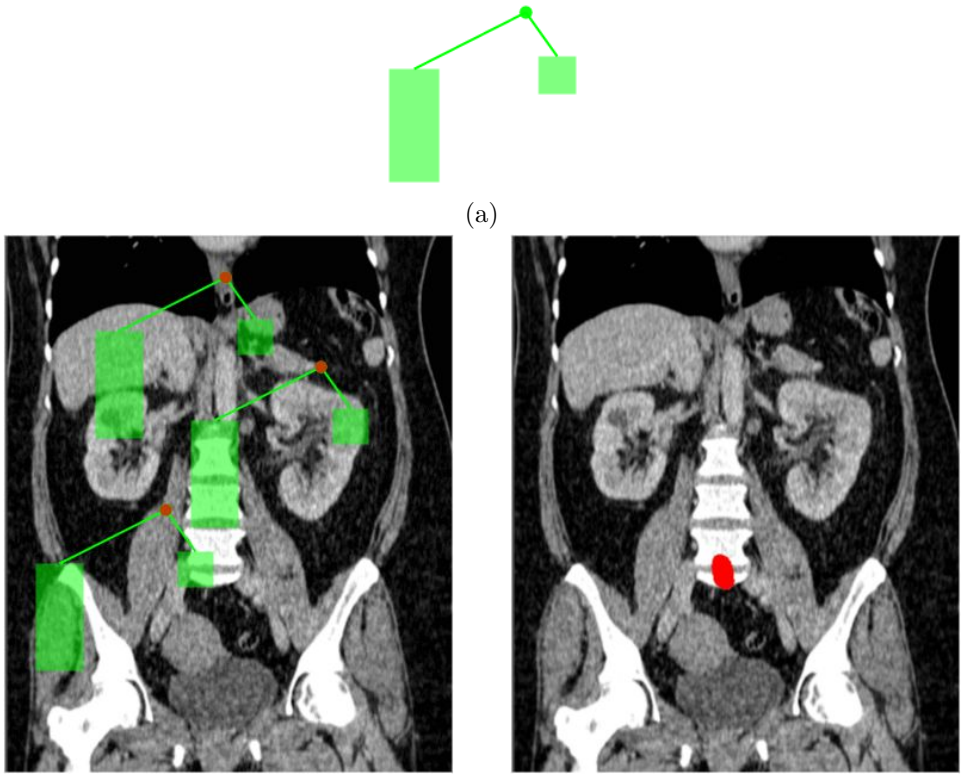
A 2D example is provided in Fig. 3.13 demonstrating the power of displaced binary patches in describing the spatial context. The used displaced binary image patches are presented in Fig. 3.13a. The feature response is calculated as mentioned in Eq. (3.9) by replacing the volume ($\mathcal{V}$) by the area ($\mathcal{A}$). Finally, Fig. 3.13c illustrates the binary separation of regions when a high threshold value is used for the split function.

The feature selects the voxels when the first patch overlaps a high intensity region while the second patch overlaps a low intensity region for a high threshold value of the split function. Considering the anatomy of the body and the given 2D example, this means the feature isolates the structure of the bottom of the spine as shown by the red voxels in Fig. 3.13c. Furthermore, we observe that the selected spine region is a good landmark to predict the location of the right femur head. The advantage of the displaced image patch differences in capturing spatial context over simple image patches is clearly visible.

Displaced unary image patches too are capable of retaining spatial context information. A displaced unary image patch is defined in the following manner:

$$h\left(\mathbf{v},\theta\right) = \frac{1}{|\mathcal{V}|}\sum_{\mathbf{q}\in\mathcal{V}} I(\mathbf{q}) \ , \tag{3.10}$$

where $\mathbf{v}$ is the voxel, $\mathcal{V}$ is the volume of the displaced patch, and $I(\mathbf{q})$ is the



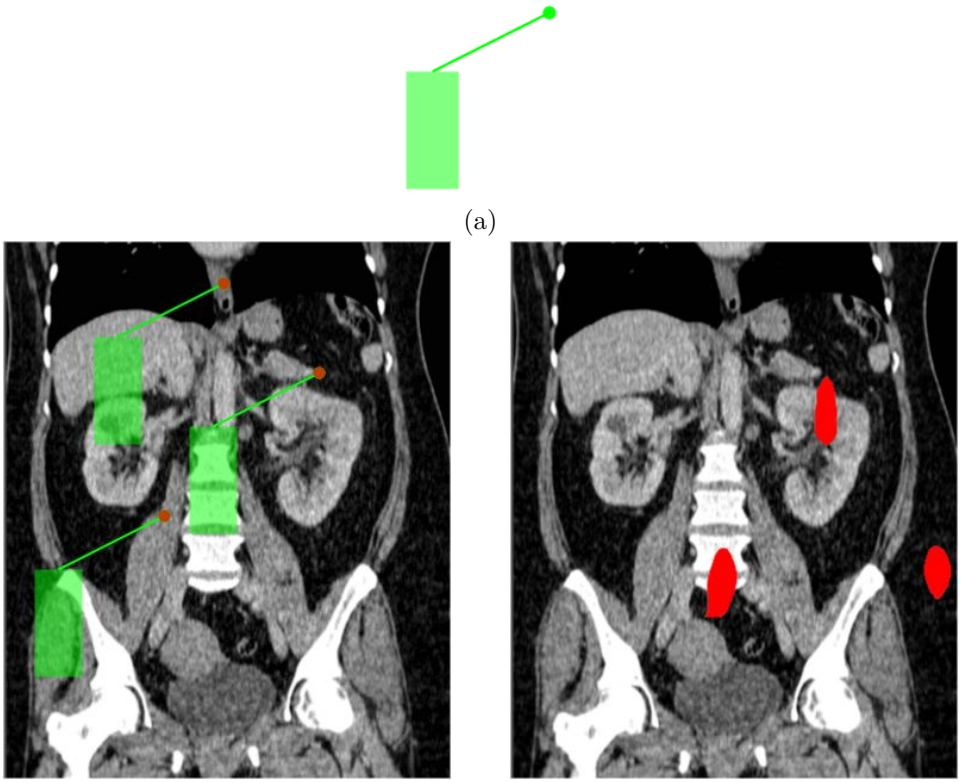
(a) (b) (c)

Figure 3.13: A 2D example where displaced binary image patches are used as a feature. (a) The two image patches displaced with respect to some origin. (b) Calculating the feature response at three voxels. (c) Resulting binary separation using a high threshold. The voxels that satisfy the test are shown in red. Due to the spatial arrangement of the feature, it selects the voxels when the first patch overlaps a high intensity region while the second patch overlaps a low intensity region. Considering the anatomy of the body and the given 2D example, this means the feature isolates the structure of the bottom of the spine as shown by the red voxels in (c).

intensity at the voxel $\mathbf{q}$. Although Eq. (3.8) and Eq. (3.10) appear the same, the concepts are different as the patch is displaced in Eq. (3.10) whereas it is not in Eq. (3.8).

An example is given in Fig. 3.14 demonstrating the ability of displaced unary patches to describe the spatial context. The displaced unary image patch used is shown in Fig. 3.14a. The binary separation of regions depending on the selected threshold value is presented in Fig. 3.14c.

Analyzing Fig. 3.14c, we observe that certain parts of the spine, fatty



(a) (b) (c)

Figure 3.14: A 2D example of a displaced unary image patch used as a feature. (a) A displaced image patch with respect to some origin. (b) Calculating the feature response at three voxels. (c) Resulting binary separation using a high threshold. The voxels that satisfy the test are shown in red.

region left to the left hip, as well as some regions near the left kidney are categorized to one region. The region near the spine is capable of localizing the right femur head region whereas the region near the left kidney locates the spine. Although not very pertinent, the region left to the left hip is capable of locating the left femur head. Although not as precise as the displaced binary image patch features, the displaced unary image patches are also capable of retaining spatial context information.

All the studies presented in Table 3.1 employ displaced unary image patches, more specifically they used the mean intensity of cuboidal volumes as the features except for the study of Pauly et al. [2011]. Pauly et al. employed local binary pattern features [Ojala et al., 1996]. The local binary patterns are intensity and scale invariant type of features that capture the

textural information. The features used for the main studies are presented in column 2 of the Table 3.4 in page 76.

The split node training consists of the creation of many split configurations and subjecting all the incoming voxels to the split function to determine whether the voxels should go to the left child or the right child (see Sect. 3.4.6.3). Finally, the best split configuration is selected (see Sect. 3.4.6.4).

3.4.6.3 Split Node Training: Generating Split Configurations

In order to train a split node, first many split configurations are made (see Fig. 3.18). Each split configuration consists of a randomly selected feature (θ) and one or two threshold values. If it comprises only one threshold value (τ), then the feature response at the voxel $\mathbf{v}$ is compared to the threshold in the following manner:

$$h(\mathbf{v}, \theta) < \tau \ . \tag{3.11}$$

If the split configuration comprises two threshold values (τ_1 and τ_2), then the feature response is tested against a value range as shown below:

$$\tau_1 < h(\mathbf{v}, \theta) < \tau_2 \ . \tag{3.12}$$

All voxels that arrive at the split node will be forwarded to all split configurations and the voxels will either be sent to the left child (if within thresholds) or the right child (if not within thresholds) of the configurations depending on the result of feature response comparison to the thresholds. Once, all voxels of all training data that arrive at the split node are treated, the best split configuration with the highest impurity reduction of the data has to be selected. This process is explained next.

3.4.6.4 Split Node Training: Best Split Selection

Another important decision of regression tree induction is the criterion used for the selection of the best split among all split configurations. The criteria based on information gain or squared error loss have been used in the literature (see Table 3.4). Except for one study [Cuingnet et al., 2012], all other 5 studies have used maximum information gain criterion to select the best

split among the candidates. Cuingnet et al. [2012] have used the squared error loss as their best split selection criterion.

Since the discussion is focused on binary random regression forests as mentioned at the beginning of this section, the information gain calculation is only presented for binary splits. Let us consider a split node where the set of incoming voxels to the node ($\mathcal{X}$) are divided into the left child ($\mathcal{X}_L$) and the right child ($\mathcal{X}_R$) (see Fig. 3.15). And the regression is formulated on the joint distribution of offset vectors ($p(\mathbf{d}, c; \mathcal{X})$) given an organ class c. Then, the information gain (IG) is defined in the following manner:

$$IG = H(\mathbf{d}, c; \mathcal{X}) - \sum_{i \in \{L,R\}} \omega_i H(\mathbf{d}, c; \mathcal{X}_i) \ , \tag{3.13}$$

where $H(\cdot)$ is the entropy and $\omega_i = \frac{|\mathcal{X}_i|}{|\mathcal{X}|}$ where $|\mathcal{X}|$ is the number of voxels.

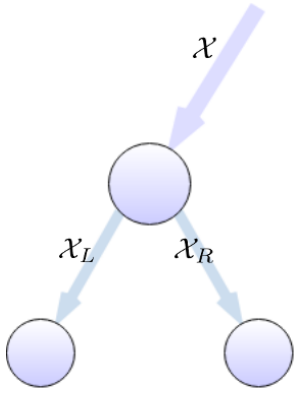


Figure 3.15: Binary split at some split node. Number of voxels at the node, left child, and right child are $|\mathcal{X}|$, $|\mathcal{X}_L|$, and $|\mathcal{X}_R|$ respectively.

The joint distribution ($p(\mathbf{d}, c; \mathcal{X})$) of offset vectors ($\mathbf{d}(\mathbf{v}, c)$) for a given organ class c is defined in the following manner:

$$p(\mathbf{d}, c; \mathcal{X}) = p(\mathbf{d} \mid c; \mathcal{X}) p(c; \mathcal{X}) \ , \tag{3.14}$$

where $p(\mathbf{d} \mid c; \mathcal{X})$ is the conditional distribution of $\mathbf{d}(\mathbf{v}, c)$ and $p(c; \mathcal{X})$ is the organ class prior probability distribution. The conditional distribution of offset vectors is defined as a multivariate Gaussian:

$$p(\mathbf{d} \mid c; \mathcal{X}) = \frac{1}{(2\pi)^{\frac{N}{2}} |\Lambda_c(\mathcal{X})|^{\frac{1}{2}}} \exp^{-\frac{1}{2}(\mathbf{d}_c - \bar{\mathbf{d}}_c)^\intercal \Lambda_c(\mathcal{X})^{-1} (\mathbf{d}_c - \bar{\mathbf{d}}_c)} \ , \tag{3.15}$$

where $N = 6$ is the dimensionality of the offset vector, $\bar{\mathbf{d}}_c$ and $\Lambda_c(\mathcal{X})$ are respectively the mean vector and covariance matrix of offset vectors of organ class c, and $\mathbf{d}_c$ is the shorten form of $\mathbf{d}(\mathbf{v}, c)$ to avoid clutter. And $\int_{\mathbb{R}^6} p(\mathbf{d}, c; \mathcal{X})\, \mathrm{d}\mathbf{d} = 1$ holds. Voxels from many training images may arrive at the given split node. In certain images, certain organs may not be fully present. In such cases, the offset vectors can not be calculated for the voxels coming from those images. Hence, the discrete organ class prior is calculated using the number of voxels for which the offset vector of the organ can be calculated at the split node. That is:

$$p(c; \mathcal{X}) = \frac{n_c(\mathcal{X})}{Z} \ , \tag{3.16}$$

where $n_c(\mathcal{X})$ is the number of voxels for which the offset vectors can be calculated, $Z = \sum_{c \in \mathcal{C}} n_c(\mathcal{X})$ is a normalizing constant such that $\sum_{c \in \mathcal{C}} p(c; \mathcal{X}) = 1$, and $\mathcal{C}$ is the set of all organs to be localized. Given a generic Gaussian distribution of a random variable $\mathbf{x} \in \mathbb{R}^N$ with a covariance matrix Λ, the differential entropy can be defined in the following manner [Ahmed and Gokhale, 1989]:

$$H(\mathbf{x}) = \frac{1}{2} \ln \left((2\pi e)^N \left| \Lambda \left(\mathbf{x} \right) \right| \right) \ . \tag{3.17}$$

Then using Eq. (3.14), Eq. (3.17) for the joint distribution can be rewritten in the following manner:

$$H(\mathbf{d}, c; \mathcal{X}) = H(c; \mathcal{X}) + \sum_{c \in \mathcal{C}} p(c; \mathcal{X}) \left(\frac{1}{2} \ln (2\pi e)^6 |\Lambda_c(\mathcal{X})| \right) \ . \tag{3.18}$$

Then substituting Eq. (3.18) in Eq. (3.13) we arrive at the final information gain formulation:

$$\begin{aligned} IG = & H(c; \mathcal{X}) - \sum_{i \in \{L,R\}} \omega_i H(c; \mathcal{X}_i) \ + \frac{1}{2} \sum_{c \in \mathcal{C}} p(c; \mathcal{X}) \left(\ln (2\pi e)^6 |\Lambda_c(\mathcal{X})| \right) \\ & - \frac{1}{2} \sum_{i \in \{L,R\}} \omega_i \sum_{c \in \mathcal{C}} p(c; \mathcal{X}_i) \ \left(\ln (2\pi e)^6 |\Lambda_c(\mathcal{X}_i)| \right) \ . \end{aligned} \tag{3.19}$$

Once the best split configuration that produces the maximum information gain is selected among all available split configurations, the training of the split node comes to an end. The feature and the threshold values

attached with the selected best split configuration are saved as the feature and threshold values of the trained split node. Once split nodes are trained, the next step is to train the leaf nodes.

The best split selection criteria used by the main studies are presented in column 3 of the Table 3.4.

Study	Features Used	Best Split Selection
(1)	Mean intensity of displaced cuboids	Max information gain
(2)	Local Binary Patterns	Max information gain
(3)	Mean intensity of displaced cuboids	Min squared loss error
(4)	Mean intensity of displaced cuboids	Max information gain
(5)	Mean intensity of displaced cuboids	Max information gain
(6)	Mean intensity of displaced cuboids	Max information gain

Table 3.4: The features and the best split selection criteria used by the studies mentioned in the Table 3.1.

3.4.6.5 Leaf Node Training

When all the split nodes are trained, the leaf nodes contain all the accumulated voxels. Training of a given leaf node consists of summarizing the offset vectors of these accumulated voxels. First, the mean vector ($\bar{\mathbf{d}}_c$) and covariance matrix (Λ_c) of the accumulated voxels' offset vectors are saved.

The studies mentioned in the Table Table 3.1 make the simplifying assumption that bounding box wall positions are uncorrelated. Consequently, that would make all the non-diagonal entries of the covariance matrix equal to zero and only the pure variance values would be non-zero. With this spatial independence assumption, the distribution of the accumulated offset vectors along each wall direction is saved for all organs as normalized histograms $\left(p\Big(\mathbf{d}^{\text{dir}} \mid c; \mathcal{X}\Big)\right.$ where $\text{dir} \in \{r, l, a, p, i, s\}$) and $\mathbf{d}^{\text{dir}} \in \{\mathbf{d}^r, \mathbf{d}^l, \mathbf{d}^a, \mathbf{d}^p, \mathbf{d}^i, \mathbf{d}^s\}$).

3.4.7 Prediction Phase

This is the phase where a trained random regression forest is used to predict the localization of multiple organs in a CT image. Though this phase is commonly known as the *testing phase* in the machine learning community, it is our belief that prediction or estimation phase is a better suited term as

the ultimate motive of the procedure is to predict or estimate the localization of multiple organs.

The general flow of the prediction phase is as follows. First and foremost, any image preprocessing steps required (as discussed in Sect. 3.4.5) are performed. The ultimate goal of these preprocessing steps is to find the final set of voxels (data points) that would be used for prediction. Once the final set of voxels is found, each voxel is pushed through each regression tree of the forest. Each voxel will be subjected to many split functions until accumulated at a leaf node ultimately. The path to the leaf node is determined by the outcome of the split functions. Once, all the selected voxels are accumulated at the leaf nodes, the number of leaf nodes that participate in prediction is determined. Then, the leaf nodes belonging to each regression tree makes a prediction of the localization of each organ using some estimation model. Finally, each prediction from each regression tree is composed in some manner to get the final ensemble prediction.

3.4.7.1 Number of Leaf Nodes Used For Prediction

Usually, the voxels of the unseen image (*i.e.*, the images that were not used during the training phase) that were pushed through a random regression forest do not all participate in the actual prediction (see column 3 of Table 3.5 in page 80). Hence, determining how many leaf nodes participate in the actual prediction ($\mathcal{L}$) is an important decision of the prediction phase.

The argument for such a selection is the fact that the displaced cuboid features have limitation of their spatial range. For example, the good latent landmarks selected automatically (the greater trochanter) by the RRF algorithm for the left femur head might not be a good landmark for the liver as the two organs are quite far apart (see Fig. 3.16). Consequently, leaf nodes that correspond to *good* landmarks for a given organ have to be selected.

To select the leaf nodes used for prediction, one such a way adopted by the researchers is to look at the determinant of the covariance matrix of the offset vectors saved at the training phase since a higher determinant would imply bigger variation of the offset vectors accumulated at the leaf node. And bigger variation of offset vectors would in turn imply dispersed voxel distribution suggesting that the set of voxels that belongs to that leaf node may not correspond to a *good* landmark. In contrast, a leaf node with a low determinant would indicate more confident prediction.

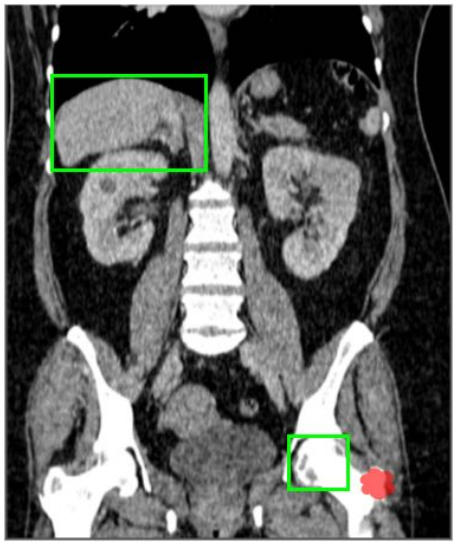


Figure 3.16: The region around the greater trochanter (the red patch) is a good landmark for the left femur head. But the same region as a landmark for the liver may not result in a good prediction as the organ and the landmark are far apart. The displaced cuboid features used for the forest may not be able to catpure such long range context information.

Hence, in order to choose the *best* leaf nodes for predicting the location of an organ, many studies sort the leaf nodes according to the determinant of the covariance matrix of the offset vectors. Then, they use the leaf nodes with the highest confidence that contain a cumulated percentage of voxels that were pushed through the forest ($\mathbf{v}_s\%$). This voxel percentage is set *a priori*. The different values used by different studies are found in column 3 of Table 3.5 in page 80.

Analyzing the percentage values used in different studies, we observe that the tendency is to use a small number of leaf nodes for the final prediction. Consequently, this implies using a small number of leaf nodes having the highest confidence for the organ localization prediction. Additionally, the use of a fewer number of voxels may result in faster prediction run times.

3.4.7.2 Estimation and Estimation Composer Models

All the main studies mentioned previously in Table 3.1 use the same estimation model of conditional posterior distribution of offset vectors to obtain the localization prediction of each regression tree.

Each voxel $\mathbf{v}$ of all the voxels of the unseen image that were pushed through the forest reaches one leaf node per regression tree ($l(\mathbf{v})$). The leaf nodes already have the conditional offset vector distribution $\left(p\left(\mathbf{d} \mid c; \mathcal{X}\right)\right)$ information stored in them from the training phase as the mean offset vector $\left(\bar{\mathbf{d}}_c\right)$, covariance matrix of offset vectors (Λ_c) and 1D distribution of the accumulated offset vectors along each wall direction $\left(p\left(\mathbf{d}^{\text{dir}} \mid c; \mathcal{X}\right)\right)$ (see Sect. 3.4.6.5). From the mean offset vector, we could obtain the mean bounding box vector in the following manner:

$$\bar{\mathbf{b}}_c(\mathbf{v}) = \hat{\mathbf{v}} - \bar{\mathbf{d}}(\mathbf{v}; c) \ . \tag{3.20}$$

Hence, the conditional distribution of bounding box vector $\left(p(\mathbf{b}_c \mid l)\right)$ can be known. If the set of leaf nodes that participate in the localization prediction of organ c is $\mathcal{L}$ (selected in the manner described in Sect. 3.4.7.1), then the posterior probability for $p(\mathbf{b}_c)$ can be obtained in the following fashion:

$$p(\mathbf{b}_c) = \sum_{l \in \mathcal{L}} p(\mathbf{b}_c \mid l) p(l) \ , \tag{3.21}$$

where $p(l(\mathbf{v})) = |l(\mathbf{v})| / \sum_{l \in \mathcal{L}} |l(\mathbf{v})|$ and $|l(\mathbf{v})|$ is the number of voxels accumulated at the leaf node $l(\mathbf{v})$. And each organ would have a different $\mathcal{L}$ selected.

In order to compose the ensemble estimation of the random regression forest out of the individual estimations of random regression trees, all the studies mentioned in the Table 3.1 use averaging. Hence, the composed ensemble estimation is:

$$p(\mathbf{b}_c) = \frac{1}{T} \sum_{t=1}^{T} \sum_{l \in \mathcal{L}} p(\mathbf{b}_c \mid l) p(l) \ , \tag{3.22}$$

where T is the number of trees in the forest. If the bounding box walls are assumed to be uncorrelated, then, this posterior distribution can be represented using 1D histogram per bounding wall direction per organ. Hence, a single organ would possess 6 1D histograms.

Now that the posterior distribution of bounding box vectors per organ is available, all that is left to do is to select the final prediction values. Most retained concept appears to be the Maximum *A–Posteriori* (MAP) strategy. With MAP, the absolute position of a bounding box of an organ

$\left(\hat{\mathbf{b}}_c\right)$ is defined in the following manner.

$$\hat{\mathbf{b}}_c = \arg\max_{\mathbf{b}_c} \; p(\mathbf{b}_c) \;\; . \tag{3.23}$$

If predictions are represented using 1D histograms per bounding wall direction, then simply finding the mode of the histogram is equivalent to finding the MAP.

The early studies used the mathematical expectation to find the absolute position of the bounding walls in the following manner [Criminisi et al., 2010; Cuingnet et al., 2012]:

$$\hat{\mathbf{b}}_c = -\int_{\mathbf{b}_c} \mathbf{b}_c p\left(\mathbf{b}_c\right) \, d\mathbf{b}_c \;\; . \tag{3.24}$$

The concepts and parameter settings of the prediction phase of the main studies are presented in column 4 and 5 of Table 3.5.

Study	Voxels pushed through forest	$\mathbf{v}_s$%	Estimation composition	Final prediction
(1)	all	1%	Average	Mathematical expectation
(2)	all	n/a	Average	Arg Max
(3)	n/a	n/a	Average	Mathematical expectation
(4)	in a regular grid of ±10 cm from center of each axial slice	75%	Average	Arg Max
(5)	random subset	n/a	Average	Arg Max
(6)	random subset of 30000 voxels	3%	Average	Arg Max

Table 3.5: The amount of voxels pushed through the forest at the prediction phase, the final percentage of voxels used for prediction ($\mathbf{v}_s$%), the estimation composition, and the final prediction strategy employed by the studies mentioned in the Table 3.1.

In the next section, we present a digest of the main algorithms involved in the multi–organ localization process using RRFs.

3.5 A Digest of The Random Regression Forest Process

This final section represents a quick digest of the process of multi–organ localization using Random Regression Forest. We emphasize on the steps of the process rather than any analysis or mathematical modeling. The idea of this section is to summarize the algorithms used for training a RRF (Sect. 3.5.1) and the algorithms used for prediction (Sect. 3.5.2).

3.5.1 Process of Forest Training

The process of the RRF training is presented as an enumerated set of steps in order to be easily referred in the incoming chapters. Additionally, a set of figures are provided that take a 2D toy example and walk through the steps (see Fig. 3.17 to Fig. 3.19). There are 6 main steps:

Step 1–Image preprocessing: first, the voxels of each training image that will be used in the training phase are selected (see Fig. 3.17a). The use of all voxels of the image or a random subsample of voxels or voxels belonging to some Region of Interest (ROI) are the available options in the literature. For further details see Sect. 3.4.5.

Step 2–Forest creation: then, a Random Regression Forest is created with the intrinsic parameters. The number of Random Regression Trees (RRTs) and the maximum decision levels of a RRT are the intrinsic parameters (see Fig. 3.17b). For further details see Sect. 3.4.2.

Step 3–Split configuration generation: next, starting from the root node, each split node is trained in the following manner. Many split configurations (see Fig. 3.18c) are made for each split node using the randomly chosen features (see Fig. 3.18a) and thresholds (see Fig. 3.18b). The features used in the study are the mean intensity of displaced cuboids. The feature response of each voxel is compared to the thresholds of each split configuration and depending on the comparison output, the voxels are sent to either the left or the right child of the split configuration. For further details see Sect. 3.4.6.2 and Sect. 3.4.6.3.

Step 4–Best split selection: once all the training voxels have accumulated at a particular split node, the best split configuration is selected

from all available split configurations using an information gain based measure calculated using the offset vectors. The feature and thresholds belonging to the selected split configuration are saved as the feature and thresholds of the trained split node (see Fig. 3.19). For further details see Sect. 3.4.6.4.

Step 5–Split node training termination: the split node training is stopped when one of the following conditions are met. 1.) When the maximum decision levels of a RRT is reached. 2.) When the number of voxels accumulated at a node is less than some threshold ($n_{\min}$). 3.) When no information gain is achieved for all split configurations. For further details see Sect. 2.3.2 page 27.

Step 6–Leaf node training: Leaf node training consists of storing the mean vector ($\bar{\mathbf{d}}_c(\mathbf{v})$) and covariance matrix (Λ_c) of offset vectors of accumulated voxels ($\mathbf{v}$) at the leaf node ($l(\mathbf{v})$) for each organ c (see Fig. 3.19). Additionally, the distribution of the accumulated offset vectors along each wall direction is saved for all organs as normalized histograms $\left(p\left(\mathbf{d}_c^{\text{dir}} \mid l(\mathbf{v})\right)\right)$ where dir $\in \{r, l, a, p, i, s\}$ and $\mathbf{d}_c^{\text{dir}} \in \{\mathbf{d}_c^r, \mathbf{d}_c^l, \mathbf{d}_c^a, \mathbf{d}_c^p, \mathbf{d}_c^i, \mathbf{d}_c^s\}$. The abbreviations r, l, a, p, i, and s stand for right, left, anterior, posterior, inferior, and superior directions respectively. For further details see Sect. 3.4.6.5.

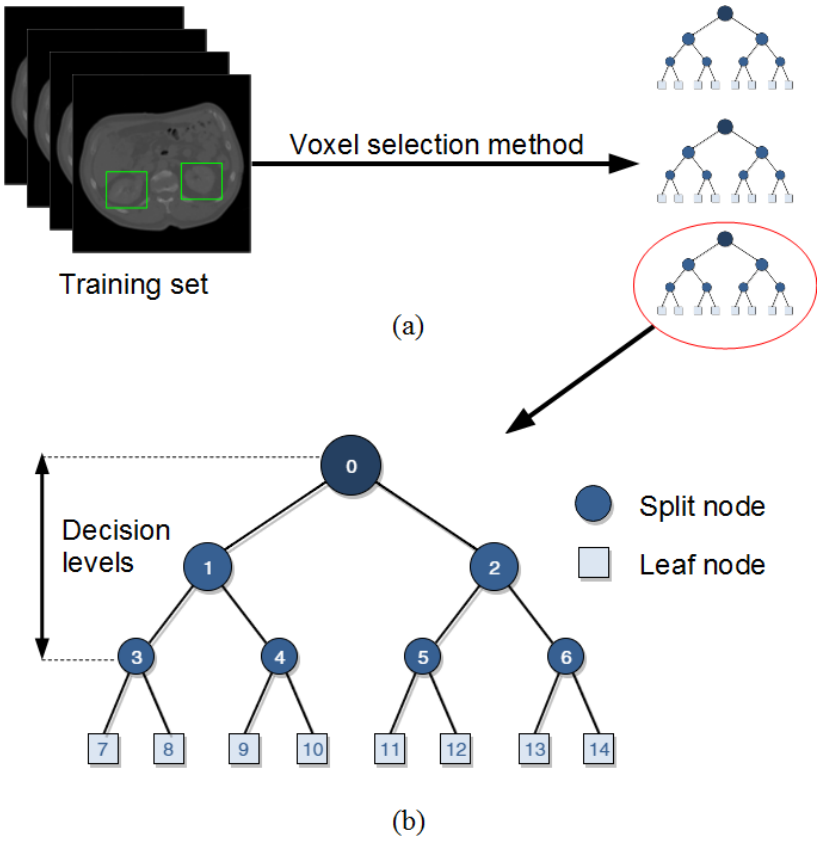


Figure 3.17: The training steps 1 and 2. (a) Training image voxel selection. (b) Forest creation.

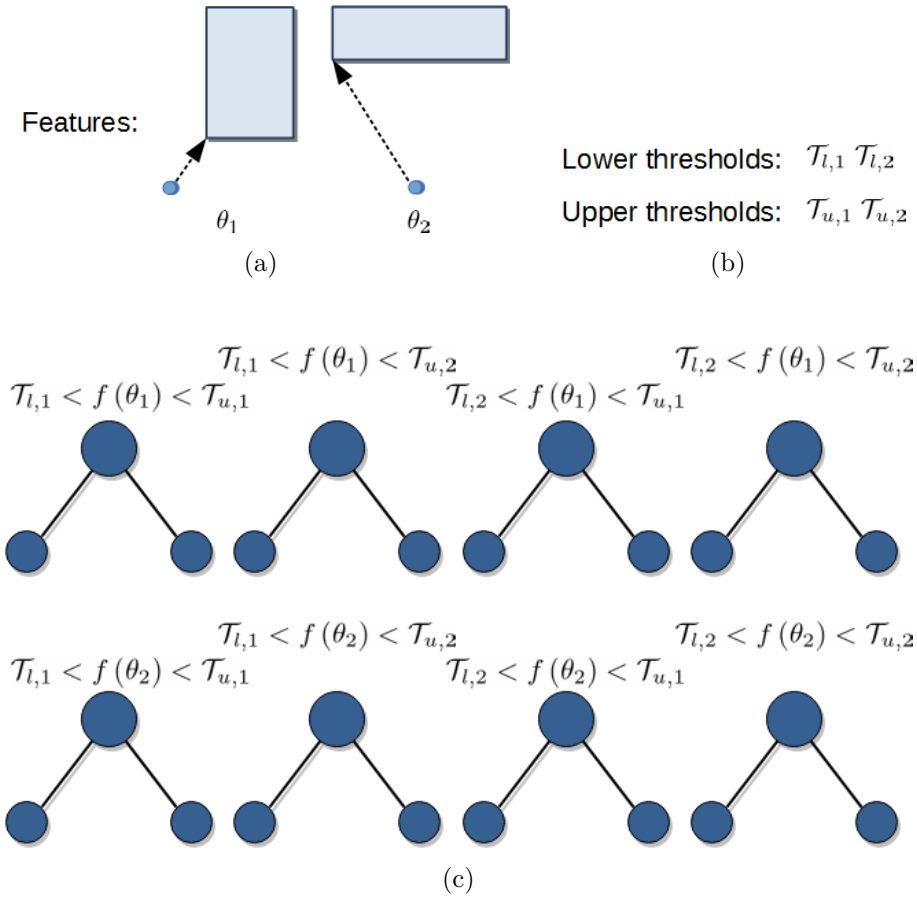


Figure 3.18: The training step 3. (a) Two features used to build the split configurations. (b) Two lower thresholds and two upper thresholds used to build the split configurations. (c) The 8 different split configurations created.

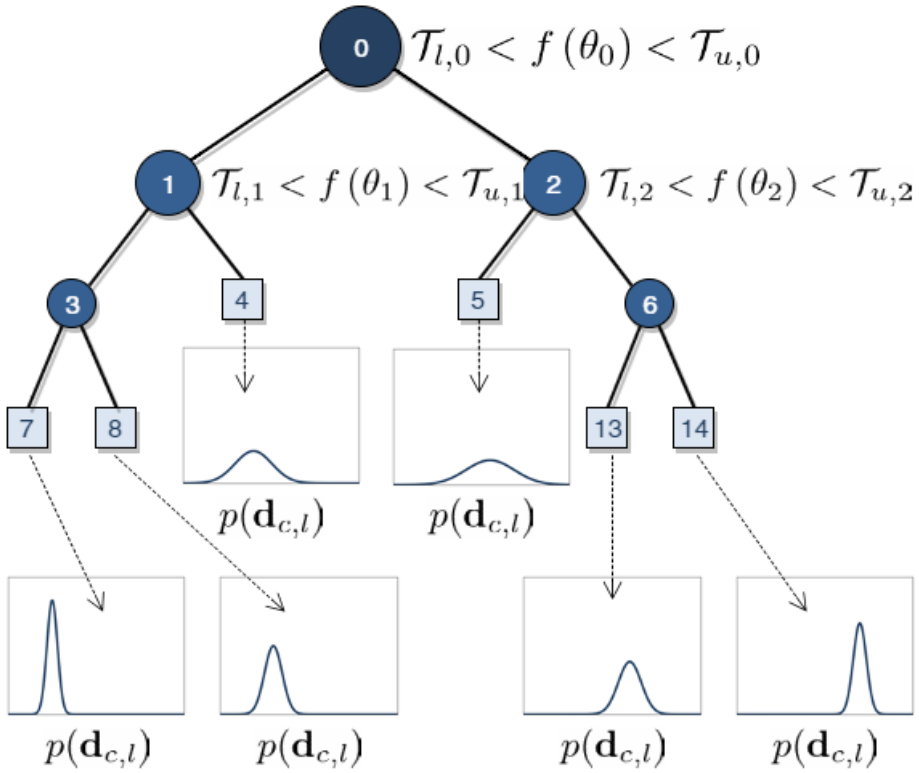


Figure 3.19: The end result of training step 4, 5, and 6. A trained RRF comprises split nodes with selected features and thresholds and leaf nodes with mean vector, covariance matrix of offset vectors, and 1D histograms of offset vector distribution along each direction. Notice that some of the split nodes from Fig. 3.17b have become leaf nodes.

3.5.2 Process of Forest Prediction

Similarly to the training phase, we present the process of the RRF prediction as an enumerated set of steps in order to be easily referred in the incoming chapters. The same 2D example that was used in the previous section is used to highlight the main steps mentioned below (see Fig. 3.20).

Step 1–Image preprocessing: similarly to the first step of the training phase, the voxels of the testing image that will be used in the prediction phase are selected. The same selection criterion used in the training phase is used in the prediction phase too. For instance, the algorithms

can use all voxels of the image or a random subsample of voxels or voxels belonging to some ROI (see Fig. 3.17a). For further details see Sect. 3.4.5.

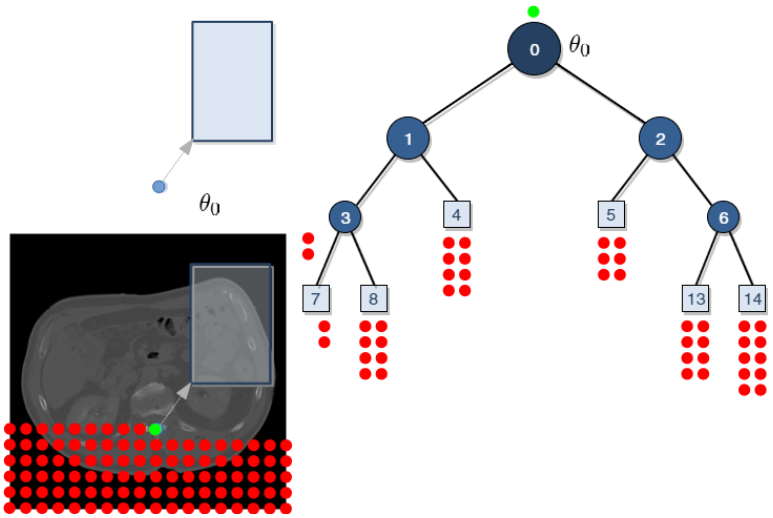
Step 2–Pushing voxels through the forest: then each selected voxel is pushed through each RRT of the RRF starting at the root node of each RRT. At every split node, the feature response is calculated using each voxel and the feature of the split node. Depending on the comparison of the feature response to the thresholds of the split node, the voxel is either sent to the left child or the right child as illustrated in Fig. 3.20. This process is repeated until the voxel is accumulated at some leaf node per RRT. For further details see the example in page 20.

Step 3–Leaf node selection for prediction: once all the selected voxels are accumulated at the leaf nodes across the whole forest, a fraction of these leaf nodes are selected to participate in the final localization prediction. Per organ, the leaf nodes are sorted according to the covariance matrix (Λ_c) of offset vectors saved at the training phase. Some of the leaf nodes at the head of sorted queue ($\mathcal{L}$) are selected to participate in the final localization prediction (see Fig. 3.21b). For further details see Sect. 3.4.7.1.

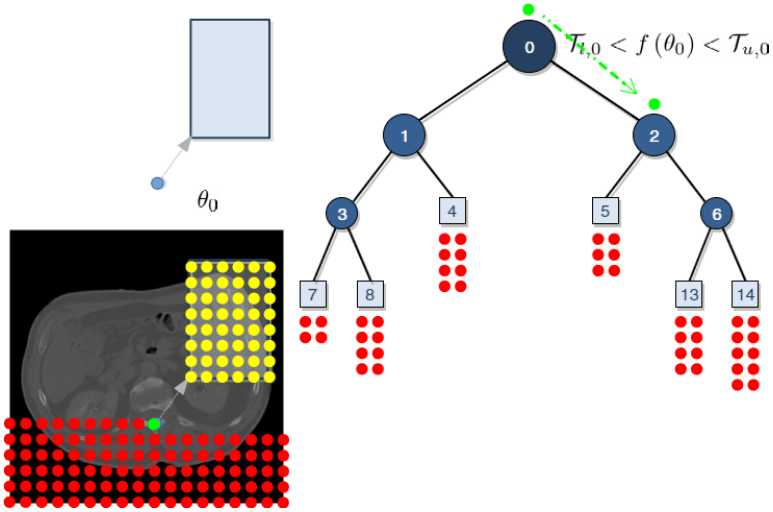
Step 4–Localization prediction by a single RRT: each leaf node that belongs to $\mathcal{L}$ makes a contribution to the conditional distribution of a bounding wall of an organ. The distribution of offset vectors that are available for each leaf node from the training phase are used for this (see Fig. 3.21b). These contributions are accumulated in a histogram per bounding wall direction per organ (see Fig. 3.21c). For further details see Sect. 3.4.7.2.

Step 5–Ensemble prediction combination: the bounding wall location distribution of each RRT is added to the same corresponding histogram to make the ensemble estimator composer. For further details see Sect. 3.4.7.2.

Step 6–Absolute localization derivation: ultimately, the absolute position of each bounding wall is obtained by finding the mode of the corresponding histogram (see Fig. 3.21c). For further details see Sect. 3.4.7.2.

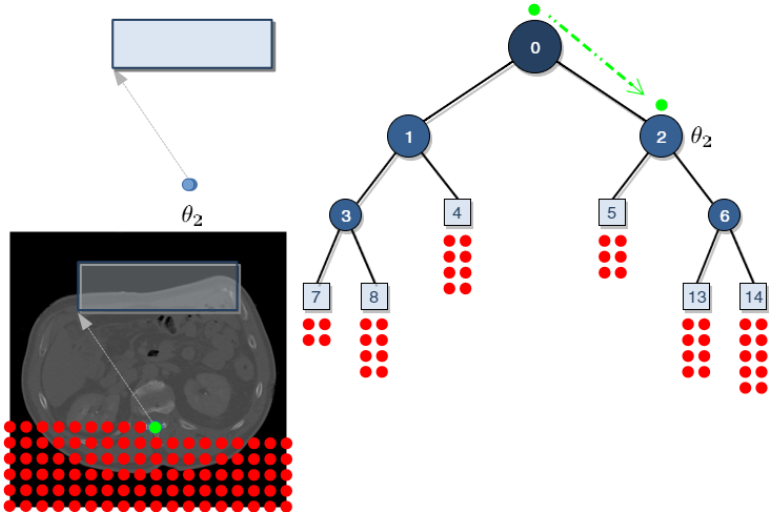


(a) The voxel (in green) starts the descent at the root node. The saved feature at the root (θ_0) is super imposed on the voxel to calculate the feature response. The voxels shown in red have already reached the corresponding leaf nodes.

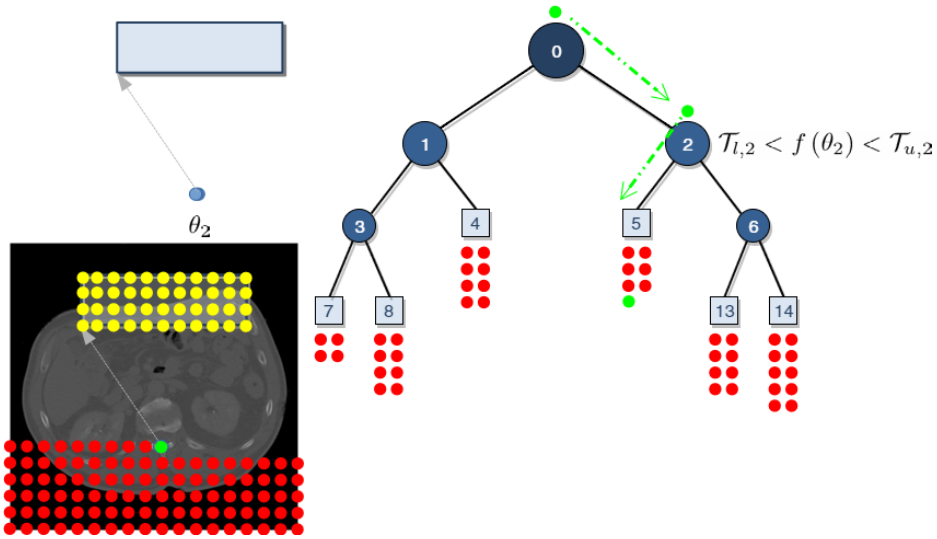


(b) The feature response is calculated by averaging the intensity values of voxels within the displaced cuboid (voxels shown in yellow). Depending on the feature response, the voxel is sent to the right split node.

Figure 3.20: Pushing a voxel through a trained forest.

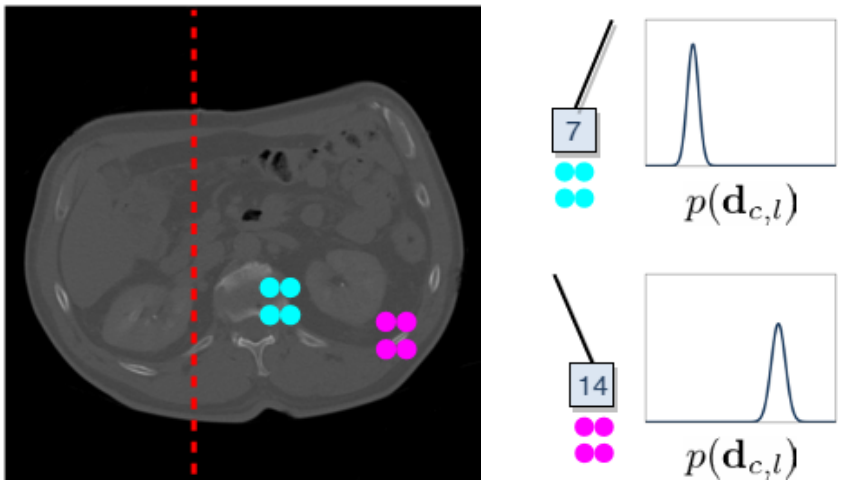


(c) At the new split node, the saved feature (θ_2) is used to calculate the feature response at the voxel.

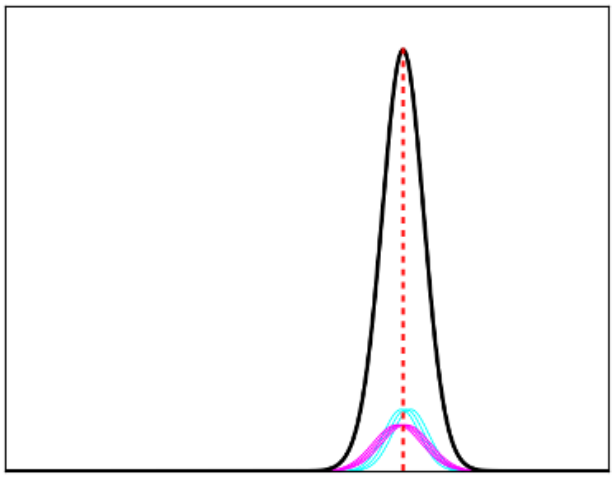


(d) The feature response is calculated by averaging the intensity values of voxels within the displaced cuboid (voxels shown in yellow). Depending on the feature response, the voxel is sent to the left node. Since it is a leaf node, the voxel has reached its final destination.

Figure 3.20: Pushing a voxel through a trained forest (cont.).



(a) (b)



(c)

Figure 3.21: Bounding wall prediction using the offset vector distributions. (a) The voxels that participate in the final prediction belong to two leaf nodes and these voxels are shown in the medical image in cyan and magenta. (b) The corresponding two leaf nodes (7 and 14) from Fig. 3.19 and their stored offset vector distributions for one direction (l–left) are presented next to them. (c) The prediction distribution of $p(\mathbf{b}_{c,l})$ from each voxel is accumulated in a single histogram per bounding wall direction. The summation of all distributions are presented in the thick black curve. Final predicted position $\hat{\mathbf{b}}_{c,l}$ is the arg max of the distribution summation (shown in red dashed line). It is also shown in (a) using a red dashed line.

In this chapter, we introduced, analyzed, and discussed the core of this dissertation; the concept of Random Regression Forest and multi–organ localization.

First, we presented the concept of ensemble methods in Sect. 3.1. Then, in Sect. 3.2, we introduced different types of Random Forests and reported their various applications in the field of medical image analysis. A special prominence was given to the literature related to RRFs in this section. Next, we introduced the decision jungles and random ferns and the differences between Random Forests (RFs) and them in Sect. 3.3.

In Sect. 3.4, we analyzed the inner workings of Random Regression Forests and their parameters in detail. The analysis was carried out in the context of multi–organ localization and it was discussed in Sect. 3.4.1. The intrinsic parameters of a RRF, the training set preparation, and image preprocessing were presented and analyzed in Sect. 3.4.2, Sect. 3.4.3, and Sect. 3.4.5 respectively. We carried out an extensive analysis of the training and prediction phases of RRFs in Sect. 3.4.6 and Sect. 3.4.7. Finally, the algorithmic steps of forest training and prediction were summarized respectively in Sect. 3.5.1 and Sect. 3.5.2 with a few illustrations.

In the next chapter, we will detail the scientific approach adopted to carry out the various studies of the dissertation along with the information on our CT image database and the benchmarking technique.

Methodology should not be a fixed track to a fixed destination, but a conversation about everything that could be made to happen.

- John Chris Jones

4

Methodology

Contents

Now that the inner workings of a Random Regression Forest and its parameters have been analyzed in Chap. 3, we have laid the foundation to move forward in exploring the impact of certain parameters and the robustness of the algorithm under various conditions. The scientific approach adopted for this exploration is presented in Sect. 4.1. Next, the database used for the various studies and its diversity are discussed in Sect. 4.2. Finally, the chapter concludes with Sect. 4.3, where we present the general evaluation criteria employed along with the relevant implementation details.

4.1 Scientific Approach

In realizing and furthering the objective of this dissertation mentioned in Sect. 1.3, we wish to carry out a rigorous analysis of the robustness of the Random Regression Forest (RRF) method in the context of multi-organ localization. The robustness of the RRF method is examined with respect to a selected set of parameters and concepts which will be explained in the remaining chapters of this dissertation. In order to reach this goal we have opted for the following scientific approach.

First, we asked ourselves a question regarding a pertinent concept (*e.g.*, does the spatial independency hypothesis hold?) or a parameter (*e.g.*, how many leaf nodes should participate in the localization prediction?) related to the RRF method. Then we analyzed the theoretical and/or mathematical implications of the question. Once the theoretical and/or mathematical implications were clear, we verified whether the practical implementations corroborated the theory using empirical studies.

In designing these empirical studies, we always adhered to the following guidelines. We only verified one concept or one parameter influence at a time using a single empirical study. The same database of Computed Tomography (CT) images was used throughout these studies with the exact same training and prediction image sets. The set of organs used in each study was identical. We used the same base software implementation for all studies to perform the localizations. The results were generated using the same machine. The localization evaluation was carried out with respect to the same evaluation measures against the same benchmark.

By adhering to the above mentioned guidelines, we were able to assess the impact of each criterion on the final localization. The following sections expand on each of the guidelines presented above.

4.2 Database

Throughout the course of this dissertation we have used the same database of CT images. The database is composed of 100 anonymized CT images belonging to 100 different patients which were directly obtained from the teaching hospital of Grenoble Alpes[1]. Since the images are those of real patients, obtained directly from the hospital, it is an excellent representation of the real world.

4.2.1 Diversity of The Database

First and foremost, the database comprised both male and female adult patients whose ages ranged from 21 years to 90 years (mean = 61.9 years and standard deviation = 18.7 years). Certain pathological symptoms (cysts, enlarged organs, etc.) were clearly observed in some of the images (see Fig. 4.1a, 4.1b, 4.1c, 4.1d, 4.1e and 4.1f). The images also showed a high variability in size. More precisely, the sizes ranged from $263 \times 263 \times 366$ mm^3 to $466 \times 466 \times 568$ mm^3 or from $512 \times 512 \times 183$ voxels to $512 \times 512 \times 284$ voxels.

Since irradiation is harmful to the patients, only the sections of the body that needed scrutinizing had been scanned. As a consequence:

[1]We are extremely grateful to Prof. Ivan Bricault of the teaching hospital of Grenoble Alpes for providing us with the database in addition to the numerous advices and suggestions.

- whole-body CT images were not present in our database and
- the imaged region varied considerably depending on the image.

As a result, all the chosen images roughly cover the abdominal and pelvic regions.

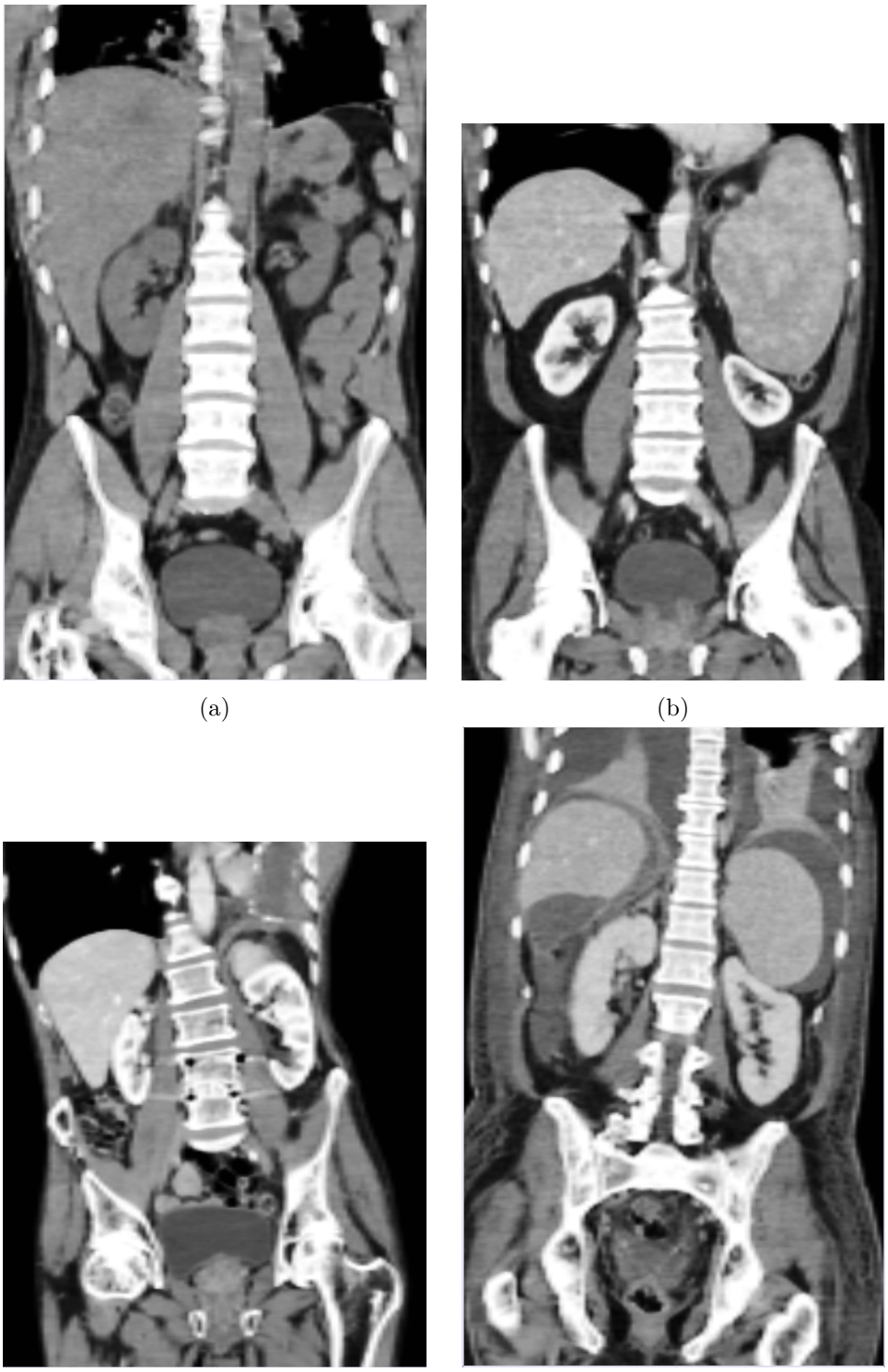
Certain patients had biomedical implants that introduced artifacts in the CT images (see Fig. 4.1g and Fig. 4.1h). In certain images, parts of the surgical instruments used by the surgeons were also visible increasing the heterogeneity of the database. The presence of these objects can be considered as a disturbance of the expected natural context of a CT image that may perturb automatic organ localization procedures.

Certain patients were imaged after introducing contrast agents (generally iodine or barium based contrast mediums) to their bodies for better visualization of targeted organs whereas other patients were not (see Fig. 4.1b, 4.1c, 4.1e, 4.1f, 4.1g and 4.1h). The existence or non-existence of contrast agents also change the CT image properties increasing the diversity of the database. Additionally, the amount of noise present in the images were different from one to another.

Three CT scanners; namely, Philips Brilliance 64, Philips Brilliance 16, and Siemens Sensation 16 were used to acquire these images. As the machines are not the same, the parts of the machines that are captured in the images (i.e., part of the machine bed) also differ[2], adding to the diversity of the dataset.

A few coronal slices of some of the images of the database are presented in Fig. 4.1. The contrast of the images are adjusted for the enhanced viewing of the abdominal soft tissue organs (window width = 350 Hounsfield Unit (HU) and window level = 40 HU [Johnson et al., 2007]).

[2]Since the random regression forest method discovers salient landmarks, it would be interesting to see whether the imaged parts of the CT machines are discovered as pertinent landmarks for certain organs.



(a) (b)

(c) (d)

Figure 4.1: A few coronal slices of the database. (a) Hepatomegaly: abnormally enlarged liver. (b) Splenomegaly: abnormally enlarged spleen. (c) Left sided empyema: pus within pleural space with pleural calcification. (d) Ascites: free fluid in the abdomen and pleural effusions: free fluid around the lungs.

(e)

(f)

(g)

(h)

Figure 4.1: A few coronal slices of the database (cont.). (e) Renal pelvis dilatation and pelvi–ureteric junction obstruction: development of a blockage near the kidney in the ureter. (f) Renal cysts. (g) Pelvic internal fixator and renal cysts. (h) Left hip joint prosthesis. Right femoral neck fixation with cannulated screws. Collapsed left lower lobe of lung.

4.2.2 Organs Used in The Study

Since our database consisted of abdominal-pelvic CT images as mentioned in Sect. 4.2.1, we chose 9 anatomical structures (called organs here after) from the abdominal-pelvic regions. The selected organs are a mix of soft tissue organs and bone structures. The left kidney, right kidney, liver, and spleen are the 4 selected soft tissue organs whereas the left femur head, right femur head, left pelvis, right pelvis, and L5 vertebra are the selected bone structures. Across the main studies carried out during this dissertation, always the same organs were used, in accordance with the unifying approach mentioned in the Sect. 4.1.

4.2.3 Gold Standard Creation

In order to train a random regression forest, the ground truth of the training set needs to be known *a priori* (see Sect. 3.4.4). Additionally, the ground truth of the testing set also needs to be known in order to evaluate the localization estimation given by a trained random regression forest using some error measures (see Sect. 4.3.2). In the case of multi-organ localization, the ground truth is the localization of each selected organ with respect to the origin of the image (regardless of the image size and voxel dimensions). As we are defining the localization using bounding boxes, the ground truth should also be the tightest bounding box containing the respective organ. But unfortunately, this ground truth can not be known.

Instead, users with expertise on localizing these organs can manually delineate tight bounding boxes containing the respective organs. Although this may be highly subjective and dependent on the user, this is the best approximation of the ground truth. Hence, human users[3] with substantial training defined the gold standard of our dataset by manually delineating all fully inclusive organs with respect to 3D bounding boxes (see Fig. 4.2).

4.2.4 Training and Testing Set Separation

The database was randomly divided into a training set (Ω_{tr}) and testing (or prediction) set (Ω_{te}) consisting of 55 and 45 images respectively similar

[3]We are grateful to Anthony Agustinos, Ahmad Bijar, Cecilia Hughes, Vincent Léal, Paul Mignon, and Sonia Selmi for manually delineating bounding boxes of all CT images of the database.

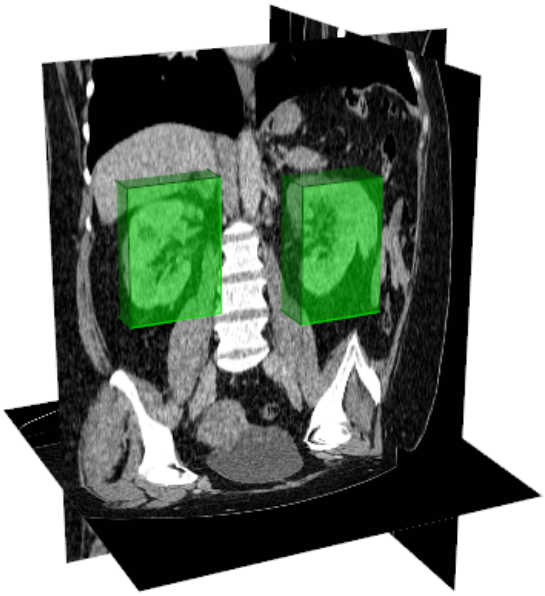


Figure 4.2: A sample image of the gold standard obtained by manual delineation. Only the left kidney and the right kidney are shown for clarity.

to [Criminisi et al., 2010]. Additionally, this separation was kept intact throughout the three main studies conducted during this dissertation in order to compare results across studies.

4.3 Benchmarking Random Regression Forests

To the best of our knowledge, the study of Criminisi et al. [2013] is the state of the art method for multi-organ localization in CT images. Naturally, their study was selected as the benchmark for our studies that followed. Unless otherwise specified, the benchmark forest comprises 4 random regression trees each having 12 maximum decision levels.

In Sect. 4.3.1, we present the details common to all implementations. The evaluation criteria, along with the statistical tests are presented in Sect. 4.3.2.

4.3.1 Implementation

Following the scientific approach defined in Sect. 4.1, random regression forest algorithm was implemented in–house as a C++ software library module using the software framework *Computer assisted medical intervention Tool Kit (CamiTK)* [Fouard et al., 2012]. The implementation of the module was carried out in a generic and modular manner so that it is only dependent on two other software libraries: the *Insight Segmentation and Registration Toolkit (ITK)* [Johnson et al., 2013] for image processing operations and the *Visualization Tool Kit (VTK)* [Schroeder et al., 2006] for visualization operations.

The studies were carried out on a quad-core Intel® Xeon® E5–1607, 3 GHz machine with 32 GB of RAM.

4.3.2 Localization Evaluation

Classically, in image segmentation each voxel of an image is classified whether belonging to the object or not. Consequently, many spatial overlap indexes can be employed to measure the *goodness* of the segmentation procedure [Zou et al., 2004; Taha and Hanbury, 2015]. Among them, Dice similarity coefficient [Dice, 1945] and Jaccard similarity coefficient [Jaccard, 1912] are two of the famous measures widely used in the literature. Both measures take into account the intersection and union of the estimated volume and gold standard. Both Dice and Jaccard similarity coefficients range from 0 to 1, where 0 and 1 mean total segmentation failure and perfect segmentation respectively.

But in the literature on multi-organ localization using random forests where further segmentation is not performed, these measures have not been used [Criminisi et al., 2010; Pathak et al., 2011; Pauly et al., 2011; Criminisi et al., 2013; Criminisi and Shotton, 2013]. One of the main reasons for not using those measures may be due to the fact that these methods do not localize the organ as one entity, but localize each wall of its bounding box separately (see Sect. 3.4.6.1). Hence, a different set of *goodness* measures are required to evaluate the multi-organ localization procedures. In the following sections a number of such evaluation measures are presented.

4.3.2.1 Bounding Wall Prediction Error

Bounding Wall Prediction Error (BWPE) seems to be the most widely used quantitative measure among the multi-organ localization community [Criminisi et al., 2010; Pathak et al., 2011; Criminisi et al., 2013; Criminisi and Shotton, 2013]. Since the organ localization strategy using random regression forests considers each bounding wall of an organ separately, the BWPE considers each wall estimation separately as well.

BWPE is defined as the absolute difference between the estimated bounding wall ($\hat{\mathbf{b}}^i$) and bounding wall of the gold standard ($\mathbf{b}^i$) (see Fig. 4.3). As there are six walls per bounding box, BWPE is given for the right, left, anterior, posterior, inferior, and superior walls. That is:

$$\text{BWPE}^i = |\mathbf{b}^i - \hat{\mathbf{b}}^i| \ , \tag{4.1}$$

where $i \in \{\text{r}, \text{l}, \text{a}, \text{p}, \text{i}, \text{s}\}$, r = right, l = left, a = anterior, p = posterior, i = inferior, and s = superior directions respectively. Since, the prediction is done per bounding box wall, the bounding wall prediction error appears to be a more justified error measure than spatial overlap measures.

When a compound error measure is required per organ, the mean BWPE is calculated taking the mean of the six BWPEs:

$$\text{mean BWPE} = \frac{1}{6} \sum_{i \in \{r,l,a,p,i,s\}} \text{BWPE}^i \ . \tag{4.2}$$

Finally, BWPEs are calculated only for the organs that are fully enclosed in the images.

4.3.2.2 Centroid–Hit Measure

Centroid–Hit Measure (CHM) is a qualitative measure that verifies whether the centroid of the predicted bounding box lies within the bounding box of the gold standard. Similarly to the bounding wall prediction error, centroid–hit measure is also not a compound measure. Centroid–hit measure is calculated along the right to left, anterior to posterior, and inferior to superior

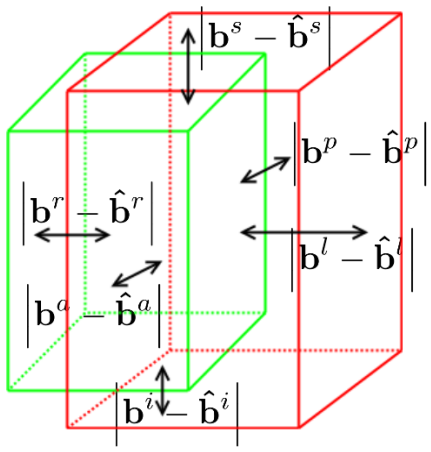


Figure 4.3: Bounding wall prediction errors between left, right, anterior, posterior, inferior, and superior walls of the predicted bounding box (in red) and gold standard (in green).

directions in the following manner.

$$\text{CHM} = \begin{cases} 1 & \text{if } \mathbf{b}^i < \frac{\hat{\mathbf{b}}^i + \hat{\mathbf{b}}^{j(i)}}{2} < \mathbf{b}^{j(i)} \\ 0 & \text{otherwise} \end{cases} , \tag{4.3}$$

where $i \in \{\text{r}, \text{a}, \text{i}\}$ and $j \in \{\text{l}, \text{p}, \text{s}\}$. The studies Pathak et al. [2011]; Criminisi et al. [2013]; Criminisi and Shotton [2013] have used the centroid–hit measure to evaluate the localization predictions.

4.3.2.3 Centroid Error

In order to obtain a compound evaluation criterion, we defined the Centroid Error (CE). The centroid error is the euclidean distance between the centroid of the prediction and the gold standard (see Fig. 4.5).

$$\text{CE} = \sqrt{\triangle x^2 + \triangle y^2 + \triangle z^2} , \tag{4.4}$$

where the distances $\triangle x = \left(\frac{\hat{\mathbf{b}}^r + \hat{\mathbf{b}}^l}{2} - \frac{\mathbf{b}^r + \mathbf{b}^l}{2}\right)$, $\triangle y = \left(\frac{\hat{\mathbf{b}}^a + \hat{\mathbf{b}}^p}{2} - \frac{\mathbf{b}^a + \mathbf{b}^p}{2}\right)$, and $\triangle z = \left(\frac{\hat{\mathbf{b}}^i + \hat{\mathbf{b}}^s}{2} - \frac{\mathbf{b}^i + \mathbf{b}^s}{2}\right)$. The centroid error measure can be thought of as a quantitative version of the centroid–hit measure described previously.

Throughout the main studies carried out during this dissertation, the localization prediction evaluation was carried out using these three measures

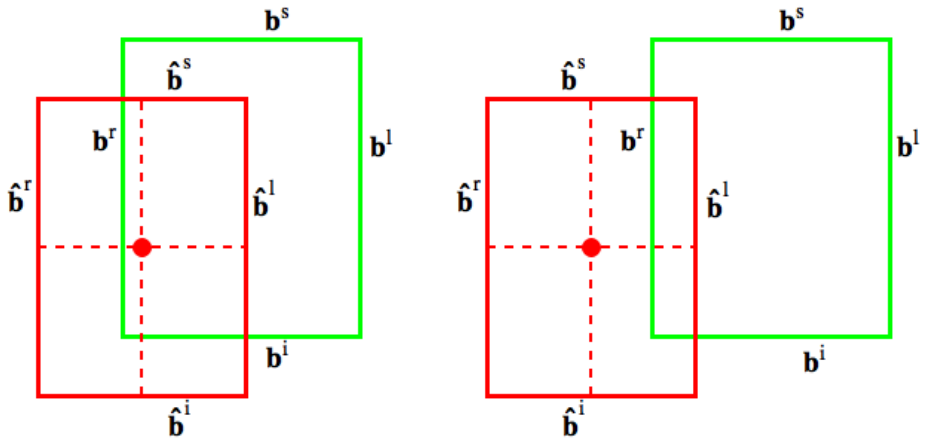


Figure 4.4: Centroid hit measure for coronal slices. (a) Centroid of the predicted bounding box (in red) lies within the gold standard bounding box (in green). (b) Centroid of the predicted bounding box falls outside the gold standard bounding box. In this case this is what is called a *centroid miss* along the right to left direction.

in accordance with the unifying approach mentioned in the Sect. 4.1.

4.3.2.4 Statistical Tests

To check the robustness of a method against a change of a parameter (or to compare two methods), we have to verify that the new results are significantly different than before. Hence, to compare a proposed modification to the benchmark, we used the Wilcoxon–Mann–Whitney test [Mann and Whitney, 1947] to check whether the two result sets were statistically significant or not. In this manner, we were able to assess whether one method (the benchmark or the proposition) was statistically better than the other.

In this chapter, we presented the scientific approach that we followed in conducting the studies presented in the next 3 chapters in Sect. 4.1. Then, the diversity of the database used and the creation of the gold standard were presented in Sect. 4.2. Finally, we concluded the chapter by providing implementation specific details along with how the localization performance can be benchmarked.

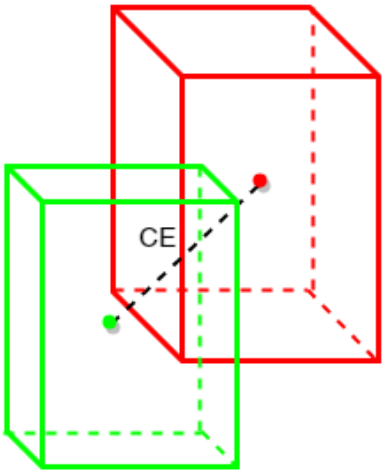


Figure 4.5: Centroid error is the euclidean distance between the predicted bounding box (in red) and the bounding box of the gold standard (in green).

In the next three chapters we are going to present the main studies conducted during this dissertation. First, in Chap. 5 we verify whether the localization prediction can be further improved by adding more spatially consistent information. Then, in Chap. 6 we propose Light Random Regression Forests, an approximation to find the localization prediction using random variables rather than describing the random process. Finally, in Chap. 7, we propose an automatic and consistent approach in order to increase the generality of RRFs.

In theory, there is no difference between theory and practice. But in practice, there is.

- Lawrence Peter "Yogi" Berra

5

Preserving Spatial Consistency of Random Regression Forests

Contents

During the analysis of random regression forests that was carried out in Chap. 3, we observed the spatial independent nature of the algorithm in the forest training phase (see Sect. 3.4.6) as well as in the prediction phase (see Sec. 3.4.7). During this chapter, we wish to find out the effect on the algorithm if spatial consistent information were to be added to it[a].

In Sect. 5.1, we pin point both the conceptual and implementation oriented spatial independent traits of the classic random regression forest formulation. The steps taken to verify the effect of spatial consistent information introduction into the classic RRF algorithm are presented in Sect. 5.2. The numerous performance as well as usability analysis that are carried out in Sect. 5.3 are followed by a concise discussion on the findings in Sect. 5.4. Finally, we conclude the chapter by Sect. 5.5 with a few concluding remarks and some perspectives.

[a]This chapter is based on an article submitted to the 7th International Workshop on Machine Learning in Medical Imaging (MLMI 2016)

5.1 Introduction

The Random Regression Forest (RRF) method is a very capable tool for multi–organ localization. Criminisi et al. have provided evidence that RRFs perform better than popular registration methods such as *Elastix* [Klein et al., 2010]. However, not only the formation of the regression problem but also certain implementation choices of the RRF algorithm do not preserve the spatial consistency.

5.1.1 Spatial Independency Traits of RRF

First, let us consider the regression problem formulation. As mentioned in the Sect. 3.4.6, the multi–organ localization problem is addressed by regressing the relative displacements of voxels from the bounding box walls of organs (see Sect. 3.4.6.1). This is achieved via offset vectors (see Fig. 5.1). An offset vector $(\mathbf{d}(\mathbf{v}, c))$, as seen in Sect. 3.4.6.1, is defined as the 6 independent displacements from each bounding box wall (namely, the right (r), left (l), anterior (a), posterior (p), inferior (i), and superior (s) walls) to a

given voxel. And it is defined in the following manner:

$$\mathbf{d}(\mathbf{v}, c) = \hat{\mathbf{v}} - \mathbf{b}(c) \ , \tag{5.1}$$

where $\mathbf{v}$ is the voxel position, c is the organ, $\mathbf{b}(c)$ is the bounding box vector, and $\hat{\mathbf{v}} = (\mathbf{v}_x, \mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_y, \mathbf{v}_z, \mathbf{v}_z)$ made from the voxel position $(\mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_z)$. Since each bounding wall direction is considered as an independent entity, the spatial consistency is not preserved.

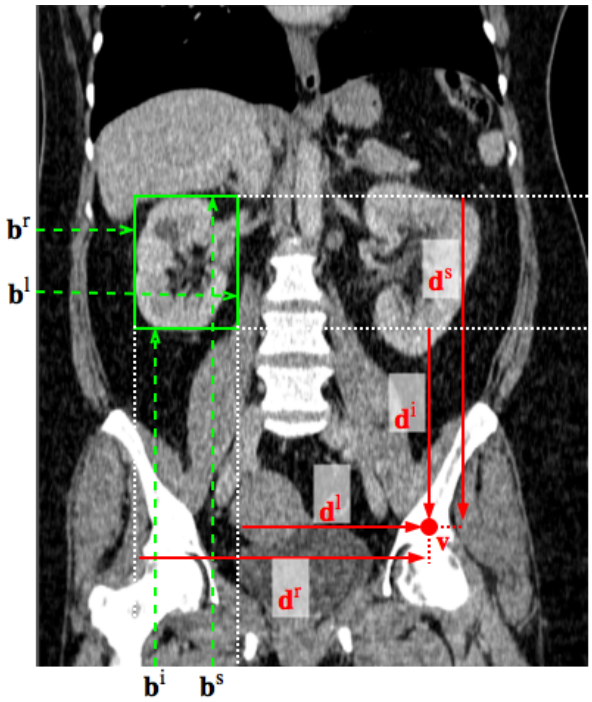


Figure 5.1: The offset vector is composed of 6 **independent** displacements from each bounding box wall to the voxel ($\mathbf{v}$). A coronal slice of a CT image and the 4 visible components of the offset vector (in red) of the right kidney of the voxel $\mathbf{v}$ and its bounding box vector (in green). As each displacement component is considered independently, the spatial consistency is not preserved.

Let us consider the current RRF implementation decisions where the spatial consistency is not preserved. In the training phase when the leaf nodes are trained, the offset vectors that accumulate at a leaf node are saved as 6 1D histograms per organ assuming wall location independence (see Sect. 3.4.6.5). Again, when the absolute location of each bounding box wall

is calculated in the prediction phase, each bounding box wall distribution is modeled using a 1D histogram (see Sect. 3.4.7.2). Consequently, the classic RRF setting hypothesizes that independent directional components are a good approximation of spatial consistency of *offset vectors* Criminisi et al. [2010, 2013].

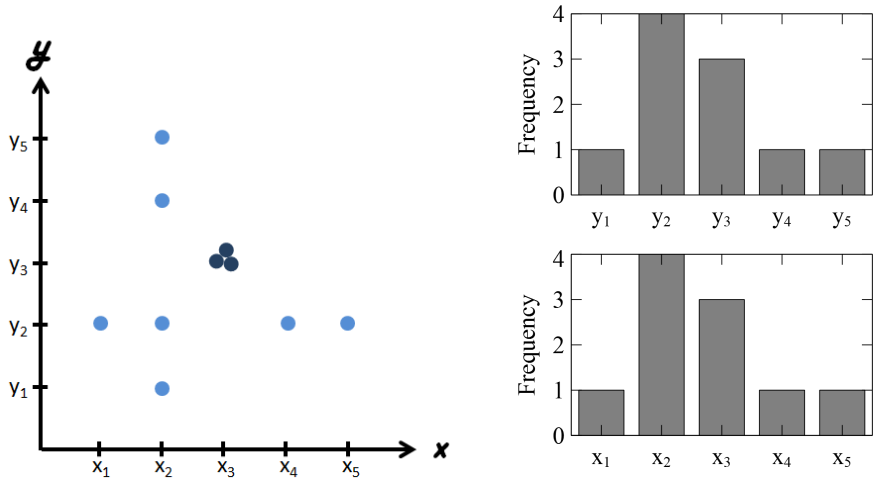
Not only it is intuitive, but it is also generally admitted that the accuracy and precision of a segmentation method is improved when spatial consistent information is taken into account [Pham et al., 2000]. As the human body adheres to a generic anatomy, it is rich with spatially consistent information. The current setting of RRF uses this spatial consistent information up to some extent by considering each bounding wall as an independent entity. But the spatial consistency can be improved by reducing the independent nature of the bounding walls. Therefore, adding more spatial consistent information in RRF should lead to more accurate and precise localizations.

5.1.2 Spatial Consistency Example

Let us elaborate our claim with the use of a simple example (see Fig. 5.2). Assume 10 instances of 2D offset vectors[1]. There are 7 offset vectors (x_1, y_2), (x_2, y_1), (x_2, y_2), (x_2, y_4), (x_2, y_5), (x_4, y_2), (x_5, y_2) with a single instance and 3 instances of the same offset vector (x_3, y_4). And we are interested in finding the mode of the offset vector distribution (similar to finding the Maximum *A–Posteriori* (MAP) of the offset vector distribution in the real organ localization scenario using a RRF). Clearly, the correct answer that we are searching for is the offset vector (x_3, y_3) with 3 occurrences.

In a setting where the spatial consistency is preserved, we arrive at the correct conclusion without any confusion (see Fig. 5.2). Yet, in a setting where the spatial consistency is not preserved, for instance, when the offset vectors are modeled as 2 1D histograms, the conclusion is not the same (see Fig. 5.2b). The modes of the 2 histograms in x and y direction are x_2 and y_2 respectively. This leads to the incorrect conclusion that (x_2, y_2) is the mode of the distribution where in fact only one instance of (x_2, y_2) is present. This simple observation implies that it may be possible to improve classic RRF by adding more spatial consistent information.

[1] Any entity with a spatial interpretation can be used instead of offset vetors.



(a) (b)

Figure 5.2: Distribution of 10 offset vectors $((x_1, y_2)$, (x_2, y_1), (x_2, y_2), (x_2, y_4), (x_2, y_5), (x_3, y_3), (x_3, y_3), (x_3, y_3), (x_4, y_2), and $(x_5, y_2))$ where the goal is to find the mode of the distribution. (a) In a setting that preserves the spatial consistency, (x_3, y_3) can clearly be seen as the mode. (b) The use of two spatially independent histograms leads to the incorrect conclusion that (x_2, y_2) is the mode of the distribution.

The goal of this study, it to verify whether such an improvement actually occurs if spatial consistent information is added to the classic RRF.

5.2 Materials and Method

In order to verify whether such an improvement is possible, we extended the classic RRF algorithm to a so called Hough–based Random Regression Forest (HbRRF) that preserves the spatial consistency. The following sections describe the main changes of HbRRF compared to the classic RRF.

5.2.1 Hough–based Random Regression Forest

A Hough–based Random Regression Forest is similar to classic a RRF but has two interpretation differences, described in Sect. 5.2.2, and a few implementation differences, described in Sect. 5.2.3 and Sect. 5.2.4.

5.2.2 Offset Vector Interpretation

The first difference between classic RRFs and HbRRFs concerns the interpretation of the offset vectors $\big(\mathbf{d}\,(\mathbf{v}, c)\,\big)$. A bounding box of an organ can be defined using its two extreme diagonal points in space. Therefore, in HbRRFs, we choose to interpret the offset vectors as the displacement of a given voxel from these two points (see Fig. 5.3) contrary to the classic interpretation of offset vectors presented in Sect. 5.1.1. Therefore:

$$\begin{aligned} \mathbf{d}(\mathbf{v}, c) &= \{\mathbf{d}_1(\mathbf{v}, c), \mathbf{d}_2(\mathbf{v}, c)\} \\ \mathbf{d}_x(\mathbf{v}, c) &= \mathbf{v} - \mathbf{b}_x(c) \ , \end{aligned} \tag{5.2}$$

where $x \in \{1, 2\}$, $\mathbf{v} = (v_x, v_y, v_z)$, c is the organ, and $\mathbf{b}_x(c)$ is the corresponding bounding box extreme point. The only difference provided by this interpretation is that it preserves the spatial consistency.

Figure 5.4 shows a schematic representation of the two different interpretations. Further analysis of the diagram emphasizes that the only difference among the two representation is purely interpretational.

The second interpretation difference is regarding the formulation of the regression problem. Instead of regressing the relative displacements of voxels from each bounding box wall of an organ independently as classic RRFs do, a HbRRF regresses the displacement of voxels from two extreme points of each bounding box of an organ. As HbRRFs consider each extreme point of a bounding box as one entity, the spatial consistency is preserved unlike in classic RRF.

5.2.3 Training Phase Differences

The first implementation difference between classic RRF and HbRRF concerns the forest training phase. More precisely, it occurs during the leaf node training (see Sect. 3.4.6.5 and step 6 of Sect. 3.5.1) and is described below. The other steps of the forest training phase do not change.

The manner in which the spatial distribution of the accumulated offset vectors are stored differ between RRF and HbRRF. In HbRRF for each extreme point of each bounding box, the exact positions given by the accumulated offset vectors as opposed to 1D directional components are saved. In this manner, the spatial consistent information is preserved and can be used in the prediction phase. Since the space requirement of saving all the

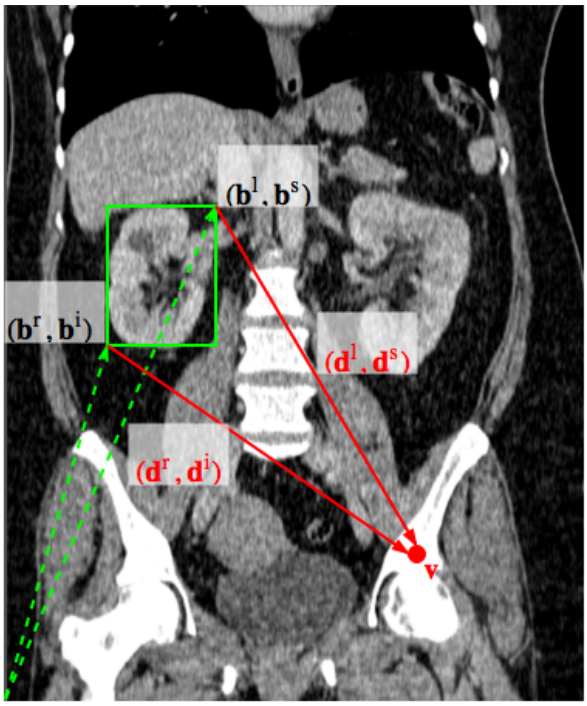


Figure 5.3: The offset vector of HbRRF is interpreted as the displacement of a given voxel from the two extreme points of an organ bounding box. This interpretation preserves the spatial consistency. A coronal slice of a CT image and the offset vector (in red) of the right kidney of the voxel $\mathbf{v}$ and its bounding box vector (in green). As each displacement component is considered as one entity, the spatial consistency is preserved.

positions is very large, only the a limited number of positions (n_{saved}) with the highest frequency are saved at each leaf.

5.2.4 Prediction Phase Differences

At the prediction phase, all selected voxels of the unseen image are pushed through the forest and are ultimately accumulated at some leaf nodes across the forest similar to the classic RRFs (*i.e.*, the first two steps of the prediction process presented in Sect. 3.5.2 are the same for both the forests).

The other four prediction process steps mentioned in Sect. 3.5.2 are different for HbRRFs. In order to determine the number of leaf nodes that participate in the prediction of each organ, the following actions are performed. First, for each extreme point of an organ bounding box, all leaf nodes of the RRF are sorted with respect to the covariance of the offset

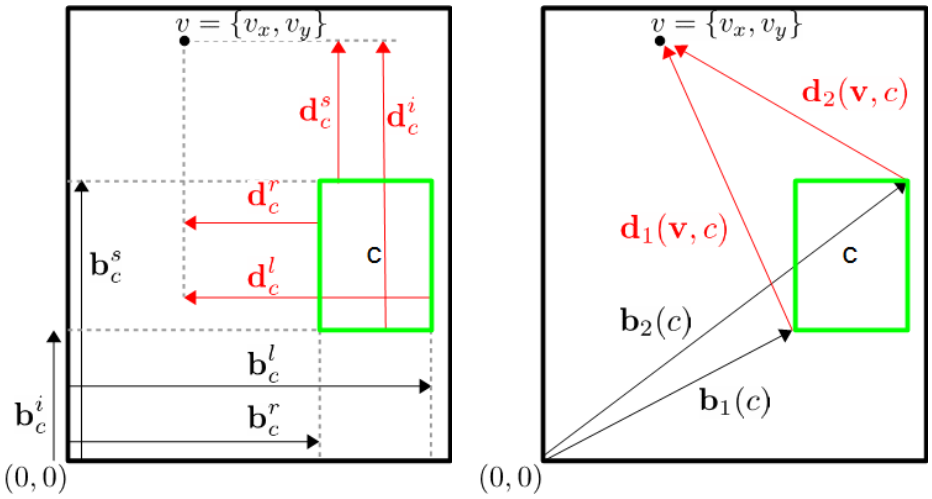


Figure 5.4: Schematic 2D representation of bounding box vectors and offset vectors in an coronal slice of a CT image. (a) In classic RRF, offset vectors are interpreted as independent displacements in four directions $\mathbf{d}_c^{dir}$. (b) In HbRRF, they are interpreted as the two displacements $\mathbf{d}_x(\mathbf{v}, c)$, one from each extreme point of the bounding box.

vectors found at the training phase (smaller covariances imply more precise predictions). The leaf nodes with the smallest covariances that accumulate at least a pre-determined percentage of image voxels p_v are selected for prediction.

Then, given an accumulated voxel at a leaf node ($\mathbf{v}$), this voxel votes for the spatial location of the bounding box extreme point ($\mathbf{b}_x(c \mid \mathbf{v})$) using the saved offset vector positions where c denotes the organ. Each offset vector position votes for a possible bounding box extreme point in the following manner:

$$\mathbf{b}_x(c \mid \mathbf{v}) = \mathbf{v} - \mathbf{d}_x(c \mid \mathbf{v}) \ , \tag{5.3}$$

where $x \in \{1, 2\}$. These votes are accumulated in a spatial volume and the position of the highest vote of the volume corresponds to the final localization of each extreme point of an organ bounding box.

5.2.5 Materials and Implementation Details

In order to verify whether the introduction of spatial consistency leads to any improvement in classic RRF, we followed the methodological details

mentioned in Chap. 4.

The dataset described in Sect. 4.2 was used for the study and was randomly divided into a training set and a prediction set of 55 and 45 images respectively, similarly to [Criminisi et al., 2010]. The set of organs described in Sect. 4.2.2 were used in this study. Prediction evaluation was carried out using mean Bounding Wall Prediction Error (BWPE), Centroid–Hit Measure (CHM), and Centroid Error (CE) as mentioned in Sect. 4.3.2. The statistical test mentioned in Sect. 4.3.2.4 was used to verify the difference between the results obtained by the two algorithms. The tests were carried out under the hardware configuration as described in Sect. 4.3.1.

The classic RRF method was implemented as described in Criminisi et al. [2013]. This implementation was modified to build a Hough–based Random Regression Forest as described in section 5.2.1 in order to compare the two methods in the exact same conditions. Each random forest (classic RRF and HbRRF) comprised 4 regression trees and each tree had 12 maximum decision levels (D) following the benchmark protocol selection mentioned in Sect. 4.3. The split node training was terminated if the number of voxels accumulated at a node was less than 25 ($n_{\min}$). Finally, two localization result sets were generated using the classic RRF (R_C) and the HbRRF (R_H) with $p_v = 0.05$ for both, and with $n_{\text{saved}} = 2,500$ for R_H.

5.3 Results

The mean Bounding Wall Prediction Error was used as the first performance evaluation metric (Er_1) and was calculated only for the organs that were fully present in the CT volumes.

Table 5.1 presents the summary of mean BWPEs (mean, standard deviation, median, and maximum of all 45 prediction results) obtained for R_C and R_H. The numbers alone do not reveal much difference between the two result sets.

In order to access the mean BWPE of individual prediction localization, the mean BWPEs for R_C and R_H obtained for each prediction image are presented as 1.5 Inter Quartile Range (IQR) box plots in Fig. 5.5. With an alpha level of 0.05 for the statistical tests, Er_1 between R_C and R_H demonstrated no statistical significance, $p-\text{value} \in [0.34 - 0.90]$. The p–values obtained for each organ is presented in Table 5.2.

Table 5.1: Summary of mean BWPEs for R_C and R_H

	Left Kidney	Right Kidney	Liver	Spleen	Left Femur Head	Right Femur Head	Left Pelvis	Right Pelvis	L5
Mean R_C	10.76	11.74	15.37	14.62	6.76	6.61	6.06	6.98	7.43
Mean R_H	10.07	11.97	14.75	14.35	6.58	6.27	5.58	5.97	7.15
Std. dev R_C	10.73	10.08	15.38	13.47	6.07	6.16	5.89	5.04	6.13
Std. dev R_H	9.51	10.40	14.85	14.04	5.93	5.89	5.20	4.33	5.77
Median R_C	7.77	9.55	10.17	10.20	5.21	4.58	4.57	6.05	5.88
Median R_H	8.00	9.21	9.82	10.55	5.42	4.57	4.23	4.93	5.68
Max R_C	82.0	62.0	91.2	73.0	32.0	31.5	34.0	20.8	28.0
Max R_H	77.0	65.0	101.3	86.0	30.0	31.5	32.0	19.0	26.0

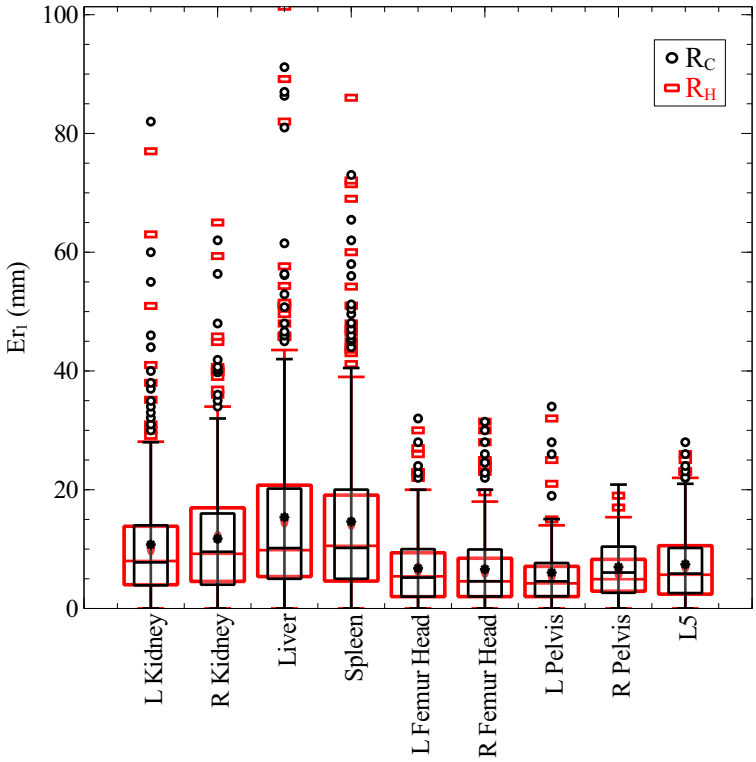


Figure 5.5: 1.5 Inter Quartile Range (IQR) box plots of mean BWPE of the 45 prediction images for R_C and R_H.

Table 5.2: Obtained p–values for the mean BWPEs

Organ	p–value
Left Kidney	0.90
Right Kidney	0.87
Liver	0.73
Spleen	0.62
Left Femur Head	0.73
Right Femur Head	0.66
Left Pelvis	0.71
Right Pelvis	0.34
L5	0.75

As the final performance evaluation metric we used the Centroid Error (CE) of each bounding box prediction (Er_2). Similarly to Er_1, Er_2 was also

calculated only for the organs that were fully present in the CT volumes. The summary of all CEs are given in the Table 5.3. Similarly to the summary of mean BWPEs, not much difference can be observed between two Er_2 value sets belonging to R_C and R_H respectively.

In order to further evaluate the centroid errors of individual prediction localizations, 1.5 IQR box plots were generated (see Fig. 5.6). Subjecting the Er_2 values to the statistical test with an alpha level of 0.05, we observed that there was no statistical significance between R_C and R_H, $p-$value $\in [0.34 - 0.96]$. All $p-$values obtained are presented in Table 5.4.

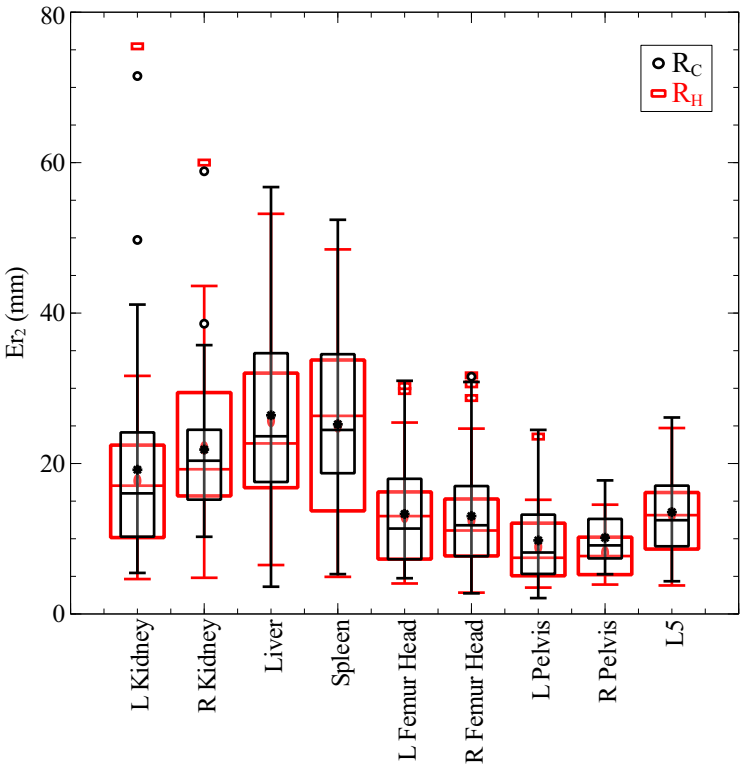


Figure 5.6: 1.5 IQR box plots of centroid errors for R_C and R_H.

The usability was evaluated using the time taken to localize all organs in a CT volume (t), the amount of Random Access Memory (RAM) required during the prediction phase to localize all organs in a given CT volume (M_p) and the amount of disk space used to store a RRF (M_s).

The results of usability evaluation are presented in Table 5.5. Time spent on localizing one CT volume using R_H is about 140 times slower compared to R_C. On average, R_H also uses about 3.6 times more RAM than R_C does.

Table 5.3: Summary of centroid errors for R_C and R_H

	Left Kidney	Right Kidney	Liver	Spleen	Left Femur Head	Right Femur Head	Left Pelvis	Right Pelvis	L5
Mean R_C	19.18	21.85	26.41	25.22	13.29	13.01	9.78	10.14	13.54
Mean R_H	17.70	22.09	25.71	24.98	12.97	12.48	8.93	8.14	13.18
Std. dev R_C	12.47	9.16	12.67	10.43	6.95	7.36	6.02	4.00	5.82
Std. dev R_H	11.67	10.21	12.04	11.00	6.86	6.99	5.53	3.48	5.32
Median R_C	16.03	20.37	23.62	24.47	11.35	11.79	8.17	9.12	12.46
Median R_H	17.06	19.24	22.67	26.34	13.01	11.10	7.47	7.70	13.14
Max R_C	71.51	58.85	56.75	52.40	31.00	31.55	24.47	17.76	26.12
Max R_H	75.46	60.01	53.20	48.47	30.30	31.72	23.58	14.53	24.72

Table 5.4: Obtained p–values for the centroid errors

Organ	p–value
Left Kidney	0.92
Right Kidney	0.70
Liver	0.96
Spleen	0.34
Left Femur Head	0.64
Right Femur Head	0.70
Left Pelvis	0.75
Right Pelvis	0.62
L5	0.83

Table 5.5: Usability evaluation measures.

	t (s)		M_p (MB)		M_s (GB)	
	R_C	R_H	R_C	R_H	R_C	R_H
Mean	4.3	613.7	806.2	2900.3	0.9	6.1
Std. dev	0.6	109.6	42.8	247.0	-	-

In addition to that, R_H uses about 6.8 times more storage capacity to store the RRF compared to R_C. All the provided evidence suggests that R_C have much better usability criteria than R_H.

5.4 Discussion

As the amount of disk space needed to store all the distributions was extremely large (17.2 GB for a HbRRF with 4 RRT), only $2,500$ distinct positions of offset vectors were saved for each extreme point of each bounding box during the training phase (see Sect. 5.2.3). The value $n_{\text{saved}} = 2,500$ was selected as we observed that many positions were unique when a large number of positions were accumulated. Consequently, those positions with reduced number of frequencies had little to no effect on the final outcome. In addition to that, allowing 100 times more positions than $n_{\min}$ ensured that the leaf nodes gathering more voxels had a higher impact on the final prediction.

The notion of spatial independence is inherently present in the BWPE as error measurements are obtained per direction. The authors of Pathak et al. [2011] introduced the centroid-hit error. The centroid-hit error is a qualitative error measure that verifies whether the centroid of the predicted bounding box is found within the gold standard bounding box. Using centroid error, we introduced a quantitative error metric that also preserved the spatial consistency by considering the centroid of a bounding box as one entity and not as independent directional components.

When split node optimization was carried out (see Sect. 5.2.3) for the HbRRF, instead of optimizing two separate extreme points, we optimized both extreme points together. This was done because the extreme points did not correspond to unique landmarks while the organs as a whole did.

5.5 Conclusion

Preserving spatial consistency of offset vectors that accumulate at the leaf nodes of a RRF, as it is done in the proposed Hough-based RRF, should have intuitively lead to better localization results as theory suggests and as it can be observed in other methods. Counter–intuitively, this study provides empirical evidence that the classic RRF method produces comparable results. In addition to that, the introduction of spatial consistency brings about a drastic reduction of the usability of the algorithm. Hence, we advocate using classic RRF over spatial consistency preserved RRF.

As presented in Table 5.5, although classic RRF performed better than the Hough-based RRF in terms of RAM and storage requirements, it still seems to be not that well suited for mobile applications. This aspect is going to be explored in the next chapter.

In this chapter, we analyzed the effect of adding more spatial consistent information in to a classic RRF method.

In Sect. 5.1, we analyzed the spatial independent nature of the classic RRF problem formulation for multi–organ localization. With a simple toy example, we demonstrated the possible improvements that may occur by the addition of spatial consistent information in Sect. 5.1.2. The Sect. 5.2 presented how we added spatial consistent information by ex-

tending the classic RRF method. We analyzed the performance and usability of the extension and the classic RRF in Sect. 5.3. Finally, we analyzed the obtained empirical results to conclude that even though adding more spatial consistent information to the classic RRF should have improved its performance, it did not in practice. We also observed that the classic RRFs use less RAM, require less storage space and that they produce results very quickly compared to the extension that preserved the spatial consistent information.

In the next chapter, we introduce Light Random Regression Forests (LRRFs), a variation of the classic RRF that describes the random process by describing the random variables rather than the process itself.

Science is not, despite how it is often portrayed, about absolute truths. It is about developing an understanding of the world, making predictions, and then testing these predictions.

- Brian Schmidt

6

Light Random Regression Forests

Contents

Classic Random Regression Forests (RRFs) used for multi–organ localization describe the random process of multivariate regression by storing the histograms of offset vectors along each bounding wall direction of each leaf node. We propose to eliminate the need to describe the random process by formulating the localization prediction based on the random variables that describe the random process[a]. This results in the introduction of Light Random Regression Forests (LRRFs).

In Sect. 6.1, a simple analysis shows that much of the storage requirement of a RRF is due to the storing of the offset vector histograms. Then, we make an interesting observation about finding the arg max of a distribution that consists of the addition of many 1D Gaussians in Sect. 6.2. How this observation is exploited in order to carry out multi–organ localization is explained in Sect. 6.3. The results of the study are presented with respect to the hypothesis validation, performance evaluation, and usability evaluation in Sect. 6.4.1, Sect. 6.4.2, and Sect. 6.4.3 respectively. A discussion is carried out about the obtained results in Sect. 6.5 before concluding the chapter by Sect. 6.6 with a few concluding remarks.

[a] This chapter is based on an article submitted to the 7th International Workshop on Machine Learning in Medical Imaging (MLMI 2016).

6.1 Introduction

In the previous chapter we observed the robustness of the classic Random Regression Forest (RRF) algorithm even though it does not preserve the spatial consistency. In Chap. 5, it was also demonstrated that the multi–organ localization capabilities of the classic RRFs are comparable to the so–called Hough–based Random Regression Forests (HbRRFs) that preserved the spatial consistency. Although the usability aspects of classic RRFs were many folds better than those of the HbRRF, classic RRFs still describe the *random process* of the multivariate regression by storing the 1D histograms of offset vectors along each bounding box wall direction per leaf node per organ (see Sect. 3.4.6.5 and step 6 Sect. 3.5.1).

On the one hand, the Random Access Memory (RAM) and storage requirements of classic RRFs may become exorbitantly high when such a RRF consists of many leaf nodes, but on the other hand, a large number of leaf

nodes are required for better localization. Considering certain current studies that employ RRFs for multi–organ localization, we observe that the number of leaves are in the order of magnitude of $10,000$s (see Table 6.1). In addition to storing the 1D histograms of offset vectors, the mean vector and covariance matrix of offset vectors are also stored at the leaf nodes during the training phase (see Sect. 3.4.6.5 and step 6 Sect. 3.5.1). Hence, one possible option for reducing the RAM and storage requirements of RRFs is to reduce the number of leaf nodes of the forest.

Study	Trees	Max Levels	Leaf Count
Criminisi et al. [2010]	12	7	1,536
Pauly et al. [2011]	6	8	1,536
Cuingnet et al. [2012]	7	15	229,376
Criminisi et al. [2013]	4	12	16,384
Gauriau et al. [2013]	7	12	28,672
Gauriau et al. [2014]	3	14	49,152

Table 6.1: The maximum number of leaf nodes possible in the RRFs used in the multi–organ localization literature. The studies are presented in the chronological order. We observe the current leaf node count falls in the order of magnitude of $10,000$s. (Maximum number of leaf nodes $= n_{\text{Trees}} \times 2^{\text{Max Levels}}$.)

In order to address this high storage and RAM requirements, decision jungles were proposed [Shotton et al., 2013] (see Sect. 3.3.1). A decision jungle consists of directional acyclic graphs instead of binary trees. The memory consumption is decreased by reducing the number of nodes of the jungle by introducing a node merging technique.

If we assume there are C number of organs to be localized and 64 Bytes of storage per entry (the size required to store a double float type), to store the mean vector ($C \times 6 \times 64$) and covariance matrix ($C \times 6 \times 6 \times 64$) of offset vector, a leaf node requires $2,688 \times C$ Bytes. Let us also assume that each histogram of offset vectors comprises 100 bins, although, the real number of bins may be greater than this amount. Then it requires $38,400 \times C$ Bytes ($C \times 6 \times 100 \times 64$) to store the histograms per leaf node. It is apparent that most of the storage is consumed by the histograms (roughly 14 times with the above mentioned simplifying assumptions). Decision jungles may not be the ideal solution as each individual leaf node of a decision jungle still stores

the 1D histograms of offset vectors [Shotton et al., 2013]. They therefore require the same amount of memory per leaf node.

As we have established, describing the random process results in a significant storage space requirement. In addition to that, these saved 1D histograms are loaded into the RAM during the prediction phase (also called the testing phase) in order to carry out the localization of an unseen image. This too, also translates into a substantial RAM requirement at the prediction phase.

In this chapter we present Light Random Regression Forests (LRRFs) which describe the random variables inherent to the random processes that were previously described in classic RRFs. By describing the random variables, the storage and RAM requirements are drastically reduced compared to the classic RRFs.

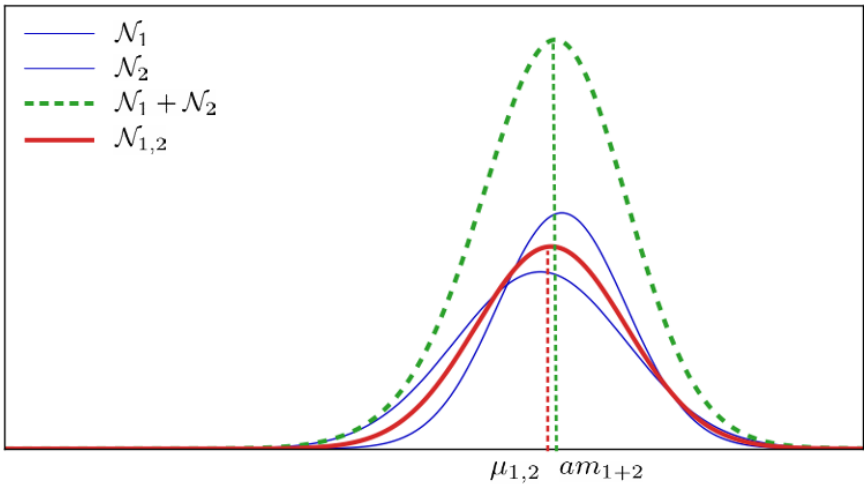
6.2 On Gaussian Distribution Summation

The foundation of LRRFs is based on the following observation on summation of 1D Gaussian distributions. A 1D Gaussian distribution ($\mathcal{G}$) is described by its mean (μ) and variance (σ^2) in the following manner:

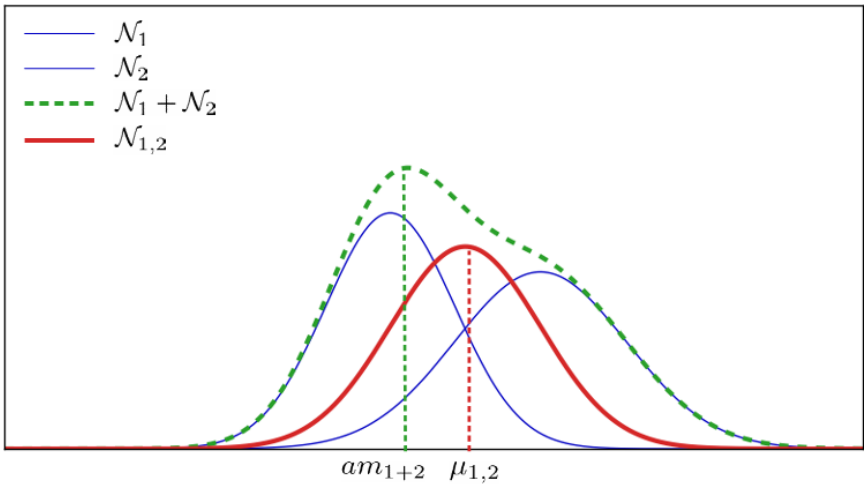
$$\mathcal{G} \sim \mathcal{N}\left(\mu, \sigma^2\right) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad . \tag{6.1}$$

A new distribution is made by summing many 1D Gaussian distributions together. The ultimate goal is to find the arg max of the final distribution. This is similar to adding many prediction distributions described by histograms together and finding the arg max of the final summed distribution in order to obtain the absolute bounding wall location in the case of multi–organ localization using RRFs (see Sect. 3.4.7.2).

Although the addition of two 1D Gaussians does not result in another 1D Gaussian, when the means of the distributions are very close, weighted means of the distributions are a good estimator of the arg max of the summation (see Fig. 6.1a). But when the means of the distributions are quite apart from each other, the weighted means are not a good estimator of the arg max (see Fig. 6.1b).



(a)



(b)

Figure 6.1: $\mathcal{N}_1$ and $\mathcal{N}_2$ are two 1D Gaussian distributions (shown in blue). $\mathcal{N}_1 + \mathcal{N}_2$ is the distribution generated by adding $\mathcal{N}_1$ and $\mathcal{N}_2$ together (shown in green). $\mathcal{N}_{1,2}$ is another Gaussian distribution constructed from $\mu_{1,2}$: the weighted means (equal weights in this case) of $\mathcal{N}_1$ and $\mathcal{N}_2$ (shown in red). The goal is to check whether $\mu_{1,2}$ is a good approximation of the arg max of $\mathcal{N}_1 + \mathcal{N}_2$ (am_{1+2}).
(a) The arg max can be closely estimated by the weighted means when the means of the original distributions are closer. (b) When the means of the original distributions are far apart, taking the arg max privileges one distribution over the other.

We propose to explore the suitability of approximating the arg max of the final bounding wall prediction histograms made up of multiple distributions as a weighted sum of the means of each individual distribution. As the theoretical soundness of the approximation may be questionable due to the assumption that the individual distributions roughly have the same mean irrespective of their variances, we validate our hypothesis before using it practically.

6.3 Materials and Method

Based on the observation presented in Sect. 6.2, we revamped the RRF algorithm to propose Light Random Regression Forests (LRRFs). The following sections describe the main changes of LRRFs compared to the classic RRFs.

6.3.1 Light Random Regression Forests

Similarly to the classic RRFs, the LRRFs are also an ensemble of Random Regression Trees (RRTs). A LRRF describes the random variables that define the random process instead of describing the random process as a classic RRF does.

Both classic RRFs and LRRFs regress the continuous conditional distribution of offset vectors $(\mathbf{d}(\mathbf{v}; c))$ as a 6D multivariate Gaussian where $\mathbf{v}$ and c denote a voxel and an organ respectively. Consequently, the 6D multivariate Gaussian results in 6 1D univariate Gaussians. The empirical evidence gathered in Chap. 5 shows that the directional independency hypothesis does not degrade the final localization results compared to a setting in which the spatial consistency of offset vectors is preserved.

The training phase of LRRFs is identical to the training phase of classic RRFs [Criminisi et al., 2013; Criminisi and Shotton, 2013] except for one simplification described in Sect. 6.3.2. The details of the prediction phase of LRRFs are presented in Sect. 6.3.3.

6.3.2 Training Phase Differences

The split node training is carried out in the usual manner by maximizing an information gain measure as mentioned in Sect. 3.4.6.3. The split node

training is stopped either when maximum tree depth (D) is reached or when the number of voxels reaching a node is lesser than a threshold ($n_{\min} = 25$).

When all the split nodes are trained, the leaf nodes contain all the accumulated voxels. Training of a given leaf node consists of summarizing the offset vectors of these accumulated voxels. The classic RRFs train the leaf node by storing the random process using histograms of the offset vectors [Criminisi et al., 2013; Criminisi and Shotton, 2013]. Going a step further, we propose to train the leaf nodes by only storing the mean ($\bar{\mathbf{d}}_{\text{dir}}$) and the variance ($\sigma^2_{\text{dir}}$) of these 1D distributions; i.e., the random variables that define the distribution $\mathcal{D}$:

$$\mathcal{D} \sim \mathcal{N}\left(\bar{\mathbf{d}}_{\text{dir}}, \sigma^2_{\text{dir}}\right) \ , \tag{6.2}$$

where dir $\in \{r, l, a, p, i, s\}$ (r = right, l = left, a = anterior, p = posterior, i = inferior, and s = superior), $\bar{\mathbf{d}}_{\text{dir}} \in \{\bar{\mathbf{d}}_r, \bar{\mathbf{d}}_l, \bar{\mathbf{d}}_a, \bar{\mathbf{d}}_p, \bar{\mathbf{d}}_i, \bar{\mathbf{d}}_s\}$ and $\sigma^2_{\text{dir}} \in \{\sigma^2_r, \sigma^2_l, \sigma^2_a, \sigma^2_p, \sigma^2_i, \sigma^2_s\}$. Consequently, LRRFs do not have to store the 1D histograms of each wall direction of each organ. This saves a lot of storage space. Hence storing of the offset vector distributions is omitted in LRRFs compared to the classic RRFs. This means that we model the random variables instead of the random process.

6.3.3 Prediction Phase Differences

During the prediction phase, all selected voxels ($\mathbf{v}$) of a previously unseen image ($\mathcal{V}$) are pushed through each RRT until they are all accumulated at some leaf nodes per RRT of the LRRF ($l(\mathbf{v})$). The set of leaf nodes that accumulates at least 75% of voxels ($\mathcal{L}_t$) and that displays the lowest variability per each organ are used for the localization prediction per each RRT. Then, each bounding wall distribution $\left(p\left(\mathbf{b}_c\right)\right)$ can be defined as:

$$p(\mathbf{b}_c) = \sum_{t=1}^{T} \sum_{l \in \mathcal{L}} p(\mathbf{b}_c \mid l) p(l) \ , \tag{6.3}$$

where $p(l) = |l(\mathbf{v})| \ / \ \sum_{l(\mathbf{v}) \in \mathcal{L}_t} |l(\mathbf{v})|$, T is the number of trees, $|l(\mathbf{v})|$ is the number of voxels in the leaf node l, c is the organ and $p(\mathbf{b}_c \mid l)$ can be derived from the saved mean offset vectors $\left(\bar{\mathbf{d}}(\mathbf{v}; c)\right)$ as:

$$\bar{\mathbf{b}}_c(\mathbf{v}) = \hat{\mathbf{v}} - \bar{\mathbf{d}}(\mathbf{v}; c) \ , \tag{6.4}$$

where $\hat{\mathbf{v}} = (\mathbf{v}_x, \mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_y, \mathbf{v}_z, \mathbf{v}_z)$ made from the voxel position $(\mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_z)$.

From Eq. (6.2), this translates into an addition of weighted 1D Gaussians. The final prediction of the absolute position of each bounding wall direction $\left(\hat{\mathbf{b}}_c\right)$ is the arg max of $p(\mathbf{b}_c)$, i.e.:

$$\hat{\mathbf{b}}_c = \arg\max_{\mathbf{b}_c} \; p(\mathbf{b}_c) \;\; . \tag{6.5}$$

In classic RRFs, $p(\mathbf{b}_c)$ is modeled by histograms per each bounding wall direction (i.e., by mimicking the random process). Then, $\hat{\mathbf{b}}_c$ is obtained by finding the mode of the histogram [Criminisi et al., 2013; Criminisi and Shotton, 2013].

Since each selected leaf node should predict approximately the correct position, means of each leaf node bounding box vectors $p(\mathbf{b}_c \mid l)$ should be relatively close to one another along each bounding wall direction. The likelihood of the above assumption being true is higher as only the high confident leaf nodes are chosen for the final prediction as mentioned above. The mean leaf node bounding box vector $(\bar{\mathbf{b}}_{c,l})$ is defined as:

$$\bar{\mathbf{b}}_{c,l} = \hat{\bar{\mathbf{v}}} - \bar{\mathbf{d}}(\mathbf{v}; c) \;\; , \tag{6.6}$$

where $\hat{\bar{\mathbf{v}}} = (\bar{\mathbf{v}}_x, \bar{\mathbf{v}}_x, \bar{\mathbf{v}}_y, \bar{\mathbf{v}}_y, \bar{\mathbf{v}}_z, \bar{\mathbf{v}}_z)$ made from the mean voxel position $(\bar{\mathbf{v}}_x, \bar{\mathbf{v}}_y, \bar{\mathbf{v}}_z)$ of the voxels accumulated at the leaf node l. As a direct consequence of the observation made in Sect. 6.2, we approximate the arg max of $p(\mathbf{b}_c)$ by calculating the sum of the means of $p(\mathbf{b}_c \mid l)$ weighted by $p(l)$ as:

$$\hat{\mathbf{b}}_c = \sum_{t=1}^{T} \sum_{l \in \mathcal{L}_t} \bar{\mathbf{b}}_{c,l} \, p(l) \;\; , \tag{6.7}$$

without describing $p(\mathbf{b}_c)$ using histograms. We replace Eq. (6.5) by Eq. (6.7). As no random process is mimicked by the use of histograms but a model of $\hat{\mathbf{b}}_c$ is directly built using the random variables, LRRFs lead to faster computational times.

6.3.4 Materials and Implementation Details

The dataset described in Sect. 4.2 was used for the study and was randomly divided into a training set and a prediction set of 55 and 45 images respectively, similarly to [Criminisi et al., 2010]. The same set of 9 organs

described in Sect. 4.2.2 were used in this study too. The same statistical test mentioned in Sect. 4.3.2.4 was used to verify the difference between the results obtained by the two algorithms. For all statistical tests carried out, α was set to 0.01. The tests were carried out under the same hardware configuration as described in Sect. 4.3.1.

A classic RRF was trained and used for prediction as described in Sect. 3.5.2. Then, the classic RRF was transformed into a LRRF by removing the stored 1D histograms of offset vectors per each wall direction as mentioned in Sect. 6.3. This enabled us to compare the two methods under the exact same conditions.

We generated three results sets from classic RRFs (R_{Ci}) and from LRRFs (R_{Li}); namely:

- R_{C1} and R_{L1} with 4 RRTs and $D = 12$ similar to Criminisi et al. [2013]; Criminisi and Shotton [2013],
- R_{C2} and R_{L2} with 4 RRTs and $D \in [1, 2, \ldots, 17]$,
- R_{C3} and R_{L3} with $[1, 2, \ldots, 20]$ RRTs and $D = 12$,

to evaluate the performance and usability of both methods.

6.3.4.1 Validation of Our Hypothesis

In order to evaluate the hypothesis of estimating the arg max of summation of 1D Gaussians by the weighted summation of their means as presented in Sect. 6.2, the following steps were carried out. First, the distribution pairs of arg max of $p(\mathbf{b}_c)$ and weighted means of $p(\mathbf{b}_c \mid l)$ denoted respectively by $\mathcal{D}_a$ and $\mathcal{D}_\mu$ were created for each wall direction per organ (6×9 distribution sets) using all 45 prediction images. Finally, each pair of $\mathcal{D}_a$ and $\mathcal{D}_\mu$ were subjected to Wilcoxon–Mann–Whitney test.

6.3.4.2 Prediction Precision

Prediction precision evaluation was carried out using mean Bounding Wall Prediction Error (BWPE), Centroid–Hit Measure (CHM), and Centroid Error (CE) as mentioned in Sect. 4.3.2. Errors were calculated only for the organs that were fully present in the CT volumes.

6.3.4.3 Usability

The usability was evaluated using the mean time (t) and the mean amount of RAM (M) required to perform a single multi–organ localization in a CT image. The amount of disk space used to store the forest (S) was also compared.

6.4 Results

The results obtained by classic RRF and LRRF were evaluated under the three categories mentioned above (hypothesis, precision, and usability).

6.4.1 Hypothesis Verification

Wilcoxon–Mann–Whitney test failed to reject the null hypothesis of $\mathcal{D}_a$ and $\mathcal{D}_\mu$ coming from the same underlying distribution with $p-$value $\in [0.04 - 0.98]$ for all 54 verification distribution pairs of $\mathcal{D}_a$ and $\mathcal{D}_\mu$. This provides strong empirical evidence that the arg max of summation of 1D Gaussians can be readily estimated by the weighted summation of their means in the context of multi–organ localization using RRFs. The total 54 $p-$values obtained are presented in Table 6.2.

Organ	Left	Right	Ante-rior	Pos-terior	Infe-rior	Supe-rior
Left Kidney	0.47	0.98	0.73	0.22	0.15	0.56
Right Kidney	0.78	0.60	0.94	0.94	0.18	0.83
Liver	0.24	0.63	0.95	0.83	0.21	0.51
Spleen	0.62	0.39	0.75	0.39	0.12	0.18
Left Femur Head	0.75	0.92	0.94	0.98	0.11	0.56
Right Femur Head	0.82	0.94	0.97	0.96	0.04	0.23
Left Pelvis	0.59	0.75	0.85	0.88	0.56	0.63
Right Pelvis	0.88	0.51	0.78	0.78	0.41	0.60
L5	0.80	0.97	0.97	0.91	0.70	0.27

Table 6.2: The p–values obtained for Wilcoxon–Mann–Whitney test for the distributions of real arg max ($\mathcal{D}_a$) and estimated arg max using weighted means of 1D Guassians ($\mathcal{D}_\mu$).

Observing the table we note that the $p-$values obtained take high values for all directions apart for the inferior direction.

6.4.2 Prediction Precision Evaluation

Results for R_{C1} and R_{L1}

A summary of the mean BWPEs ($Er_{1,1}$) of all 45 predictions obtained for R_{C1} and R_{L1} are presented in Table 6.3. Analyzing the values themselves do not lead to a clear conclusion as in some cases R_{C1} appear to have better localization results while in some cases R_{L1} appear to do better.

In order to access the mean BWPEs of individual localization predictions, the mean BWPE obtained for R_{C1} and R_{L1} for each prediction image are presented in 1.5 Inter Quartile Range (IQR) box plots in Fig. 6.2. The obtained $Er_{1,1}$ failed to reject the null hypothesis that both error measures of R_{C1} and R_{L1} for all organs originated from the same underlying distribution with $p-$value $\in [0.03 - 0.78]$. The $p-$values obtained for all 9 organs are presented in Table 6.4. These values indicate that the error is statistically the same for the two methods.

The centroid–hit measures obtained for the localization predictions are presented in Table 6.5. The centroid–hit measure is presented as a fraction between the centroid–hits and all predictions since some predictions were not considered for the error measure calculation as the organs were not fully present in the prediction image. Both R_{C1} and R_{L1} appear to produce similar centroid–hit measures. Additionally, we observed that centroid–hit measure was the highest in right to left direction whereas it was the lowest in inferior to superior direction.

The centroid errors ($Er_{1,2}$) of individual localization predictions are presented in 1.5 IQR box plots in Fig. 6.3. We observed that the inter quartile ranges of R_{L1} are smaller than those of R_{C1} for all organs except the right pelvis.

The obtained $Er_{1,2}$ failed to reject the null hypothesis that both error measures of R_{C1} and R_{L1} for all organs originated from the same underlying distribution with $p-$value $\in [0.03 - 0.96]$. All the $p-$values of the statistical significance test of the two result sets are shown in Table 6.6. We observed that the liver produced the smallest $p-$value for $Er_{1,1}$ (0.03) and the largest $p-$value for $Er_{1,2}$ (0.96).

	Left Kidney	Right Kidney	Liver	Spleen	Left Femur Head	Right Femur Head	Left Pelvis	Right Pelvis	L5
Mean R_{C1}	11.32	12.10	14.42	14.33	7.17	6.81	6.39	6.78	7.60
Mean R_{H1}	11.52	10.98	15.82	14.84	7.67	7.42	7.50	7.43	8.70
Std. dev R_{C1}	10.86	10.12	13.76	13.23	6.35	6.42	6.42	4.78	6.44
Std. dev R_{H1}	10.15	8.85	12.66	12.87	7.01	6.96	6.55	5.55	7.82
Median R_{C1}	8.09	10.00	10.00	10.45	6.00	4.80	4.35	6.43	6.00
Median R_{H1}	9.81	9.53	12.79	12.25	6.06	5.85	5.54	6.10	6.70
Max R_{C1}	82.0	64.0	80.3	73.0	32.0	34.5	34.0	19.4	33.0
Max R_{H1}	86.3	63.3	70.4	80.2	46.4	45.8	30.9	21.5	58.7

Table 6.3: Summary of mean BWPEs for R_{C1} and R_{L1} in mm.

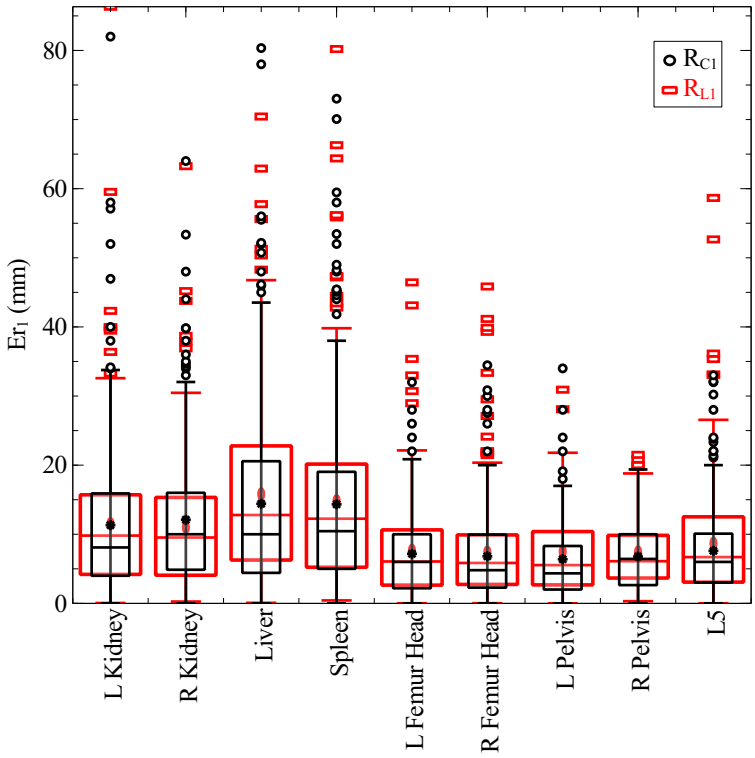


Figure 6.2: 1.5 IQR box plots of mean BWPE of each prediction image for R_{C1} and R_{L1}.

Organ	$p-$value
Left Kidney	0.46
Right Kidney	0.35
Liver	**0.03**
Spleen	0.38
Left Femur Head	0.41
Right Femur Head	0.14
Left Pelvis	0.17
Right Pelvis	0.78
L5	0.13

Table 6.4: Obtained $p-$values for the mean BWPEs for R_{C1} and R_{L1}. Note the very small $p-$value for the liver.

Results for R_{C2} and R_{L2}

The mean BWPEs (Er_2) of all organs obtained were in the range of [9.85 -16.53] for R_{C2} and in the range of $[9.99 - 16.23]$ for R_{L2}. The standard

Organ	Right to Left		Anterior to Posterior		Inferior to Superior	
	R_{C1}	R_{L1}	R_{C1}	R_{L1}	R_{C1}	R_{L1}
Left Kidney	1.00	1.00	1.00	1.00	0.89	**0.93**
Right Kidney	1.00	1.00	1.00	1.00	0.98	0.98
Liver	1.00	1.00	1.00	1.00	1.00	1.00
Spleen	1.00	1.00	1.00	1.00	1.00	1.00
Left Femur Head	1.00	1.00	0.96	**0.98**	0.91	0.91
Right Femur Head	1.00	1.00	0.98	**1.00**	0.98	0.98
Left Pelvis	1.00	1.00	1.00	1.00	1.00	1.00
Right Pelvis	0.98	0.98	0.93	**0.96**	0.96	**0.98**
L5	1.00	1.00	1.00	1.00	**0.93**	0.91

Table 6.5: Centroid–hit measure values for R_{C1} and R_{L1}. The measures are given as a fraction between centroid–hits and all predictions. The bold values indicate when centroid–hit measure was better for one method than for the other.

Organ	$p-$value
Left Kidney	0.58
Right Kidney	0.53
Liver	**0.96**
Spleen	0.89
Left Femur Head	0.60
Right Femur Head	0.38
Left Pelvis	0.20
Right Pelvis	0.57
L5	0.03

Table 6.6: Obtained $p-$values for the centroid errors for R_{C1} and R_{L1}. Note the very high $p-$value for the liver.

deviations of R_{C2} and R_{L2} were between $[10.21 - 15.17]$ and $[9.59 - 14.23]$ respectively. Er_2 decreased with the number of decision levels for both R_{C2} and R_{L2}. Er_2 are presented in Fig. 6.4. We observe Er_2 decreasing with the number of decision levels for both R_{C2} and R_{L2}. The two error curves appear to be converging at 17 decision levels.

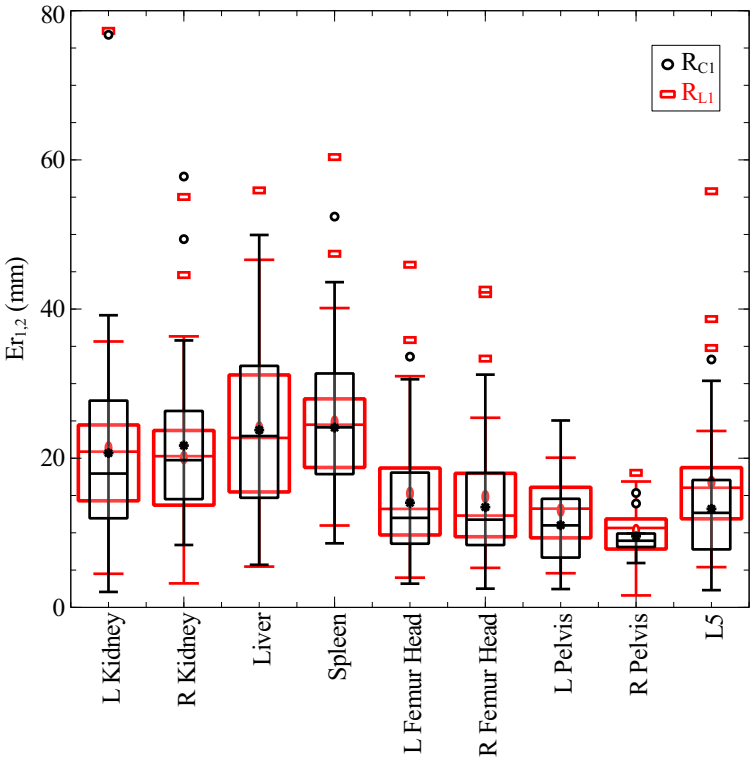


Figure 6.3: 1.5 IQR box plots of centroid errors of each prediction image for R_{C1} and R_{L1}.

Results for R_{C3} and R_{L3}

Similarly, the mean BWPEs (Er_3) of all organs obtained were in the range of $[10.25 - 10.39]$ for R_{C3} and in the range of $[10.89 - 10.98]$ for R_{L3}. The standard deviations of R_{C3} and R_{L3} were between $[10.21 - 10.73]$ and $[10.08 - 10.19]$ respectively. Er_3 did not decrease a lot with the number of trees.

6.4.3 Usability Evaluation

The metrics t_2, M_2, and S_2 for R_{C2} and R_{L2} are presented in Fig. 6.6. Both t_2 and S_2 for R_{C2} grow exponentially compared to the same values of R_{L2}. For 17 decision levels, t_2, M_2, and S_2 are $(2.2\,\text{s}, 19.5\,\text{s})$, $(117\,\text{MB}, 1147\,\text{MB})$, and $(171\,\text{MB}, 5221\,\text{MB})$ for R_{L2} and R_{C2} respectively. Hence for 17 decision levels, LRRF is approximately 9 times faster, takes about 10 times less RAM, and uses about 30 times less storage space compared to the classic RRF.

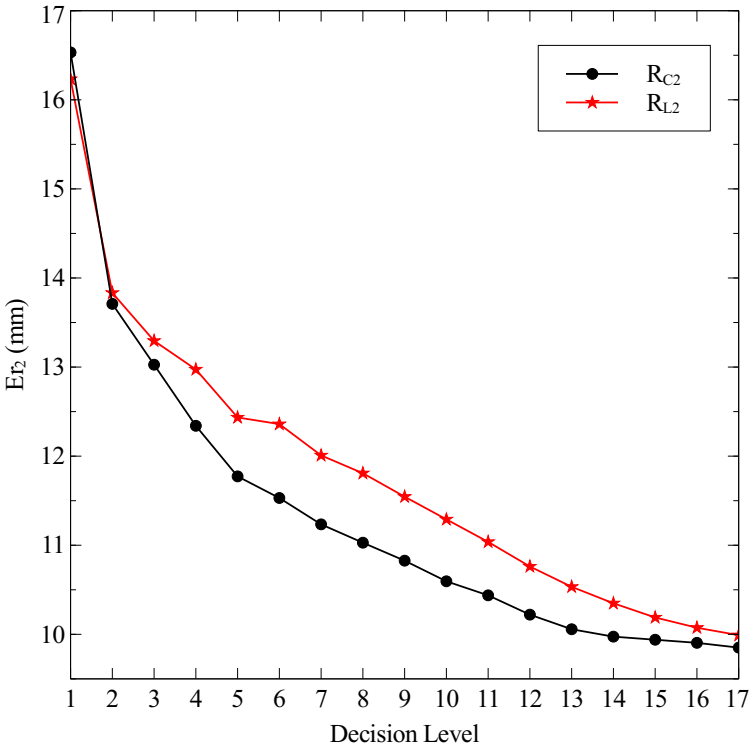


Figure 6.4: Mean BWPEs of all organs obtained for R_{C2} and R_{L2}.

Similar to the previous case, t_3, M_3, and S_3 for R_{C3} and R_{L3} are presented in Fig. 6.7. All three metrics increase proportionally with the number of trees. However, the rate of increase is much slower for R_{L3} compared to R_{C3}. For 20 trees, t_3, M_3, and S_3 are (6.2 s, 29.3 s), (164 MB, 2226 MB), and (34 MB, 4440 MB) for R_{L3} and R_{C3} respectively. This implies that for 20 RRTs the localization using LRRFs is roughly about 5 times faster, takes about 14 times less RAM, and uses about 130 times less storage space compared to using classic RRFs.

6.5 Discussion

As demonstrated in Sect. 6.4.2, for all 9 organs, the LRRFs and classic RRFs provide similar prediction results. As calculation of a mean position of a set of voxels is much faster than the accumulation of histograms per each voxel, LRRFs produce results very quickly compared to classic RRFs. All the above provided evidence suggest that LRRFs have much better usability

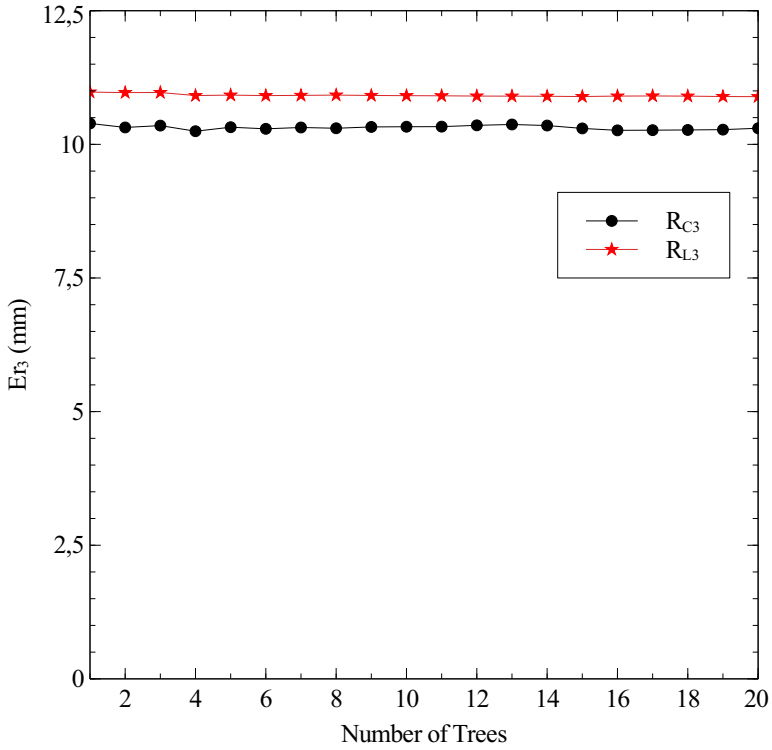


Figure 6.5: Mean BWPEs of all organs obtained for R_{C3} and R_{L3}.

criteria than classic RRFs with similar prediction capabilities.

It is interesting to notice the difference between the mean BWPE and the centroid error. The mean BWPE averages the absolute differences between the 6 bounding walls of the predicted bounding box and the gold standard (see Sect. 4.3.2.1) whereas the centroid error reports the euclidean distance between the centroid of the predicted bounding box and the gold standard (see Sect. 4.3.2.3). The two error measures calculated for the liver using the same prediction results sets R_{C1} and R_{L1} provide an excellent example. The distribution of mean BWPE between R_{C1} and R_{L1} for the liver was the least similar among the 9 organs (with a $p-$value of 0.03) while the distribution of centroid error between R_{C1} and R_{L1} for the liver was the most similar among the 9 organs (with a $p-$value of 0.96).

The number of significant outliers of R_{L1} is greater than that of R_{C1} for relatively small bone structures (left femur head, right femur head, and L5 vertebra) for $Er_{1,1}$ (see Fig. 6.2). The same observation is made concerning $Er_{1,2}$ as well (see Fig. 6.3). This may suggest that LRRFs are less powerful

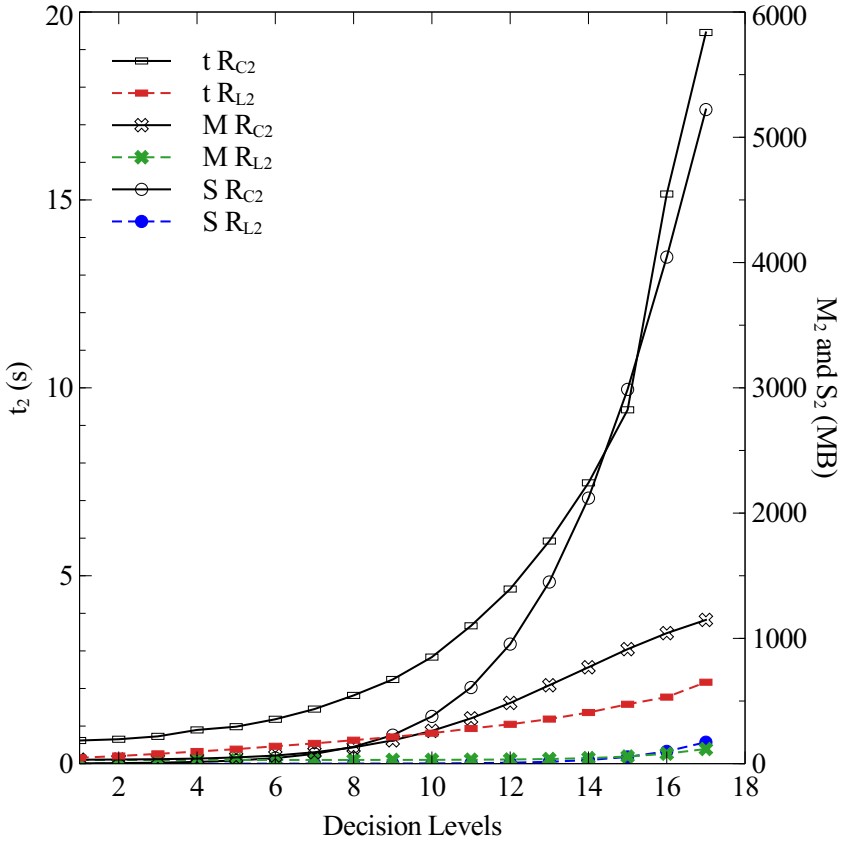


Figure 6.6: Evolution of t_2, M_2, and S_2 with number of decision levels for R_{C2} and R_{L2}.

at localizing smaller organs. But interestingly, the centroid–hits measures for left femur head and right femur head of R_{L1} are slightly better than those obtained for R_{C1} (see Table 6.5). This may imply that LRRFs may be more robust (for example it can be better in determining the best place for the seeds in region growing methods).

Increasing the number of trees did not have a big impact on the localization results (see Fig. 6.5). In contrast, increasing the number of decision levels did (see Fig. 6.4); which emphasizes the importance of having deeper RRTs. However, from R_{C2} and R_{L2}, it is apparent that, as the number of decision levels increases, the feasibility of classic RRFs for multi–organ localization becomes questionable.

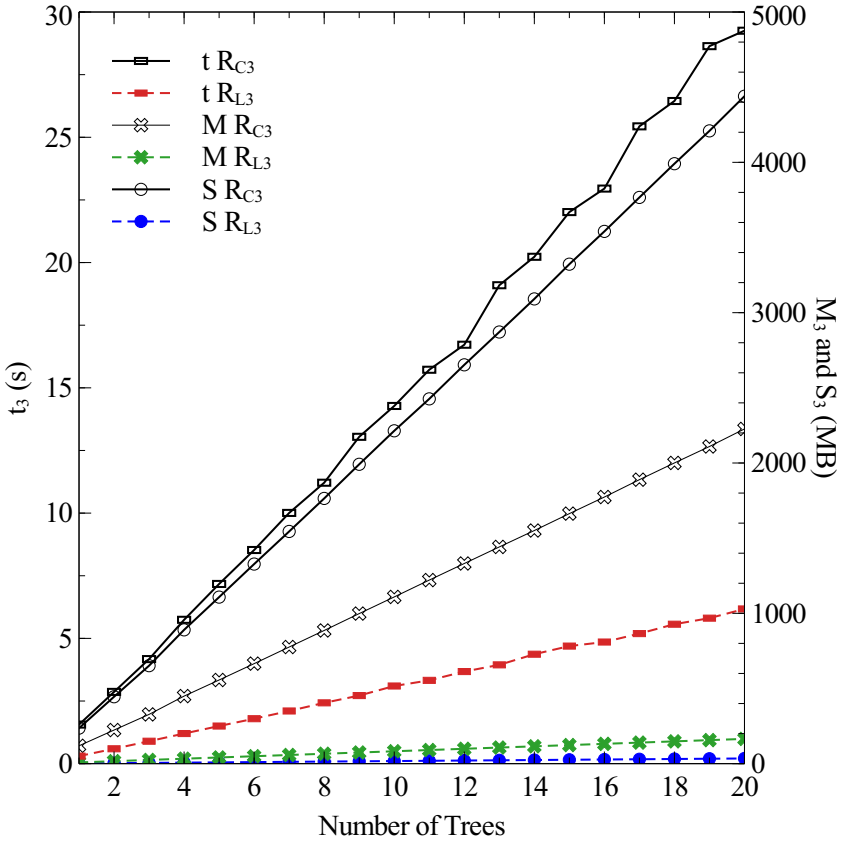


Figure 6.7: Evolution of t_3, M_3, and S_3 with number of trees for R_{C3} and R_{L3}.

6.6 Conclusion

We propose LRRFs as an alternative to classic RRFs that only describes the random variables that are inherent to the random process which results in a huge RAM and storage requirement reduction enabling the growth of random forests having deeper trees.

Although estimating the arg max of summation of 1D Gaussians by weighted individual means might be questionable (as the sum of 1D Gaussians is not a Gaussian distribution), the empirical results we obtained indicate that they are statistically alike with respect to the mean bounding wall prediction error and centroid error. Additionally, this estimation results in huge gains in speed and memory, opening opportunities for newer kind of RRF.

In this chapter, we proposed a variation of classic Random Regression Forests (RRFs) termed Light Random Regression Forests (LRRFs). In order to produce multi–organ localization predictions, LRRFs use the random variables that define the random process instead of describing the random process as classic RRFs do.

We observed that the arg max of a distribution made out by summing many 1D Gaussians can be readily estimated by the weighted means of individual 1D Gaussians if the means are relatively close to one another.

During the prediction phase of the RRF algorithm, the leaf nodes that participate in the final prediction localization are the ones that are most confident about their predictions. This is ensured by sorting the leaf nodes per organ according to the determinant of the covariance matrix of the offset vectors stored from the training phase. Hence, we assumed that the prediction of the selected leaf nodes point roughly to the same location enabling us to employ the observation mentioned previously.

The empirical evidence gathered during the study on the precision of the LRRF provided comparable results to the classic RRF while the usability criteria of LRRFs outperformed the classic RRFs by many folds. The reduction in the execution time may open the possibility of LRRFs for more time critical applications. The large reductions in storage and RAM of LRRFs enable the growth of random forests comprising deeper trees and allows a wider spread of devices where it can be integrated (*e.g.*, smart phones or other mobile devices).

In the next chapter, we are going to propose a consistent and automatic selection criteria for three important parameters of the RRF induction process depending on the perceived useful information that each image voxel may possess.

Strive for continuous improvement, instead of perfection.

- Kim Collins

7

Toward Automatic and Consistent Parametrization

Contents

How relevant voxels are selected from the images for the training and prediction phases and how many forest leaf nodes are picked for the final organ localization prediction are important parameters that affect the final localization capabilities and usability measures of the method.

The Sect. 7.1 introduces a variety of voxel selection choices available in the literature, ranging from using random voxels or selecting a region of interest whereas an arbitrary fixed amount of leaf nodes are picked for final prediction. The approach followed in order to reduce the dependency on arbitrary parameters in selecting the amount of voxels used in training and prediction phases as well as the number of leaf nodes used for final localization prediction in presented in Sect. 7.2. The results of the study are presented in Sect. 7.3. The obtained results will be discussed in Sect. 7.4 before concluding the chapter by Sect. 7.5 with a few concluding remarks and perspectives.

7.1 Introduction

During the analysis of Random Regression Forests (RRFs) that was carried out in Chap. 3, we observed that the number of image voxels that participates in three steps of the algorithm was decided *a priori* by the user: 1) the number of voxels that is used in training, 2) the number of voxels that is pushed through the whole forest at the prediction phase, and 3) the number of voxels that is used for the final prediction calculation (see Sect. 3.4.5 and Sect. 3.4.7.1). The values employed by various studies are presented in the Table 7.1.

As mentioned in the Sect. 3.4.5, the use of too fewer number of voxels may decrease the localization capabilities of the RRF algorithm as overfitting may occur. Although the use of too many voxels may not decrease the localization capabilities it may affect the training and prediction times. Hence, the choice of the *correct* number of voxels is an important criterion for the balance between the localization capabilities and usability measures of the method.

To the best of our knowledge, no information is found in the literature describing how to choose these arbitrary values selected *a priori* by the user other than stating that the values are application specific. This may suggest the use of trial and error methods or the use of range of values per

Study	Used in training	Pushed through the forest	Final % in prediction (p_v)
Criminisi et al. [2010]	all	all	1%
Pauly et al. [2011]	all	all	n/a
Cuingnet et al. [2012]	n/a	n/a	n/a
Criminisi et al. [2013]	in a regular grid of ±10 cm from center of each axial slice	in a regular grid of ±10 cm from center of each axial slice	75%
Gauriau et al. [2013]	random subset of 40000 voxels	random subset	n/a
Gauriau et al. [2014]	random subset of 30000 voxels	random subset of 30000 voxels	3%

Table 7.1: The number of voxels used for training, pushed through the forest at the prediction phase, and used for the final prediction calculation. These values that were previously presented in Table 3.3 and Table 3.5 are reproduced here for convenience. (n/a: information not available)

each variable and selecting the value combination that produces the *best* results on the training or the prediction set. Consequently, the use of such arbitrary "tuned" values may discourage the wider acceptance of RRFs as a general tool for automatic multi–organ localization. Therefore, it may be interesting to look for a mechanism to replace these arbitrary values by automatic selection procedures.

In this chapter we aim at reducing the dependency on these arbitrary parameters and propose two consistent alternatives in order to obtain a better generalization and encourage the usage of RRFs in the field of medical image analysis. The two proposals comprise:

1. a generic method to reduce the influence of the background noise and
2. an automatic method for picking the leaf nodes used for the final organ localization prediction based on the first proposal.

7.2 Materials and Method

The next sections describe the approach followed in order to reduce the dependency on arbitrary parameters in selecting the number of leaf nodes used for final localization prediction. First, we look at how the influence of the background noise can be reduced using a simple automatic image processing operation without using any ad hoc parameters. Then, the final leaf node selection criterion is developed based on the same image processing operation.

7.2.1 Reducing The Influence of The Background Noise

Given a Computed Tomography (CT) image, most of the background consists of air whereas the foreground mainly consists of the patient's body and part of the examination table. As a RRF is searching for good landmarks in order to localize anatomical structures, it is unlikely that those landmarks are located in the background [Criminisi and Shotton, 2013]. Consequently, choosing the voxels that belong to the body of the patient can be considered as a logical selection.

On the other hand, using a predefined constant region may prove too restrictive as patient morphological differences are extremely large. An example is provided in Fig. 7.1b where voxels belonging to the grid of ± 10 cm from the center of each axial slice are used in training and prediction similar to Criminisi et al. [2013]. In such a scenario the voxels are originated from a very restrictive region. Randomly selecting a fixed amount of voxels as employed in Gauriau et al. [2013, 2014] would include voxels from the background that may contain irrelevant information due to the noise (see Fig. 7.2a).

We made the observation that the background was not uniform (i.e., there is noise in the background too). We could observe the noise present in the background at a very low Hounsfield Units (HUs) (see Fig. 7.2a). At the training phase, a region purely consisting of the background voxels may be further split into two more regions due to this noise. On the contrary, if the region is uniform, it may not be split further (see Fig. 7.2b).

Due to the above mentioned observations, instead of either using a fixed ROI of a given arbitrary dimension for all images similar to [Criminisi et al., 2013] or randomly selecting an arbitrary number of voxels per image sim-

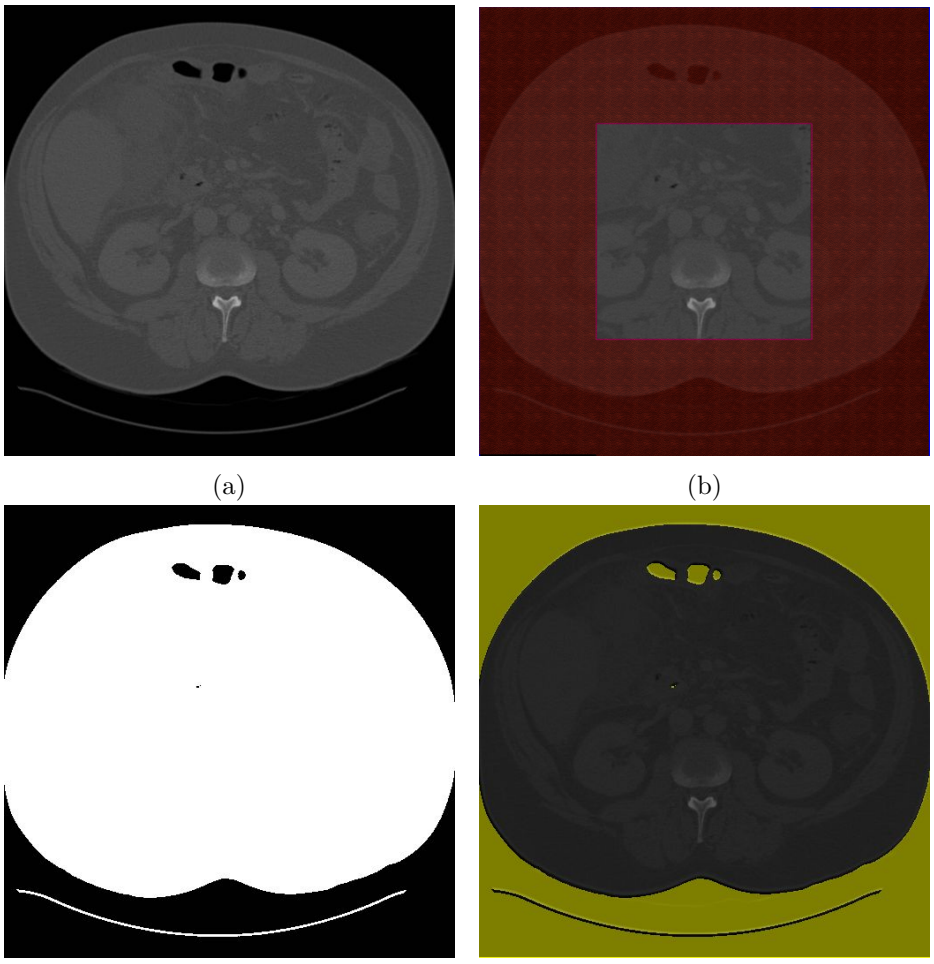
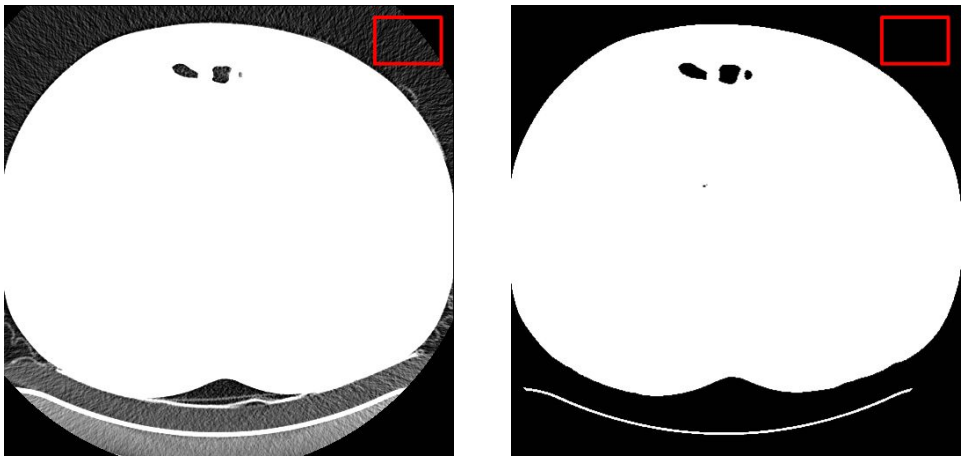


Figure 7.1: Different Region of Interests (ROIs) of an axial slice of a CT image (Original images without any contrast adjustments). (a) Original image slice. (b) The centered unmasked square depicts the ROI (±10 cm grid from the center of the slice) that was used in Criminisi et al. [2013]. The masked area shown in red (which is not considered in the training and prediction phases) contains a big portion of the patient's body. (c) Image slice after applying the Otsu thresholding. White roughly represents the foreground whereas black roughly represents the air-like regions of the slice. (d) Image slice after unifying the air-like regions using the mask created by Otsu thresholding (shown in yellow). The masked area represents the unified regions.

ilar to [Gauriau et al., 2013, 2014], we decided to choose all the voxels of the image after removing the effect of the noise of the background voxels.



(a) (b)

Figure 7.2: The effects of the background noise on the axial slice used in Fig. 7.1. Red rectangle represent the background area to be split. (a) At a very low HUs, the background noise could be observed. Consequently, a pure background region may be further split due to the noise present. (b) When the background is transformed into a uniform region by setting all the voxels belonging to the background to the same HU value, then, a pure background region can not be further split.

Attention, we do not remove the background itself as it may be relevant, only its noise.

Every training and testing image was subjected to an Otsu threshold Otsu [1975] which resulted in a binary image (see Fig. 7.1c). The Otsu thresholding method is based on statistics that separates the image into 2 voxel clusters (one with information and the other with background) and automatically chooses the optimal thresholding value. This optimal thresholding value is image dependent. The resulting binary image was then used as a mask to the original image to generate a new image where the zero valued voxels were set to an arbitrary zero (origin) value (see Fig. 7.1d). In the case of CT, we chose the value to be -1000 Hounsfield Unit (HU) which corresponds to air [Mah et al., 2010]. Values of the rest of the voxels were not modified. By unifying all the air–like regions to have the same value, the eventual effect of the noise of these regions may have on the localization is minimized.

Although all voxels of the training and testing images were used in the training and prediction phases respectively, the influence of the air–like re-

gions was expected to be minimal. In addition to that, this voxel discrimination is used to select the voxel percentage that participate in the ultimate bounding box prediction (p_v) in the prediction phase as described below.

Then the usual steps of the training phase are carried out in order to train a RRF (see Sect. 3.4.6).

7.2.2 Picking The Leaf Nodes Used for Organ Localization Prediction

During the prediction phase, once the unseen CT image is preprocessed to unify the air–like regions as mentioned in the previous section, all the voxels of the image are pushed through the forest. Once all selected voxels have been pushed through the forest, the leaf nodes that accumulated these voxels are sorted with respect to the covariance of the offset vectors accumulated during the training phase for each organ. This sorting step is intuitive as smaller covariance implies higher confidence in localization. Then the number of leaf nodes (n_p) that participate in the prediction among the selected voxels should be determined for each anatomical structure. To the best of our knowledge, selecting an arbitrary percentage of the selected voxels (p_v), i.e., an arbitrary number of leaf nodes, is the only method proposed in the literature [Criminisi et al., 2013; Gauriau et al., 2013, 2014] (see column 4 Table 7.1). As it will be demonstrated in Sect. 7.3.1 there is no single p_v (hence, n_p) that would always produce optimal localization.

Following our previous assumption that the voxels belonging to the non–air–like regions carry most of the useful information, our aim was to select the percentage of voxels that corresponds to these regions. This percentage (p_v) was calculated by estimating the *separative power* of the Otsu threshold:

$$p_v = \frac{n_{\mathcal{V}_f}}{n_{\mathcal{V}}} \times 100\,\% \ , \tag{7.1}$$

where, $n_{\mathcal{V}_f}$ is the number of voxels that belong to the foreground found after the Otsu threshold operation and $n_{\mathcal{V}}$ is the total number of voxels in the image. Finally, n_p is the number of leaf nodes at the head of the sorted queue that accumulates at least p_v percentage of voxels.

The rest of the prediction details are identical to the steps described in Sect. 3.4.7.

7.2.3 Materials and Implementation Details

In order to verify the influence of background noise unification and automatic selection of p_v on the RRF performance, we followed the methodological procedure described in Chap. 4.

The dataset described in Sect. 4.2 was used for the study and was randomly divided into a training set and a prediction set of 55 and 45 images respectively, similarly to [Criminisi et al., 2010]. The same set of organs described in Sect. 4.2.2 were used in this study too. Prediction evaluation was carried out using mean Bounding Wall Prediction Error (BWPE) as mentioned in Sect. 4.3.2. The same statistical test mentioned in Sect. 4.3.2.4 was used to verify the difference between the results obtained by the two algorithms. For all statistical test carried out, α was set to 0.01. The tests were carried out under the same hardware configuration as described in Sect. 4.3.1.

The classic RRF method was implemented as described in Criminisi et al. [2013] and is denoted by I^1. The second implementation incorporated the two changes proposed in this study and is denoted by I^2. Both I^1 and I^2 comprised 4 regression trees and 12 maximum decision levels following the benchmark protocol mentioned in Sect. 4.3 and the study presented in Chap. 6 that showed the convergence is nearly obtained for these values. As proposed in Criminisi et al. [2013], a subset of voxels of each image volume was used for training and prediction phases in I^1. This subset was chosen by sampling on a regular grid within ± 10 cm of the center of each axial slice of each image volume (see Fig. 7.1b). In contrast, all voxels belonging to an image volume (see Sect. 7.2.1) were used in I^2 for training and prediction phases.

7.3 Results

The goal of this section is two fold. First, we wish to prove that there is no optimal arbitrary number of leaf nodes that produce the *best* localization results. The second goal is to provide empirical evidence that the automatic selection of the number of leaf nodes produces as good results as the state of the art method does.

7.3.1 No Optimal Number of Leaf Nodes for Prediction

First, the 55 images used in the training phase (the training set Ω_T) were again applied to the trained RRF obtained using I^1 to predict the localization of the 9 anatomical structures (R_T^1). Through R_T^1 we wanted to figure out whether an optimal p_v value could be determined *a priori* (i.e., before the prediction phase).

For each image used for training and for each organ 100 different predictions were made using 100 different image voxel percentages $p_v \in [1 - 100]$ following the procedure described in Sect. 3.4.7 and Sect. 3.5.2. When an anatomical structure of interest was fully present in an image, BWPEs were calculated for all 6 walls for all 100 different voxel percentages using the corresponding predictions and gold standard.

The mean BWPEs for all voxel percentage configurations are presented in Fig. 7.3. Er increased with p_v and the best predictions were obtained with the smallest voxel percentage ($p_v = 1$), i.e., the smallest number of leaf nodes. This is the expected behavior as the training error is minimized by the RRF. Thus the smallest p_v should produce the smallest BWPE. In addition to that, the localization prediction of bone structures (left femur head, right femur head, left pelvis, right pelvis, and L5 vertebra) appears to be better than the prediction of soft tissue organs (left kidney, right kidney, liver, and spleen).

The remaining 45 testing images (Ω_P) were used to generate the result set R_P^1 using I^1. Resulting BWPEs (Er) of R_P^1 are presented in Fig. 7.4. For 7 organs, Er obtained for $p_v = 1$ in R_P^1 was not the lowest among the observed Er for the different voxel percentages. But $p_v = 1$ resulted in the best prediction for the spleen and right femur head. The worst prediction for the spleen can be observed in the range $35 < p_v < 50$. The localization prediction for the right pelvis appears to be stable across all p_v. For the other seven anatomical structures, Er appears to increase continuously for $p_v > 10$. As there is no single constant value for p_v that results in the best localization prediction, using an arbitrary value of p_v may not be optimal. Hence we can state that there does not exist an optimal number of leaf nodes that produce the best localization results.

Among the 100 values that were used, $p_v = 75$ used in Criminisi et al. [2013] did not produce the best predictions for any anatomical structure. Similarly to R_T^1, the localization prediction in R_P^1 appears to be better for

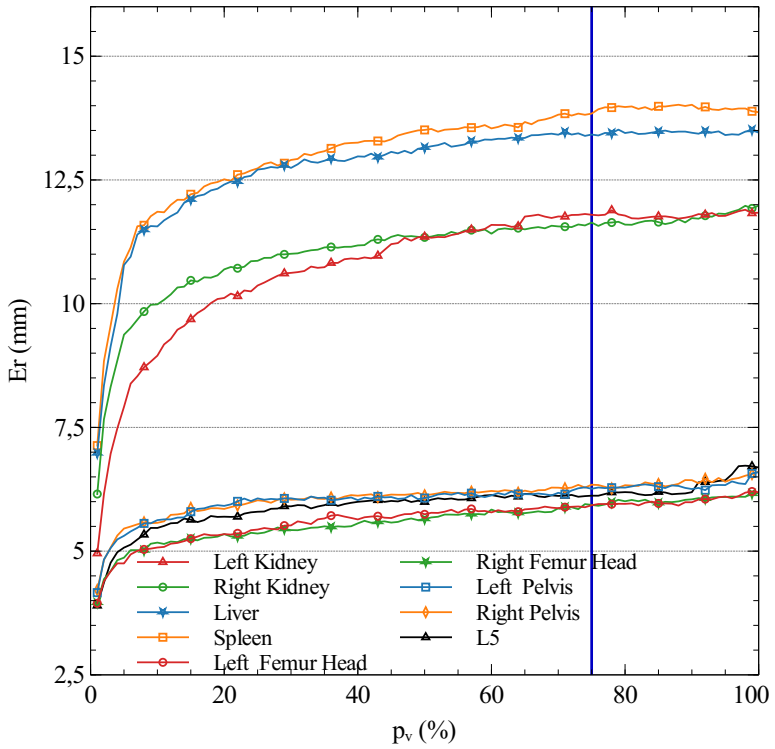


Figure 7.3: Mean BWPEs (Er) for I^1 using Ω_T (the image set used for training) (R_T^1). The blue vertical line represents $p_v = 75$ as used in [Criminisi et al., 2013]. All 9 organs produce the smallest BWPE for $p_v = 1$. And this is the expected behaviour as the images used for training were used for generating the localization predictions.

bone structures than for soft tissue organs.

Er obtained for Ω_P were greater than Er obtained for Ω_T but the ranges of Er obtained for Ω_P were less than those obtained for Ω_T for all 9 organs.

7.3.2 Use of The New Voxel Percentage Selection Criterion

In I^2, p_v was chosen as the percentage of voxels that was classified as the foreground by the Otsu threshold operation (see Sect. 7.2.2). Hence, this value was unique for each image used at the prediction phase. These values ranged from 29.8% to 72.3% for the 45 images of Ω_P with a mean and standard deviation of 51.5% and 9.0% respectively. These values were very different from p_v found in the literature (1%, 3%, 75%).

Two result sets (R_T^2 and R_P^2) were generated using I^2 and the above

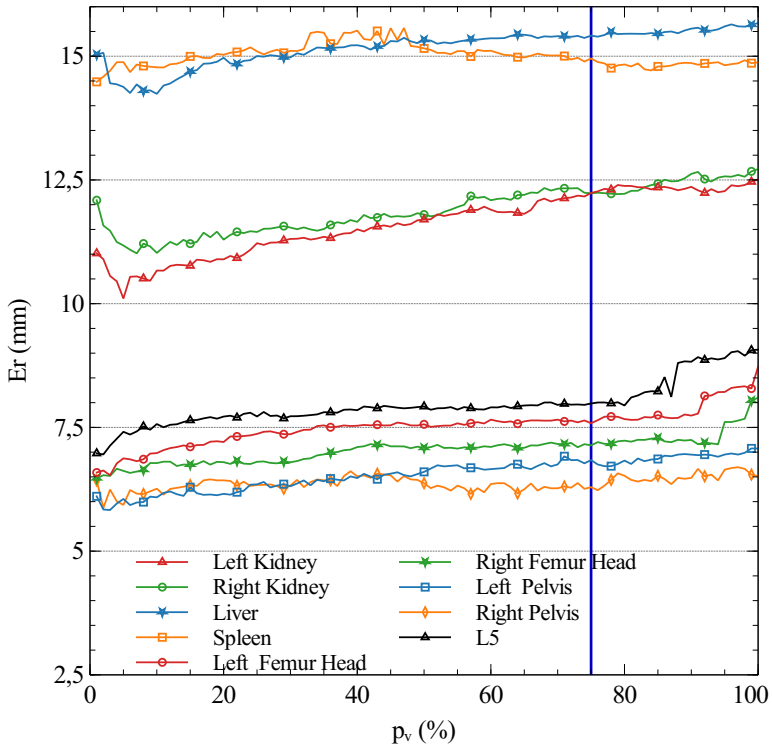


Figure 7.4: Mean BWPEs (Er) for I^1 using Ω_P as the prediction image set (R_P^1). The blue vertical line represents $p_v = 75$ as used in [Criminisi et al., 2013]. No single p_v can be picked for all 9 organs that produces the smallest mean BWPE.

found voxel percentages. The predictions were made following the procedure described in Sect. 3.4.7 and Er were calculated in the same manner as described above.

Table 7.2 summarizes the mean, standard deviation, median, and max BWPEs obtained for each organ for the images of Ω_T. R_T^2 appears to behave in the same manner as R_T^1 where Er for bone structures are less than those for soft tissue organs. Another interesting observation is that the median BWPE values for all anatomical structures are less than Er. Also, the difference between Er and median BWPE is always greater than the difference between Er and standard deviation values for each anatomical structure.

Similarly to the Table 7.2, the Table 7.3 summarizes the mean, standard deviation, median, and max BWPEs obtained for each organ for the images

	Mean	Std. Dev	Median	Max
Left Kidney	12.2	10.6	9.6	60.0
Right Kidney	12.0	10.2	9.2	49.0
Liver	13.5	12.2	10.3	97.0
Spleen	13.9	12.5	10.6	72.0
Left Femur Head	6.9	7.4	4.9	58.0
Right Femur Head	6.8	6.7	5.3	54.7
Left Pelvis	7.2	7.2	5.4	57.3
Right Pelvis	7.4	6.5	6.0	38.0

Table 7.2: Mean BWPEs (R_T^2) generated using I^2 and Ω_T. All measurements are in mm.

of Ω_P. Er for soft tissue organs are greater than those for bone structures similar to the previous result sets. Same behavioral comparisons as in R_T^2 can be observed regarding Er, median BWPE and standard deviation values. Localization prediction of R_P^2 resulted in greater errors than R_T^2.

	Mean	Std. Dev	Median	Max
Left Kidney	12.8	11.9	9.9	89.0
Right Kidney	13.3	10.9	11.5	75.0
Liver	15.1	13.7	10.6	75.0
Spleen	14.5	13.5	11.0	85.0
Left Femur Head	8.9	7.7	7.0	41.0
Right Femur Head	8.7	7.6	5.0	36.0
Left Pelvis	8.4	6.1	6.1	23.0
Right Pelvis	9.4	7.6	7.9	37.0

Table 7.3: Mean BWPEs (R_P^2) generated using I^2 and Ω_P. All measurements are in mm.

As mentioned above, I^2 uses a unique voxel percentage for each image used for prediction that results in a unique prediction. Consequently, a direct comparison between R_P^1 and R_P^2 is not possible. Hence, similarly to Criminisi et al. [2013], the localization predictions obtained by R_P^1 for $p_v = 75$ ($R_{P,75}^1$) were compared to R_P^2. The resulting Inter Quartile Range (IQR) box plots are given in Fig. 7.5.

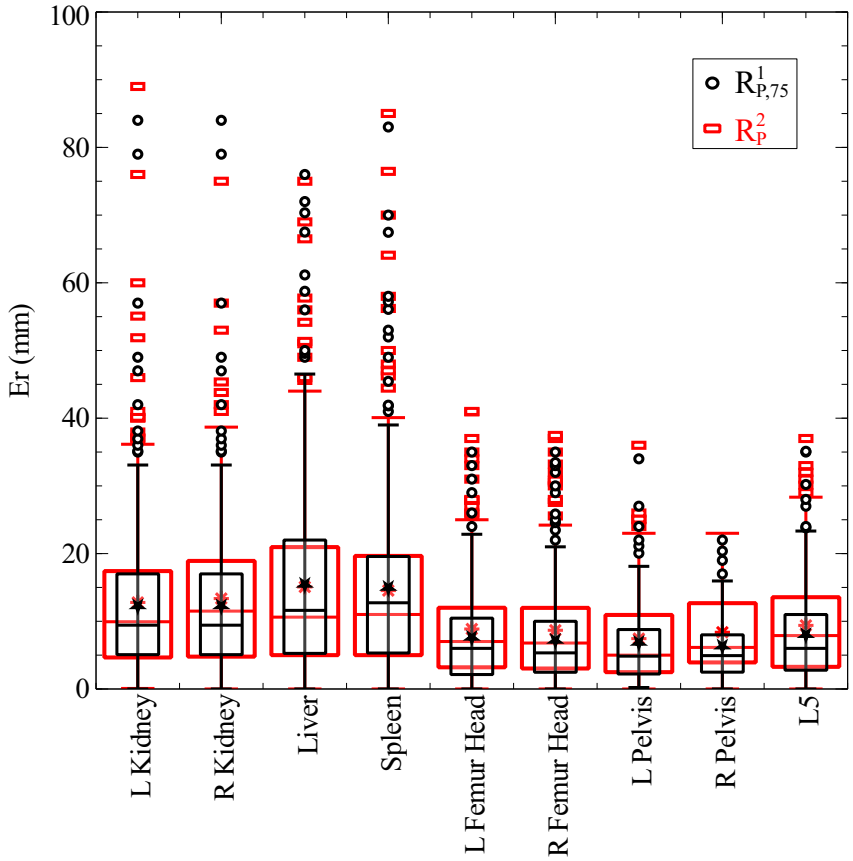


Figure 7.5: 1.5 Inter Quartile Range (IQR) box plots of mean BWPEs for each prediction image of $R^1_{P,75}$ and R^2_P.

Er for the liver and spleen of R^2_P were slightly less than that of $R^1_{P,75}$. For the other 7 organs, Er of R^2_P were marginally greater than those of $R^1_{P,75}$. The Inter Quartile Ranges (IQRs) of Er follow the same behavior. For all organs Er were greater than median BWPE for both $R^1_{P,75}$ and R^2_P. The number of outliers and their range appear to be quite similar for both $R^1_{P,75}$ and R^2_P. Finally and more importantly regarding our hypothesis, no statistical significance was observed between $R^1_{P,75}$ and R^2_P for any organ with p–value $\in [0.03 - 0.86]$.

A comparison of maximum BWPE ($max\ Er$) between I^1 and I^2 provides a good indication about the suitability of the methods for localization prediction. Fig. 7.6 represents $max\ Er$ of R^1_T.

For all the organs apart from the spleen, $max\ Er$ increased with p_v and the smallest $max\ Er$ were obtained with $p_v = 1$, i.e., the smallest n_p. In

addition to that, the greatest ranges for $max\ Er$ can be observed for the liver, left pelvis, and L5 vertebra. Furthermore, there is no clear distinction between the bone structures and soft tissue organs as previously observed for Er values.

In the same figure, $max\ Er$ generated by R_T^2 are indicated by bigger symbols. The maximum BWPE generated by R_T^2 for the left kidney, left femur head, right femur head, and right pelvis were always greater than $max\ Er$ values generated by R_T^1 for all values of p_v. Interestingly, $max\ Er$ for the right femur head, left pelvis, and right pelvis were obtained for the same image of Ω_T.

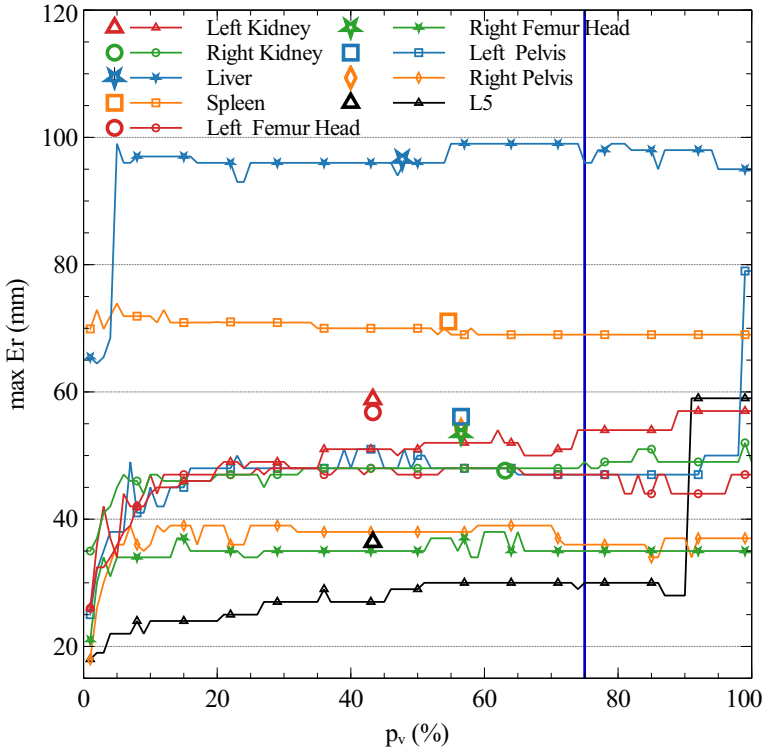


Figure 7.6: $max\ Er$ along any direction for I^1 using Ω_T as the prediction image set. The values indicated by bigger symbols correspond to the values obtained using I^2 and Ω_T. The blue vertical line represents $p_v = 75$.

Similarly, Fig. 7.7 represents $max\ Er$ of R_P^1. Similar to the observations made for Fig. 7.4, for 8 organs, $max\ Er$ obtained for $p_v = 1$ was not the smallest among all the different voxel percentages. But $p_v = 1$ resulted in the smallest $max\ Er$ only for the left femur head. Very large range of

max Er values can be observed for the left femur head, right femur head, and L5 vertebra. Interestingly, *max Er* seems to decrease with p_v for the liver.

As in the previous figure, *max Er*s generated by R_P^2 are indicated by bigger symbols in the same figure. Unlike in R_T^2, there is no *max Er* value generated by R_P^2 that is greater than *max Er* values generated by R_P^1 for all p_v for any organ. Contrary to R_T^2, the distinction between the bone structures and soft tissue organs can clearly be seen in R_P^2. The maximum BWPE for the right kidney, and liver were obtained for the same image of Ω_P.

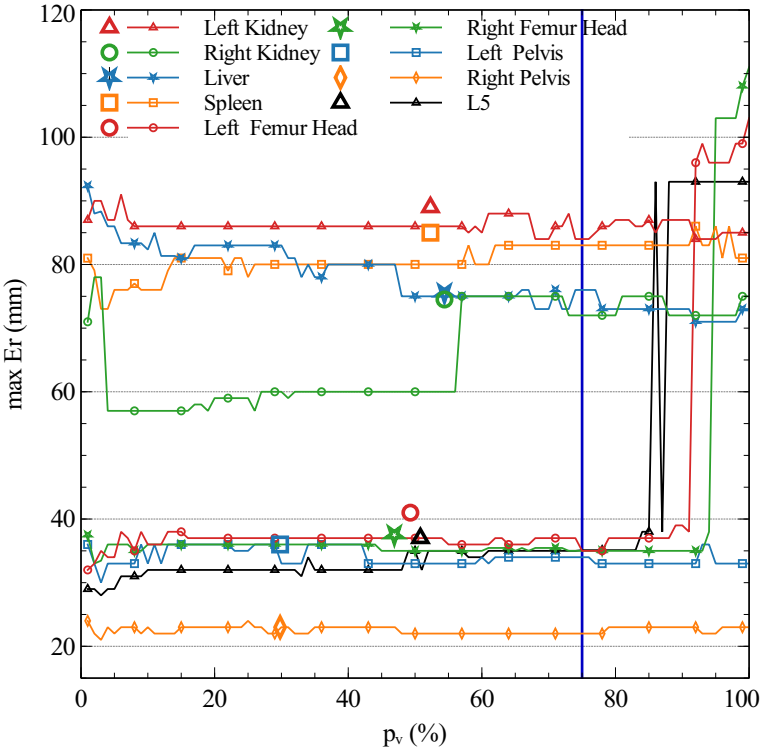


Figure 7.7: *max Er* along any direction for I^1 using Ω_P as the prediction image set. The values indicated by bigger symbols correspond to the values obtained using I^2 and Ω_P. The blue vertical line represents $p_v = 75$.

Although, *Er* obtained for R_P^1 and R_P^2 were greater than *Er* obtained for R_T^1 and R_T^2 respectively, the same does not hold true with respect to *max Er*.

In all R_T^1, R_P^1, R_T^2, and R_P^2, *Er* were representative of all the images.

Fig. 7.8 was generated in order to look at the variation of Er of a single image with different p_v values. It illustrates Er of the left kidney for 5 testing images for 100 different p_v values (from 1 to 100). The minimum Er of image 1 to image 5 are observed at $p_v = 100, 95, 4, 1$, and 11 respectively. On the other hand, the maximum Er of image 1 to image 5 are observed at $p_v = 1, 2, 96, 100$, and 100 respectively. This clearly illustrates that no single arbitrary value is capable of producing better results for all images.

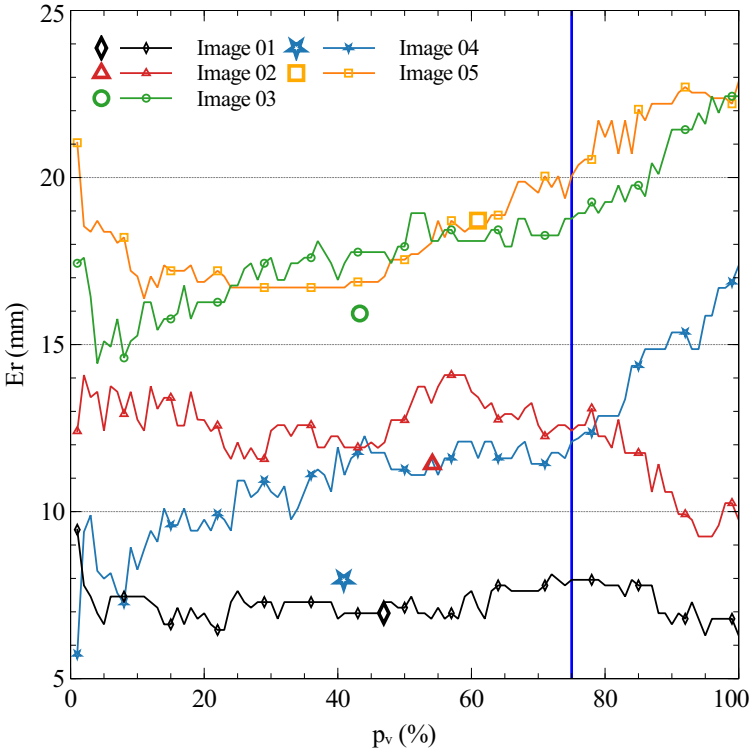


Figure 7.8: Mean BWPEs (Er) of the left kidney for 5 prediction images using I^1. Bigger symbols illustrate Er obtained by I^2. The blue vertical line represents $p_v = 75$.

7.4 Discussion

Using voxels inside a regular grid within ±10 cm of the center of each axial slice of the image volume as in I^1 may lead to scenarios (see Fig. 7.1b) where much of the patient body is not considered as possible landmarks for training of the RRF. This may lead to sub-optimal RRFs.

A regular CT image generally consists of patient's body, bed of the CT scanner, and background. The background mostly consists of air. But air is also present in patient's lungs and intestines. As CT images are represented using HU, the gray levels of the CT image can be directly mapped to the imaged physical matter in medical grade CT scans as well as cone beam computed tomography [Mah et al., 2010].

We proposed the image preprocessing steps performed in I^2 (see Sect. 7.2.1) in order to distinguish between air–like regions and the rest. After the Otsu threshold operation, the air–like regions are automatically clustered into one class and the rest into the other. Setting the graylevel of voxels belonging to air–like regions to -1000 signifies assigning them with the HU of air. Since RRF training process is not expected to use the subtle differences of the air–like regions which mostly consist of noise but rather discriminate the body regions, we suggest that unifying the air–like regions is advisable. In addition to that, this method of reducing the influence of the background noise can be directly applied to other image modalities such as MRI where only the representative zero value needs to be adjusted depending on the modality.

In I^2, as all the voxels in an image were used for training, the air-like regions inside the body were also considered. Since the size of an image slice in the dataset ranged from 26.3×26.3 cm^2 to 46.6×46.6 cm^2 and since I^1 only used ± 10 cm grid area from the center of each axial slice, it may have missed a number of landmarks that lie outside that range. If the voxels belonging to the non–air–like regions were solely used for training in I^2, the landmarks that may exist near the skin may not have been captured by the training process in addition to loosing the probable landmark regions inside or near the lungs and intestines. These possible land marks were likely already lost in I^1 as it did not consider any voxel out of the ± 10 cm grid from the center of each axial slice.

R_T^1 provides compelling evidence for the correctness of RRF. When predicting the localization of a single organ, the leaf nodes having the highest confidence in prediction are found at the head of the selection queue due to the sorting step carried out (see Sect. 3.4.7.1). Thus, choosing the smallest percentage of voxels for prediction, results in selecting these highest confident leaf nodes. For all images in Ω_T, the best predictions are obviously expected with the smallest p_v (*i.e.*, using the smallest n_p). In addition to

that, the accuracy of prediction for all images in Ω_T is expected to continuously decrease when the used percentage of voxels increases. It is the expected behavior as the predictions were made using the same images which were used for training (Ω_T) the RRF. This was exactly what was observed in R_T^1 (see Fig. 7.3). Obtaining the smallest $max\ Er$ of R_T^1 for $p_v = 1$ for all 9 organs provides further evidence to this claim.

Er obtained for $p_v = 1$ in R_P^1 did not provide the smallest Er for all organs except for the spleen and right femur head. It may be due to the fact that the best leaf nodes chosen from the training phase (the ones with the least variation of the covariance matrix of the stored offset vectors) may have collected a small number of voxels or no voxels at all during the prediction phase. It may also hint that I^1 is more specific to the provided training images since the voxels are extracted from a very restrictive ROI.

7.5 Conclusion

Even after analyzing the localization predictions generated using the same images used for training, a single p_v (hence, n_p) value that gives better results with previously unseen images can not be found. Instead, providing RRF method with the highest amount of relevant information both in training and prediction phases while reducing the influence of the background appears to produce comparable results as the state of the art results.

In addition to that, using the proposed fully automatic consistent method for picking the forest leaves (n_p) for organ localization prediction may boost the usability of RRF not only in medical image processing but also in other fields as well.

In extending the present work, it would be interesting to see whether further automatic parametrization is possible. Comparing the impact of randomly selecting voxels of an image volume for training and prediction against randomly selecting voxels from ROI where useful information is believed to exist may lead to interesting results as well. Finally, devising a mechanism to assess the quality and confidence of the localization prediction without having to use manual localization results will be essential to the wide usage and adaptability of RRF method.

In this chapter, we looked at three important steps of RRF induction process. Namely, 1) how voxels are chosen for training, 2) how voxels are chosen to be pushed through a trained forest in the prediction phase, and 3) how many leaf nodes are picked for the final localization prediction.

We observed in the literature that a variety of voxel selection choices are available for step 1 and 2, ranging from using a random sub sample of voxels or selecting the voxels belonging to a region of interest. The number of leaf nodes that are picked for the final prediction was seen to be an arbitrary fixed amount depending on the study, suggesting a trial and error approach.

We hypothesized that the contribution of background voxels to the organ localization is minimal. Then, we observed that the noise present in the background voxels is not desirable to the localization procedure. Consequently, we proposed to make the noise present in the background voxels uniform by applying an Ostu threshold to the CT images and setting all background voxels to the same HU value of air (-1000 HU) which nullified the effect of the background noise.

With the set of new images whose background noise is nullified we used all voxels belonging to an image to train the RRF. Similarly, all voxels of a prediction image whose background noise was nullified were pushed through the forest at the prediction phase. Finally, the number of leaf nodes selected for final prediction was calculated as the percentage of voxels that belong to the foreground.

The empirical evidence gathered during the study provided comparable results to the state of the art results while the elimination of arbitrary values enhanced the general usability and adaptability of the RRF method.

Next, we are going to conclude the dissertation by providing a general discussion, a few concluding remarks, and finally a few perspectives.

Reasoning draws a conclusion, but does not make the conclusion certain, unless the mind discovers it by the path of experience.

- Roger Bacon

8

Discussion and Conclusion

Contents

The final chapter of this dissertation first presents an extensive discussion on the organ detection capabilities of the Random Regression Forest (RRF) method in Sect. 8.1. Then, the general conclusions of

the whole study are presented in Sect. 8.2 before presenting the future perspectives that stemmed from it in Sect. 8.3.

8.1 A Discussion on Organ Detection

This section discusses the organ detection capabilities of the Random Regression Forest (RRF) method. Up to now we have used RRFs for multi–organ localization making the simple assumption that the organs we were looking for were present in the images. An interesting question is what happens when an organ we are trying to localize is absent? Certain pathologies may require the ablation of organs that may lead to such scenarios. Organ detection is the ability of RRFs to detect the absence of an organ in such a situation.

First, we present what is mentioned in the literature on organ detection using RRFs in Sect. 8.1.1. Then, with the use of a simple toy example, in Sect. 8.1.2, we demonstrate the weaknesses of the existing detection criteria and call for the improvement of the RRF methodology in order to be more effective at organ detection.

8.1.1 Literature on Organ Detection Using RRFs

To the best of our knowledge, among the several studies that focus on multi–organ localization using RRFs, only two studies have presented specific details related to organ detection [Criminisi et al., 2010, 2013]. The ideas presented in the two studies are described below.

At the time of the localization prediction, we know that the posterior distribution of the bounding box vector $p(\mathbf{b}_c)$ for the organ (c) is found in the following manner:

$$p(\mathbf{b}_c) = \sum_{t=1}^{T} \sum_{l \in \mathcal{L}} p(\mathbf{b}_c \mid l) p(l) \ , \tag{8.1}$$

where T is the number of trees in the forest, l is a leaf node, $\mathcal{L}$ is the set of leaf nodes selected for prediction of the localization of the organ c, $p(\mathbf{b}_c \mid l)$ is the conditional distribution of bounding box vector, and $p(l)$ is the probability distribution prior (see Sect. 3.4.7.2). Then the absolute position $\left(\hat{\mathbf{b}}_c\right)$ of the bounding box of the organ c is determined by finding the arg max of the

above distribution:

$$\hat{\mathbf{b}}_c = \arg\max_{\mathbf{b}_c} \; p(\mathbf{b}_c) \;\; , \tag{8.2}$$

or as we demonstrated in Chap. 6 by the weighted summation of the mean bounding box vectors ($\bar{\mathbf{b}}_{c,l}$):

$$\hat{\mathbf{b}}_c = \sum_{t=1}^{T} \sum_{l \in \mathcal{L}_t} \bar{\mathbf{b}}_{c,l} \, p(l) \;\; , \tag{8.3}$$

where $\bar{\mathbf{b}}_{c,l} = \hat{\bar{\mathbf{v}}} - \bar{\mathbf{d}}(\mathbf{v}; c)$, $\bar{\mathbf{d}}(\mathbf{v}; c)$ is the mean offset vectors saved from the training phase, and $\hat{\bar{\mathbf{v}}} = (\bar{\mathbf{v}}_x, \bar{\mathbf{v}}_x, \bar{\mathbf{v}}_y, \bar{\mathbf{v}}_y, \bar{\mathbf{v}}_z, \bar{\mathbf{v}}_z)$ made from the mean voxel position $(\bar{\mathbf{v}}_x, \bar{\mathbf{v}}_y, \bar{\mathbf{v}}_z)$ of the voxels accumulated at the leaf node l during the prediction phase.

Criminisi et al. [2010] stated that an organ is declared present if the probability of the absolute position $\hat{\mathbf{b}}_c$ is greater than a threshold β in the following manner:

$$p\Big(\mathbf{b}_c = \hat{\mathbf{b}}_c\Big) > \beta \;\; . \tag{8.4}$$

Although they stated that $\beta = 0.5$ was selected for their study, there was no mention on how to select β, the authors probably meant application specific heuristic tuning.

In Criminisi et al. [2013], the authors proposed a different method for the organ detection that was based on the prediction confidence. They proposed to measure the confidence by fitting a 6D Gaussian with diagonal covariance matrix $\hat{\Lambda}$ using the 1D histograms in the vicinity of $\hat{\mathbf{b}}_c$. Then, they used $|\hat{\Lambda}|^{-1/2}$ as the measure of confidence of the final prediction. They suggested to use a threshold β depending on this confidence value and this parameter was proposed to be tuned depending on the application.

We believe that the latter proposal may be more appropriate than the former as a measure of detection. If the localization prediction points to a region in a medical image where an organ is not present, then it is only natural to expect the confidence of the prediction to be low. But is it really what happens in the real world?

We agree that the RRF method automatically selects the latent landmarks to localize a given organ [Criminisi et al., 2010, 2013]. What happens when all *supposed* landmarks are in place but not the organ? Then, is the confidence of the localization prediction low? Since the spatial context of

the vicinity of the organ is not affected by the presence of the organ itself, we believed that further investigation on the matter may provide interesting results.

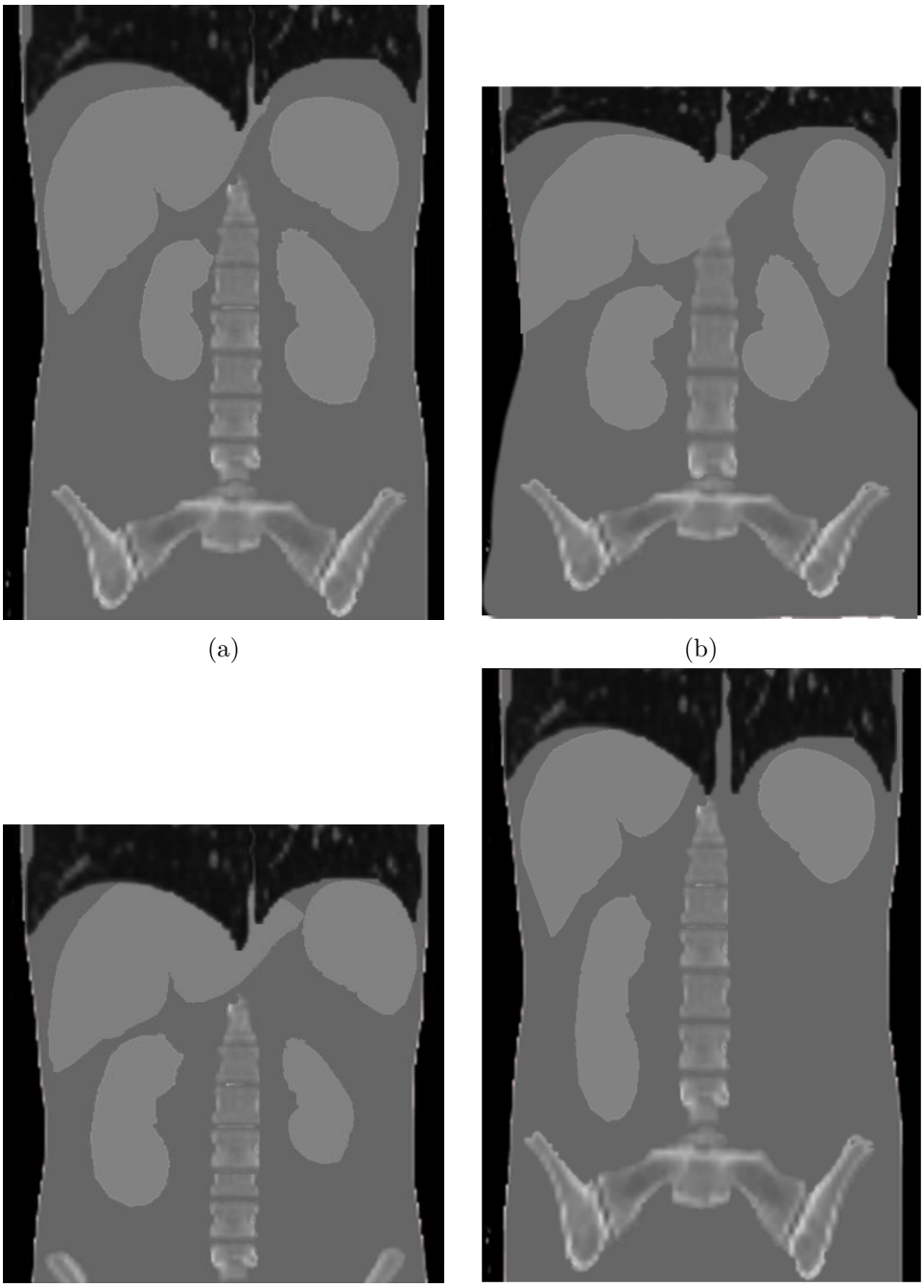
8.1.2 A Toy Example of Organ Detection

Since we did not possess any Computed Tomography (CT) images where the organs were totally absent, we decided to create a synthetic set of 2D images. The images were used in a toy example that further investigated organ detection using RRFs. The images were made to mimic a coronal slice of a CT image that comprised cross sections of the liver, spleen, left kidney, right kidney, spine, and upper part of the hip bone. We used 4 coronal slices of real CT images belonging to 4 different patients to extract the shapes of the liver, spleen, left kidney, right kidney, and the body. These extracted organs were given the same uniform gray level value. The extracted body shapes were given a darker gray level value. The visible part of the lungs, spine, and upper hip bone were cropped from those 4 images without any gray level modification operation. The background was set to uniform 0 Hounsfield Unit (HU). Finally, the shapes were subjected to different deformations to form the synthetic dataset.

Our synthetic 2D images consisted of 12 training images and 4 testing images. Two training images and two testing images are presented in Fig. 8.1. The first testing image (I_1) has its lower region cropped (see Fig. 8.1c) whereas the second testing image (I_2) was generated without the left kidney (see Fig. 8.1d).

A RRF comprising 4 Random Regression Trees (RRTs) each having 7 maximum decision levels was trained to predict the localization of the liver, spleen, left kidney, and right kidney according to the normal training procedure described in Sect. 3.4.6. Mean intensity of displaced rectangles (since images are 2D) were used as the set of features of the forest.

The trained RRF was used to localize the 4 organs in each testing image in the prediction phase using the general procedure presented in Sect. 3.4.7. The resulting predictions for the left kidney of I_1 and I_2 are presented in Fig. 8.2. We observe that the left kidney localization prediction of I_1 is acceptable although not very accurate (see Fig. 8.2a). On the other hand, a non-existent left kidney is localized in I_2 (see Fig. 8.2b). Yet, as the landmarks that surround the left kidney are all in their *correct* places in

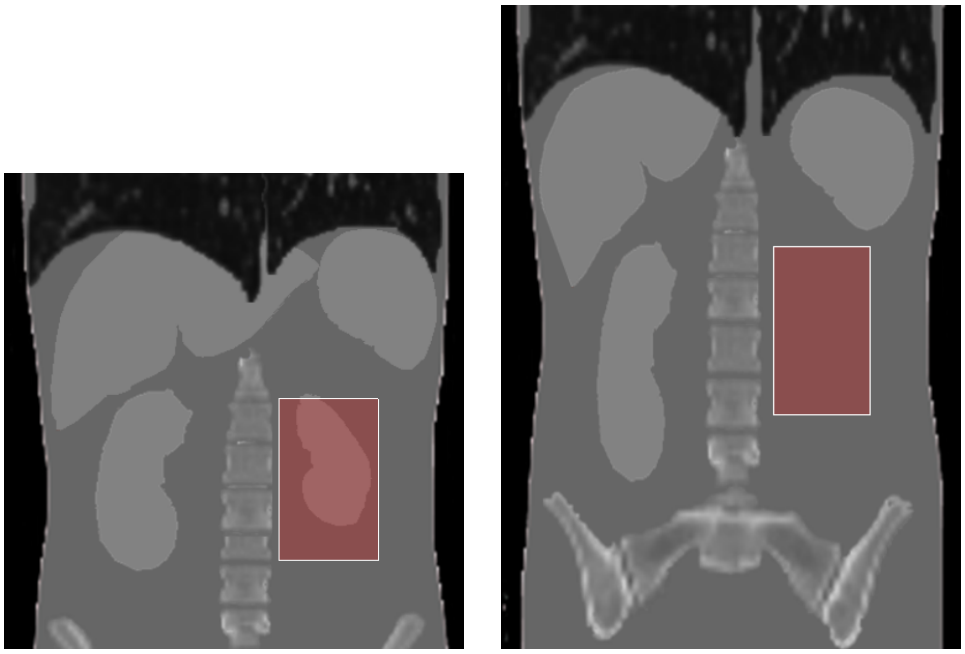


(a) (b)

(c) (d)

Figure 8.1: A few sample images of the synthetic database. (a,b) Two training images. (c) A testing image (I_1) which has its lower part cropped. (d) Another testing image (I_2) where the left kidney is missing.

I_2, this prediction is in total agreement with the concept of multi–organ localization using RRFs.



(a) (b)

Figure 8.2: Sample predictions of the left kidney. (a) The localization prediction is roughly correct although not very accurate in I_1. (b) Although the organ is not present in I_2, the RRF algorithm localizes the region where the organ would have been present. Since the landmarks of the left kidney are *correctly* positioned, this prediction is coherent. The predicted bounding box of I_2 appears to be elongated than the bounding box of I_1 following the general morphology of the two images.

The above presented scenario in Fig. 8.2 is an ideal opportunity to measure the strength of the organ detection proposals that were discussed in Sect. 8.1.1. The distributions of $p(\mathbf{b}_c)$ along the right and left directions for I_1 are presented as a red curve in Fig. 8.3. The corresponding distributions for I_2 are presented in the same figure using the blue dashed curves. Both curves are very similar in shape and in placement for both directions. The probability of the final predicted localization $p\Big(\mathbf{b}_c = \hat{\mathbf{b}}_c\Big)$ lie very close to each other. As their shapes are similar, the variance of the curves are also similar. Hence, the organ detection using either of the criteria presented in Sect. 8.1.1 will fail to distinguish I_2 from I_1.

The distributions of $p(\mathbf{b}_c)$ along the inferior and superior directions for

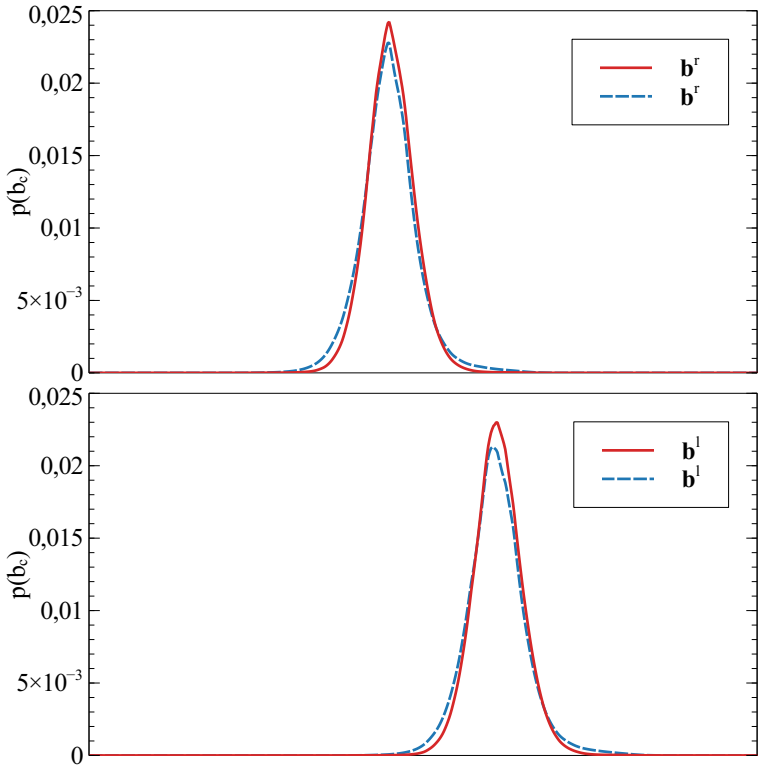


Figure 8.3: $p(\mathbf{b}_c)$ along the right (top figure) and left (bottom figure) directions. Both $p\left(\mathbf{b}_c = \hat{\mathbf{b}}_c\right)$ and variance of each distribution of I_1 (in red) appears to be very similar to the corresponding distribution of I_2 (in blue). Consequently, organ non-existance can not be determined using either of the criteria described in Criminisi et al. [2010, 2013] in this scenario.

I_1 are presented as a red curve in Fig. 8.4. The same distributions of I_2 are presented in the same figure using the blue dashed curves. Unlike the previous case, the shape of the curves are very different from one another. The distributions of I_1 produce two local maximas whereas the distributions of I_2 produce a single maxima. If we consider the variance of the two set of curves, the variance of the curves corresponding to I_1 are much larger than those corresponding to I_2. Additionally, if we consider $p\left(\mathbf{b}_c = \hat{\mathbf{b}}_c\right)$, the value for the curves corresponding to I_1 are smaller than those corresponding to I_2. Interestingly, according to the two criteria mentioned in Sect. 8.1.1, I_1 will be selected as the image where the organ is non–existent whereas in reality, the left kidney is not present in the other image, I_2.

This simple example illustrates the tricky nature of organ detection using

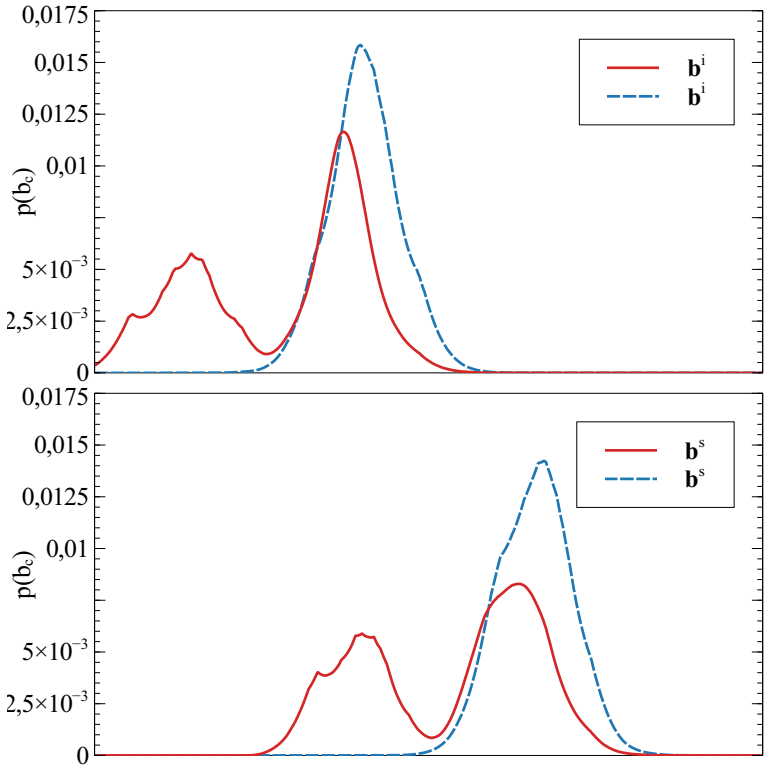


Figure 8.4: $p(\mathbf{b}_c)$ along the inferior (top figure) and superior directions (bottom figure). The variance of both curves of I_1 is much larger than corresponding values for I_2. Furthermore, $p\left(\mathbf{b}_c = \hat{\mathbf{b}}_c\right)$ of I_1 is smaller than that of I_2. Hence, according to the detection criteria mentioned in Sect. 8.1.1, I_1 might be selected as the organ non–exitent image. But the left kidney is non–exitent in I_2.

RRFs. It also suggests that much work needs to be carried out in order for RRFs to have strong organ detection capabilities.

It is our belief that organ detection should not only consider the final prediction distribution, but consider other possibilities too. This is based on the fact that the surroundings alone can not determine whether an organ is present or not in a given medical image. Incorporating classical intensity based segmentation methods or augmenting the current RRF method with organ specific information may be interesting ideas to pursue in making the current RRF method more powerful at organ detection.

The general conclusions of the dissertation are provided in the next section.

8.2 General Conclusion

The main aim of the dissertation was to carry out a detailed study of the use of the RRF method for multi–organ localization in CT images.

One of the most important outcomes of the study, on a theoretical level, was the advancement of the knowledge of using RRFs for multi–organ localization in CT images through a thorough understanding of the inner workings of the methodology. To this extent, we analyzed the basic building block of the RRF methodology, the decision tree, in Chap. 2, and the RRF methodology itself in Chap. 3 along with a thorough bibliographical review. The main steps of the decision tree induction and the best split selection process were analyzed in Chap. 2. The analysis of the RRF methodology was carried out with respect to the multi–organ localization problem in Chap. 3. This chapter presented and analyzed how multi–organ localization problem was reformulated as a relative displacement regression problem along with an analysis of the main phases of the RRF method.

Chapter 4 presented the scientific approach followed in conducting the subsequent studies. This unified approach laid the foundations for the achievement of the more practically oriented expectations of the dissertation. The first study presented in Chap. 5 augmented the classic RRFs with spatial consistent information contrary to the classic RRFs which rely on direction independent hypothesis. The empirical evidence gathered during the study lead to the conclusion that the augmented RRF method was not significantly different from the classic RRF method. In Chap. 6, we proposed Light Random Regression Forests (LRRFs) that were more memory efficient and faster than classic RRFs while maintaining the same localization capabilities as the classic RRFs. Our proposal for selecting the number of forest leaf nodes for the final prediction instead of using arbitrary values as done in the literature presented in Chap. 7 was a first step towards a more generalized RRF framework. While our proposal was more systematic and consistent, eliminating the need to tune more parameters, it did not reduce the localization capabilities.

The proposals put forward in Chap. 5, Chap. 6, and Chap. 7 are not restricted to random regression forest by any means. They are most certainly applicable in the context of random forests and can be made to suit in the more general context of machine learning. This satisfies the final expectation

laid down for the dissertation that our proposals may be applied in other forms of random forests.

Although our proposals stemmed from either a theoretical hypothesis or an observation, the fact that no statistical significant results were obtained between the classic RRF method and any of the proposals may be potentially attributed to two factors. First, it may suggest that the classic RRF method is highly robust and that the proposed extensions did not modify the core statistical process of the classic RRF method. Secondly, it may be due to the fact that the wide variation within our image database is canceling out the eventual differences in the localization performances of the different proposals. As our aim was to provide a tool for the clinicians for the eventual practical use, having such a diverse database was not only unavoidable but mandatory as well.

Finally, we have taken the first steps of making the Light Puncture Robot (LPR) autonomous by enabling fully automatic multi–organ localization using RRFs.

Next, we present the general perspectives that stemmed from this dissertation.

8.3 General Perspectives

The studies conducted during the course of this dissertation unveiled numerous theoretical and practical perspectives. The short term perspectives and long term perspectives are addressed respectively in Sect. 8.3.1 and Sect. 8.3.2.

8.3.1 Short Term Perspectives

An immediately feasible task would be to study the influence of various other parameters of the RRF algorithm on the multi–organ localization that were not studied during this dissertation. It will also be interesting to combine the LRRF proposed in Chap. 6 with the automatic parametrization proposed in Chap. 7.

Now that a robust multi–organ localization method is at our disposal, one of the next steps would be to link the localization to classical segmentation methods in order to segment organs fully. As obtaining multi–organ

segmentation may appear too ambitious, as an initial step, segmenting one organ at a time may be the way forward.

Throughout the studies carried out during the dissertation, we observed that the localization of bone structures (the left femur head, right femur head, left pelvis, right pelvis, and L5 vertebra) using RRFs produced relatively smaller errors than the localization of soft tissue organs (the liver, spleen, left kidney, and right kidney). This may be due to the fact that the bone structures are fairly similar across different patients whereas the soft tissue organs may be subjected to large variations in size, shape and spatial localization. In addition to that, bone structures have very high HU values compared to soft tissue organs which may make them relatively easily locatable too. In contrast, all the soft tissue organs used for the dissertation possess very similar HU values making them harder to distinguish. It may be interesting to carry out separate localization of soft tissue organs and bone structures. Analyzing the outcomes of such an experiment may provide insight to using different intrinsic parameters of RRF for the two scenarios.

In order to share the strength and heterogeneity of our dataset, proposing a fully automatic multi–organ localization challenge as a satellite event in a relevant conference such as the Medical Image Computing & Computer Assisted Intervention (MICCAI) conference might be very interesting. This will provide an opportunity for the numerous algorithms specialized in localization to compete and measure their strengths and weaknesses against one another ultimately contributing to the betterment of the field.

8.3.2 Long Term Perspectives

It is our belief that much work needs to be done in order to make the RRF method a powerful organ detection tool (see Sect. 8.1). If organ specific information incorporation is selected as the way forward, then, how organ specific information is going to be inserted into the current RRF setting without many changes, and how to couple this information with the prediction distribution confidence in order to decide on organ detection will be very interesting questions to answer.

Porting the RRF algorithm for multi–organ localization in other image modalities other than CT images would be another path to explore. The first step taken in that direction would be to devise image modality specific

features that capture the naturally present contextual information of medical images. Once the suitable features are discovered, the application of the RRF method would follow the same principles.

Supervised machine learning techniques specialized in organ segmentation require segmented organs as their training set. Segmenting an organ manually or semi-automatically takes a long time. Hence finding experts willing to segment organs is not a trivial task. In this setting, our long term dream goal is to perform multi–organ segmentation using random regression forests as the base tool without providing a segmented training set but only a localized training set. The tackling of this conundrum would present ample theoretical and practical opportunities for research. Our team wishes to explore these paths in the quest to make the Light Puncture Robot an autonomous robot.

Finally, we hope that the generalization advancements of the RRF method made during this dissertation will follow more research work that pursue the same generalization notion in order to make the wonderful tool of RRFs be used across different medical applications in years to come.

We would like to conclude the dissertation with a quotation from William Belson, the founding father of decision trees: "*The method as I have described it is, it is true, a movement towards a more empirical way of doing things; but it is just as much a movement away from a sophistication which is too often either baffling or misleading*" [Belson, 1959].

References

N. A. Ahmed and D. Gokhale. Entropy expressions and their estimators for multivariate distributions. *IEEE Transactions on Information Theory*, 35 (3):688–692, 1989.

H. Altınçay. Decision trees using model ensemble-based nodes. *Pattern recognition*, 40(12):3540–3551, 2007.

Y. Amit and D. Geman. Randomized inquiries about shape: An application to handwritten digit recognition. Technical report, DTIC Document, 1994.

A. Arauzo-Azofra, J. L. Aznarte, and J. M. Benítez. Empirical study of feature selection methods based on individual feature evaluation for classification problems. *Expert Systems with Applications*, 38(7):8170–8177, 2011.

D. H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122, 1981.

M. M. Barbieri and J. O. Berger. Optimal predictive model selection. *Annals of Statistics*, pages 870–897, 2004.

F. Baumann, F. Li, A. Ehlers, and B. Rosenhahn. Thresholding a random forest classifier. In *Advances in Visual Computing*, pages 95–106. Springer, 2014.

W. A. Belson. Matching and prediction on the principle of biological classification. *Applied statistics*, pages 65–75, 1959.

M. Ben-Bassat. Use of distance measures, information measures and error bounds in feature evaluation. *Handbook of statistics*, 2:773–791, 1982.

S. Bernard, L. Heutte, and S. Adam. On the selection of decision trees in random forests. In *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, pages 302–307. IEEE, 2009.

H. Blockeel, L. De Raedt, and J. Ramon. Top-down induction of clustering trees. In *Proceedings of the 15th International Conference on Machine Learning*, pages 55–63. Morgan Kaufmann, 1998.

V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos. A review of feature selection methods on synthetic data. *Knowledge and information systems*, 34(3):483–519, 2013.

A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

L. Breiman. Pasting small votes for classification in large databases and on-line. *Machine Learning*, 36(1-2):85–103, 1999.

L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

L. Breiman and C. Stone. Parsimonious binary classification trees. *Technology Service Corporation, Santa Monica, Calif. Tech. Rep. TSCCSD-TN-004*, 1978.

L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, 1984.

I. Bricault, N. Zemiti, E. Jouniaux, C. Fouard, E. Taillant, F. Dorandeu, and P. Cinquin. Light puncture robot for CT and MRI interventions. *IEEE Engineering in Medicine and Biology Magazine*, 27(3):42–50, 2008.

P. Bromiley, J. Adams, and T. Cootes. Localisation of vertebrae on DXA images using constrained local models with random forest regression voting. In *Recent Advances in Computational Methods and Clinical Applications for Spine Imaging*, pages 159–171. Springer, 2015.

W. Buntine. Learning classification trees. *Statistics and computing*, 2(2): 63–73, 1992.

C. Chen and G. Zheng. Fully automatic segmentation of ap pelvis x-rays via random forest regression and hierarchical sparse shape composition. In *Computer Analysis of Images and Patterns*, pages 335–343. Springer, 2013.

C. Chen and G. Zheng. Fully automatic segmentation of ap pelvis x-rays via random forest regression with efficient feature selection and hierarchical sparse shape composition. *Computer Vision and Image Understanding*, 126:1–10, 2014.

I. Chikalov. *Average Time Complexity of Decision Trees (Intelligent Systems Reference Library) (Volume 21)*. Springer, 2011 edition, 2013.

C. Chu, C. Chen, and G. Zheng. Fully automatic ct segmentation for computer-assisted pre-operative planning of hip arthroscopy. In *Computer-Assisted and Robotic Endoscopy*, pages 55–63. Springer, 2014.

C. Chu, C. Chen, L. Liu, and G. Zheng. Facts: fully automatic ct segmentation of a hip joint. *Annals of biomedical engineering*, 43(5):1247–1259, 2015.

P. Cinquin, E. Bainville, C. Barbe, E. Bittar, V. Bouchard, L. Bricault, G. Champleboux, M. Chenin, L. Chevalier, Y. Delnondedieu, et al. Computer assisted medical interventions. *Engineering in Medicine and Biology Magazine, IEEE*, 14(3):254–263, 1995.

E. M. Clarke. Tree–based models. In *in Statistical Models*, pages 377–419, 1992.

A. Criminisi and J. Shotton, editors. *Decision Forests for Computer Vision and Medical Image Analysis (Advances in Computer Vision and Pattern Recognition)*. Springer, 2013 edition, 2013.

A. Criminisi, J. Shotton, and S. Bucciarelli. Decision forests with long-range spatial context for organ localization in CT volumes. In *MICCAI Workshop on Probabilistic Models for Medical Image Analysis*, 2009.

A. Criminisi, J. Shotton, D. Robertson, and E. Konukoglu. Regression forests for efficient anatomy detection and localization in CT studies. In *Medical Computer Vision. Recognition Techniques and Applications in Medical Imaging*, pages 106–117. Springer, 2010.

A. Criminisi, D. Robertson, E. Konukoglu, J. Shotton, S. Pathak, S. White, and K. Siddiqui. Regression forests for efficient anatomy detection and localization in computed tomography scans. *Medical image analysis*, 17 (8):1293–1303, 2013.

R. Cuingnet, R. Prevost, D. Lesage, L. D. Cohen, B. Mory, and R. Ardon. Automatic detection and segmentation of kidneys in 3D CT images using random forests. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012*, pages 66–74. Springer, 2012.

M. Dantone, J. Gall, C. Leistner, and L. V. Gool. Human pose estimation using body parts dependent joint regressors. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3041–3048, June 2013.

M. Dantone, J. Gall, C. Leistner, and L. V. Gool. Body parts dependent joint regressors for human pose estimation in still images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2131–2143, Nov 2014.

B. De Ville. Applying statistical knowledge to database analysis and knowledge base construction. In *Artificial Intelligence Applications, 1990., Sixth Conference on*, pages 30–36. IEEE, 1990.

L. R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

R. Donner, B. H. Menze, H. Bischof, and G. Langs. Global localization of 3d anatomical structures by pre-filtered hough forests and discrete optimization. *Medical image analysis*, 17(8):1304–1314, 2013.

R. O. Duda and P. E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.

T. Ebner, D. Stern, R. Donner, H. Bischof, and M. Urschler. Towards automatic bone age estimation from mri: Localization of 3d anatomical landmarks. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2014*, pages 421–428. Springer, 2014.

G. Fanelli, J. Gall, and L. Van Gool. Real time head pose estimation with random regression forests. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 617–624. IEEE, 2011.

C. Fouard, A. Deram, Y. Keraval, and E. Promayon. CamiTK: a modular framework integrating visualization, image processing and biomechanical modeling. In *Soft tissue biomechanical modeling for computer assisted surgery*, pages 323–354. Springer, 2012.

Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 148–156, 1996.

J. Friedman, T. Hastie, R. Tibshirani, et al. Additive logistic regression: a statistical view of boosting. *The annals of statistics*, 28(2):337–407, 2000.

J. H. Friedman. A recursive partitioning decision rule for nonparametric classification. *IEEE Transactions on Computers*, (4):404–408, 1977.

J. H. Friedman. A tree-structured approach to nonparametric multiple regression. In *Smoothing techniques for curve estimation*, pages 5–22. Springer, 1979.

J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

J. Gall and V. Lempitsky. Class–specific hough forests for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2009.*, pages 1022–1029, 2009.

J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(11):2188–2202, 2011.

H. E. Garrett. *Statistics in psychology and education.* Longmans, Green and Company, 1947.

R. Gauriau, R. Cuingnet, R. Prevost, B. Mory, R. Ardon, D. Lesage, and I. Bloch. A generic, robust and fully-automatic workflow for 3D CT liver segmentation. In *Abdominal Imaging. Computation and Clinical Applications*, pages 241–250. Springer, 2013.

R. Gauriau, R. Cuingnet, D. Lesage, and I. Bloch. Multi-organ localization combining global-to-local regression and confidence maps. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2014*, pages 337–344. Springer, 2014.

E. Geremia, O. Clatz, B. H. Menze, E. Konukoglu, A. Criminisi, and N. Ayache. Spatial decision forests for MS lesion segmentation in multi-channel magnetic resonance images. *NeuroImage*, 57(2):378–390, 2011.

P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon. Efficient regression of general-activity human poses from depth images. In *2011 International Conference on Computer Vision*, pages 415–422, Nov 2011.

B. Glocker, J. Feulner, A. Criminisi, D. R. Haynor, and E. Konukoglu. Medical image computing and computer-assisted intervention – miccai 2012. chapter Automatic Localization and Identification of Vertebrae in Arbitrary Field-of-View CT Scans, pages 590–598. Springer Berlin Heidelberg, 2012.

G. A. Gorry and G. O. Barnett. Experience with a model of sequential diagnosis. *Computers and Biomedical Research*, 1(5):490–507, 1968.

L. Gu and T. Peters. 3d segmentation of medical images using a fast multistage hybrid algorithm. *International Journal of Computer Assisted Radiology and Surgery*, 1(1):23–31, 2006.

D. Han, Y. Gao, G. Wu, P.-T. Yap, and D. Shen. Robust anatomical landmark detection for MR brain image registration. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2014*, pages 186–193. Springer, 2014.

H. Hao, C.-L. Liu, and H. Sako. Comparison of genetic algorithm and sequential search methods for classifier subset selection. In *Proceedings of the*

Seventh International Conference on Document Analysis and Recognition-Volume 2, pages 765–769. IEEE Computer Society, 2003.

T. K. Ho. Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282. IEEE, 1995.

T. K. Ho. The random subspace method for constructing decision forests. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(8): 832–844, 1998.

A. Ittner and M. Schlosser. Discovery of relevant new features by generating non-linear decision trees. In *International conference on knowledge discovery and data mining*, pages 108–113, 1996.

A. J. Izenman. *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning (Springer Texts in Statistics)*. Springer, 1 edition, 2008.

P. Jaccard. The distribution of the flora in the alpine zone. *New phytologist*, 11(2):37–50, 1912.

H. J. Johnson, M. McCormick, L. Ibáñez, and T. I. S. Consortium. *The ITK Software Guide*. Kitware, Inc., third edition, 2013. http://www.itk.org/ItkSoftwareGuide.pdf.

T. R. Johnson, B. Krauss, M. Sedlmair, M. Grasruck, H. Bruder, D. Morhard, C. Fink, S. Weckbach, M. Lenhard, B. Schmidt, et al. Material differentiation by dual energy CT: initial experience. *European radiology*, 17(6):1510–1517, 2007.

G. V. Kass. An exploratory technique for investigating large quantities of categorical data. *Applied statistics*, pages 119–127, 1980.

B. M. Kelm, S. Mittal, Y. Zheng, A. Tsymbal, D. Bernhardt, F. Vega-Higuera, S. K. Zhou, P. Meer, and D. Comaniciu. Detection, grading and classification of coronary stenoses in computed tomography angiography. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2011*, pages 25–32. Springer, 2011.

J. Kim, Y. Duchin, G. Sapiro, J. Vitek, and N. Harel. Clinical deep brain stimulation region prediction using regression forests from high-field mri. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 2480–2484. IEEE, 2015.

S. Klein, M. Staring, K. Murphy, M. A. Viergever, and J. P. Pluim. Elastix: a toolbox for intensity–based medical image registration. *IEEE transactions on medical imaging*, 29(1):196–205, 2010.

I. Kostrikov and J. Gall. Depth sweep regression forests for estimating 3d human pose from images. In *British Machine Vision Conference, BMVC*, volume 1, pages 5–18. BMVC Press, 2014.

R. Kothari and M. Dong. Decision trees for classification: A review and some new results. *Pattern Recognition: From Classical to Modern Approaches, edited by SK Pal and A. Pal (World Scientific, Singapore, 2001)*, pages 169–184, 2000.

S. B. Kotsiantis. Decision trees: a recent overview. *Artificial Intelligence Review*, 39(4):261–283, 2013.

S. Lavallee and P. Cinquin. Computer assisted medical interventions. In *3D Imaging in Medicine*, pages 301–312. Springer, 1990.

R. S. Ledley and L. B. Lusted. Reasoning foundations of medical diagnosis symbolic logic, probability, and value theory aid our understanding of how physicians reason. *Science*, 130(3366):9–21, 1959.

V. Lempitsky, M. Verhoek, J. A. Noble, and A. Blake. Random forest classification for automatic delineation of myocardium in real-time 3d echocardiography. In *Functional Imaging and Modeling of the Heart*, pages 447–456. Springer, 2009.

R. I. Lerman and S. Yitzhaki. A note on the calculation and interpretation of the gini index. *Economics Letters*, 15(3):363–368, 1984.

B. Li, R. Zhou, C. Yang, M. Q.-H. Meng, G. Xu, and C. Hu. Capsule endoscopy images classification by random forests and ferns. In *Information Science and Technology (ICIST), 2014 4th IEEE International Conference on*, pages 414–417. IEEE, 2014.

Y. Li, S. Wang, and X. Ding. Person-independent head pose estimation based on random forest regression. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 1521–1524. IEEE, 2010.

T.-S. Lim, W.-Y. Loh, and Y.-S. Shih. An empirical comparison of decision trees and other classification methods. Technical Report 979, University of Wisconsin, 1998.

T.-S. Lim, W.-Y. Loh, and Y.-S. Shih. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine learning*, 40(3):203–228, 2000.

C. Lindner, S. Thiagarajah, J. M. Wilkinson, G. A. Wallis, and T. F. Cootes. Accurate fully automatic femur segmentation in pelvic radiographs using regression voting. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012*, pages 353–360. Springer Berlin Heidelberg, 2012.

W.-Y. Loh. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23, 2011.

W.-Y. Loh and Y.-S. Shih. Split selection methods for classification trees. *Statistica sinica*, pages 815–840, 1997.

W.-Y. Loh and N. Vanichsetakul. Tree-structured classification via generalized discriminant analysis. *Journal of the American Statistical Association*, 83(403):715–725, 1988.

G. Louppe. *Understanding Random Forests: From Theory to Practice*. PhD thesis, University of Liège, Belgium, 2014. arXiv:1407.7502.

P. Mah, T. E. Reeves, and W. D. McDavid. Deriving hounsfield units using grey levels in cone beam computed tomography. *Dentomaxillofacial Radiology*, 39(6):323–335, 2010.

H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947.

M. Mibulumukini, N. Riche, M. Mancas, B. Gosselin, and T. Dutoit. Biologically plausible context recognition algorithms. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 2612–2616. IEEE, 2013.

F. Milletari, M. Yigitsoy, and N. Navab. Left ventricle segmentation in cardiac ultrasound using hough-forests with implicit shape and appearance priors. In *MICCAI Challenge on Endocardial Three-dimensional Ultrasound Segmentation (CETUS)*, pages 49–56, September 2014.

F. Milletari, S.-A. Ahmadi, C. Kroll, C. Hennersperger, F. Tombari, A. Shah, A. Plate, K. Boetzel, and N. Navab. Robust segmentation of various anatomies in 3D ultrasound using hough forests and learned data representations. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015*, pages 111–118. Springer, 2015.

A. Montillo and H. Ling. Age regression from faces using random forests. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 2465–2468, Nov 2009.

F. Moosmann, B. Triggs, and F. Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *Twentieth Annual Conference on Neural Information Processing Systems (NIPS'06)*, pages 985–992. MIT Press, 2007.

F. Moosmann, E. Nowak, and F. Jurie. Randomized clustering forests for image classification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(9):1632–1646, 2008.

J. Morgan and R. Messenger. THAID: a sequential search program for the analysis of nominal scale dependent variables. *Survey Research Center, Institute for Social Research, University of Michigan.[251]*, 1973.

J. N. Morgan and J. A. Sonquist. Problems in the analysis of survey data, and a proposal. *Journal of the American statistical association*, 58(302): 415–434, 1963.

A. Mouton, T. P. Breckon, G. T. Flitton, and N. Megherbi. 3D object classification in baggage computed tomography imagery using randomised clustering forests. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 5202–5206. IEEE, 2014.

E. M. Mugambi, A. Hunter, G. Oatley, and L. Kennedy. Polynomial-fuzzy decision tree structures for classifying medical data. *Knowledge-Based Systems*, 17(2):81–87, 2004.

K. P. Murphy. *Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning series)*. The MIT Press, first edition, 2012.

S. K. Murthy. Automatic construction of decision trees from data: A multi-disciplinary survey. *Data mining and knowledge discovery*, 2(4):345–389, 1998.

S. K. Murthy, S. Kasif, S. Salzberg, and R. Beigel. Oc1: A randomized algorithm for building oblique decision trees. In *Proceedings of AAAI*, volume 93, pages 322–327, 1993.

S. K. Murthy, S. Kasif, and S. Salzberg. A system for induction of oblique decision trees. *Journal of artificial intelligence research*, 2:1–32, 1994.

T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996.

A. Oliva and A. Torralba. The role of context in object recognition. *Trends in cognitive sciences*, 11(12):520–527, 2007.

M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *Computer Vision and Pattern Recognition, 1997. IEEE Computer Society Conference on*, pages 193–199. IEEE, 1997.

O. Oshin, A. Gilbert, J. Illingworth, and R. Bowden. Action recognition using randomised ferns. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 530–537. IEEE, 2009.

T. M. Oshiro, P. S. Perez, and J. A. Baranauskas. How many trees in a random forest? In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 154–168. Springer, 2012.

N. Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.

M. Özuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

M. Özuysal, M. Calonder, V. Lepetit, and P. Fua. Fast keypoint recognition using random ferns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(3):448–461, 2010.

N. R. Pal and S. K. Pal. A review on image segmentation techniques. *Pattern recognition*, 26(9):1277–1294, 1993.

C. P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Computer vision, 1998. sixth international conference on*, pages 555–562. IEEE, 1998.

S. D. Pathak, A. Criminisi, J. Shotton, S. White, D. Robertson, B. Sparks, I. Munasinghe, and K. Siddiqui. Validating automatic semantic annotation of anatomy in dicom CT images. In *SPIE Medical Imaging*, pages 796704:1–11. International Society for Optics and Photonics, 2011.

O. Pauly. *Random Forests for Medical Applications*. PhD thesis, Technical University of Munich, Germany, 2012.

O. Pauly, B. Glocker, A. Criminisi, D. Mateus, A. M. Möller, S. Nekolla, and N. Navab. Fast multiple organ detection and localization in whole-body mr dixon sequences. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2011*, pages 239–247. Springer, 2011.

D. L. Pham, C. Xu, and J. L. Prince. Current methods in medical image segmentation 1. *Annual review of biomedical engineering*, 2(1):315–337, 2000.

J. Quinlan. Discovering rules form large collections of examples: A case study. *Expert Systems in the Micro Electronic Age. Edinburgh Press: Edingburgh*, 1979.

J. R. Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

M. G. Roberts, T. F. Cootes, and J. E. Adams. Automatic location of vertebrae on DXA images using random forest regression. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012*, pages 361–368. Springer, 2012.

M. Robnik-Šikonja. Improving random forests. In *Machine Learning: ECML 2004*, pages 359–370. Springer, 2004.

J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso. Rotation forest: A new classifier ensemble method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(10):1619–1630, 2006.

L. Rokach. Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography. *Computational Statistics & Data Analysis*, 53(12):4046–4072, 2009.

L. Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33 (1-2):1–39, 2010.

S. R. Safavian and D. Landgrebe. A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3):660–674, 1991.

W. Schroeder, K. Martin, and B. Lorensen. *Visualization Toolkit: An Object–Oriented Approach to 3D Graphics, 4th Edition.* Kitware, 4th edition, 12 2006.

C. Shannon and W. Weaver. *The Mathematical Theory of Communication.* University of Illinois Press, 1949.

N. Sharma, L. M. Aggarwal, et al. Automated medical image segmentation techniques. *Journal of medical physics*, 35(1):3, 2010.

Y.-S. Shih. Families of splitting criteria for classification trees. *Statistics and Computing*, 9(4):309–315, 1999.

J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 81(1):2–23, 2009.

J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1297–1304, June 2011.

J. Shotton, T. Sharp, P. Kohli, S. Nowozin, J. Winn, and A. Criminisi. Decision jungles: Compact and rich models for classification. In *Advances in Neural Information Processing Systems*, pages 234–242, 2013.

A. A. Taha and A. Hanbury. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. *BMC medical imaging*, 15: 1–29, 2015.

E. Taillant, J.-C. Avila-Vilchis, C. Allegrini, I. Bricault, and P. Cinquin. CT and MR compatible light puncture robot: Architectural design and first experiments. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 145–152. Springer, 2004.

Y. Tang, Z. Sun, and T. Tan. Real-time head pose estimation using random regression forests. In *Biometric Recognition*, pages 66–73. Springer, 2011.

A. Tejani, D. Tang, R. Kouskouridas, and T.-K. Kim. Latent-class hough forests for 3D object detection and pose estimation. In *Computer Vision–ECCV 2014*, pages 462–477. Springer, 2014.

J. Troccaz. Computer and robot-assisted medical intervention. In *Springer Handbook of Automation*, pages 1451–1466. Springer, 2009.

M. Tschannen, L. Vlachopoulos, C. Gerber, G. Székely, and P. Fürnstahl. Regression forest-based automatic estimation of the articular margin plane for shoulder prosthesis planning. *Medical image analysis*, 31:88–97, 2016.

P. E. Utgoff and C. E. Brodley. Linear machine decision trees. Technical report, Amherst, MA, USA, 1991.

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I:511–I:518. IEEE, 2001.

D. Wang, X. Liu, L. Jiang, X. Zhang, and Y. Zhao. Rough set approach to multivariate decision trees inducing. *Journal of Computers*, 7(4):870–879, 2012.

H. R. Warner, A. F. Toronto, and L. G. Veasy. Experience with baye's theorem for computer diagnosis of congenital heart disease*. *Annals of the New York Academy of Sciences*, 115(2):558–567, 1964.

S. M. Weiss, C. A. Kulikowski, S. Amarel, and A. Safir. A model-based method for computer-aided medical decision-making. *Artificial intelligence*, 11(1-2):145–172, 1978.

G. Yan and B. Wang. An automatic kidney segmentation from abdominal ct images. In *Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference on*, volume 1, pages 280–284. IEEE, 2010.

N. Zemiti, I. Bricault, C. Fouard, B. Sanchez, and P. Cinquin. LPR: A CT and MR-compatible puncture robot to enhance accuracy and safety of image-guided interventions. *Mechatronics, IEEE/ASME Transactions on*, 13(3):306–315, 2008.

O. Zettinig, A. Shah, C. Hennersperger, M. Eiber, C. Kroll, H. Kübler, T. Maurer, F. Milletarì, J. Rackerseder, C. S. zu Berge, et al. Multimodal image-guided prostate fusion biopsy based on automatic deformable registration. *International journal of computer assisted radiology and surgery*, 10(12):1997–2007, 2015.

X. Zhen, Z. Wang, A. Islam, M. Bhaduri, I. Chan, and S. Li. Direct estimation of cardiac bi-ventricular volumes with regression forests. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2014*, pages 586–593. Springer, 2014.

X. Zhen, Z. Wang, A. Islam, M. Bhaduri, I. Chan, and S. Li. Multi-scale deep networks and regression forests for direct bi-ventricular volume estimation. *Medical image analysis*, 30:120–129, 2016.

K. H. Zou, S. K. Warfield, A. Bharatha, C. M. Tempany, M. R. Kaus, S. J. Haker, W. M. Wells, F. A. Jolesz, and R. Kikinis. Statistical validation of image segmentation quality based on a spatial overlap index 1: Scientific reports. *Academic radiology*, 11(2):178–189, 2004.

Index