DEGREE PROJECT IN COMPUTER SCIENCE AND ENGINEERING, SECOND CYCLE, 30 CREDITS

*STOCKHOLM, SWEDEN 2019*

# Towards unification of organ labeling in radiation therapy using a machine learning approach based on 3D geometries

**GIORGIO RUFFA**

**KTH ROYAL INSTITUTE OF TECHNOLOGY**
**SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE**

# Towards unification of organ labeling in radiation therapy using a machine learning approach based on 3D geometries

GIORGIO RUFFA

Master in Computer Science and Engineering (II226X)
Date: July 2, 2019
Supervisor: Amir Payberah
Examiner: Vladimir Vlassov
School of Electrical Engineering and Computer Science
Host company: RaySearch Laboratories AB
Supervisors: Marco Trincavelli and Fredrik Löfman
Swedish title: Mot enad organnamngivning i strålterapi med hjälp av en maskininlärningsmetod baserad på 3D geometrier

# Abstract

In radiation therapy, it is important to control the radiation dose absorbed by *Organs at Risk* (OARs). The OARs are represented as 3D volumes delineated by medical experts, typically using computed tomography images of the patient. The OARs are identified using user-provided text labels, which, due to a lack of enforcement of existing naming standards, are subject to a great level of heterogeneity. This condition negatively impacts the development of procedures that require vast amounts of standardized data, like organ segmentation algorithms and inter-institutional clinical studies. Previous work showed that supervised learning using deep-learning classifiers could be used to predict OARs labels. The input of this model was composed of 2D contours of the OARs, while the output was a standardized label. In this work, we expanded this approach by qualitatively comparing the performance of different machine learning algorithms trained on a clinical data set of anonymized prostate cancer patients from the Iridium Kankernetwerk clinic (Belgium). The data set was partitioned in a semi-automatic fashion using a divide-and-conquer-like approach and various 2D and 3D encodings of the OARs geometries were tested. Moreover, we implemented a reject class mechanism to assess if the inference probability yielded by the model could be used as a measure of confidence. The underlining goal was to restrict human intervention to rejected cases while allowing for a reliable and automatic standardization of the remaining ones. Our results show that a random forest model trained on simple 3D-based manually engineered features can achieve the twofold goal of high classification performance and reliable inferences. In contrast, 3D convolutional neural networks, while achieving similar classification results, produced wrong, but confident, predictions that could not be effectively rejected. We conclude that the random forest approach represents a promising solution for automatic OAR labels unification, and future works should investigate its applications on more diversified data sets.

# Sammanfattning

En viktig faktor i strålbehandling är att kontrollera hur mycket av strålningen som absorberas av riskorgan. Med hjälp av medicinska bilder, vanligtvis från datortomografi, konturerar medicinska experter riskorgan som sedan representeras som tredimensionella volymer. Riskorganens typ anges via manuell namngivning från den medicinska experten. Detta samt bristande efterlevnad av namngivningsprotokoll, har resulterat i hög heterogenitet bland angivna organnamn. Där denna heterogenitet bromsar utvecklingen av metoder som kräver stora mängder standardiserade data, såsom organsegmenteringsalgoritmer, därutöver försvårar det studier som utförs på intraklinisk basis. Tidigare arbete inom fältet för namngivning av konturerade organ har visat att övervakad inlärning med djupinlärningsklassificerare kan användas för att automatiskt identifiera riskorgannamn. Indata till denna modell bestod av tvådimensionella riskorgankonturer och utdata bestod av standardiserade riskorgannamn. Detta arbete bygger vidare på det tidigare tillvägagångssättet genom att kvalitativt jämföra och utvärdera olika maskininlärningsalgoritmers prestanda för samma ändamål. Algoritmerna tränades på en klinisk datamängd bestående av anonymiserade prostatacancerpatienter från den belgiska kliniken Iridium Kankernetwerk. Datamängden partitionerades på ett semi-automatiserat vis med hjälp av ett tillvägagångssätt inspirerat av söndra-och-härska-tekniken och flera typer av två- och tredimensionell representationer av patientbilderna testades. Vidare implementerades en mekanism för att utvärdera om inferenssannolikheten från modellen kunde användas som ett tillförlitligt konfidensmått. Med bakomliggande mål att enbart behöva involvera mänsklig inblandning i de fall som bedöms som extra svåra av mekanismen och på så sätt åstadkomma en automatisk standardiseringen av resterande fall. Resultaten visar att en random forest-modell som tränats på enkla och manuellt designade variabler kan uppnå de två uppsatta målen: hög klassificeringsprestanda och pålitlig inferens. Jämförelsevis lyckades tredimensionella faltningsnätverk uppnå likvärdiga klassificeringsresultat men producerade felaktiga prediktioner som inte var möjliga att avfärda på ett effektivt sätt. Vår slutsats är att den framtagna random forest-metoden är en lovande lösning för automatisk och standardiserad namngivning av riskorgan. Framtida arbete bör utvärdera metoden på data med större variation.

# Acknowledgements

I would like to express my very great appreciation to Geert De Kerf and Piet Dirix from Iridium Kankernetwerk in Belgium, for allowing the usage of anonymized patients data under the data transfer agreement in place with RaySearch Laboratories AB covering this project. I would also like to thank Yi Wang, from the Massachusetts General Hospital, whose vast experience brought an incredibly valuable clinical perspective to this work.

My very great appreciation goes to Amir Payberah, Marco Trincavelli and Fredirk Löfman, for the assistance and guidance through all the project. I am also thankful to all the people in RaySearch, especially the members of the machine learning department, who made me feel as part of the team from day one. An honorable mention goes to Marcus Nordström, Karl Berggren, and Hanna Gruselius for the support and proof reading.

This work comes as an important milestone in a life-changing experience that started two years ago. It was not an easy path to follow, especially at the beginning, but if I managed to go through it, and eventually enjoy it, it was mostly for the unconditional support provided by Lina Balzan, Franco Ruffa and Silvia Carretta, the latter being the only person who was crazy enough to follow me in this journey despite the great sacrifices.

I would like to offer my special thanks to my Spanish second-family: Eva, Fernando, Laura, Carlos, Miriam and all the other, more or less official, residents of Ferraz 9. Without your joy and generosity this experience would not have been so memorable.

# Contents


**1 Introduction** **2**
- 1.1 Problem Formulation . . . 4
- 1.2 Purpose . . . 6
- 1.3 Goal . . . 7
- 1.4 Research Questions and Hypothesis . . . 7
- 1.5 Research Methodology . . . 8
- 1.6 The Importance of a Geometric Approach . . . 9
- 1.7 Benefits, Ethics and Sustainability . . . 10
- 1.8 Limitations . . . 11
- 1.9 Outline . . . 12

**2 Background** **13**
- 2.1 Data in Radiation Therapy . . . 13
  - 2.1.1 CT Scans . . . 13
  - 2.1.2 RT Patient Modeling . . . 17
  - 2.1.3 The DICOM Standard . . . 20
- 2.2 Naming Standards . . . 22
  - 2.2.1 Ontologies . . . 22
  - 2.2.2 AAPM TG-263 . . . 23
- 2.3 Shape Descriptor and Normalized Central Moments . . . 24
- 2.4 Machine Learning Algorithms . . . 26
  - 2.4.1 Supervised Learning and Classification . . . 26
  - 2.4.2 Training Process and Classification Metrics . . . 27
  - 2.4.3 Decision Trees . . . 29
  - 2.4.4 Random Forest . . . 30
  - 2.4.5 3D Convolutional Neural Networks . . . 30
- 2.5 Machine Learning in Medical Imaging . . . 31

**3 Method** **35**
3.1 Raw Data Encoding . . . 37
3.2 Raw Data Set Analysis . . . 39
3.3 Ground Truth . . . 40
3.4 Encoding and Model Training . . . 45
3.4.1 2D Transverse Slices . . . 45
3.4.2 3D Feature Engineering . . . 49
3.4.3 3DCNN . . . 53
3.5 Classification Evaluation . . . 55
3.6 Inference Probability Evaluation . . . 56

**4 Results** **59**
4.1 Baseline: DT on 2D slices . . . 59
4.2 Random Forest: an ensemble of decision trees . . . 61
4.3 Slice Based Majority Voting: towards 3D features . . . 62
4.4 Classification of 3D Volumes . . . 64
4.5 Inference Probability as Confidence Measure . . . 68
4.6 Rejected Cases . . . 72
4.7 Summary . . . 76

**5 Discussion** **77**

**A Appendix** **81**
A.1 Frequent Item Sets for OAR Names Detection . . . 81
A.2 Raw Data Set Additionals . . . 82
A.3 Target ROI Names Identification . . . 83
A.4 3D Engineered Features . . . 85
A.5 Extended Results . . . 88
A.6 RF Hyper-parameters Tuning . . . 89
A.7 Experimental Setup and Running Times . . . 92
A.8 VOXNet Training Process . . . 94

**Bibliography** **95**

# List of Acronyms and Abbreviations

**3DCRT** 3D Conformal Radiotherapy

**CT** Computed Tomography

**DL** Deep Learning

**DT** Decision Tree

**EHR** Electronic Health Record (synonim of EMR)

**EMR** Electronic Medical Record

**FN** False Negative[s]

**FP** False Positive[s]

**IOD** Information Object Definition

**IRMA** Content-Based Image Retrieval in Medical Applications

**LR** Logistic Regression

**ML** Machine Learning

**MOS** Multy Organ Segmentation

**MRI** Magnetic Resonance Imaging

**NCMs** Normalized Central Moments

**OARs** Organs At Risk

**PACS** Picture Archiving and Communication System

**RF** Random Forest

**ROC** Receiver Operating Characteristic curve

**RT** Radiation Therapy

# Chapter 1

# Introduction

"*Artificial intelligence will revolutionize health care!*"

These kinds of claims have become increasingly frequent both in the health care industry and the academia [1, 2, 3, 4, 5, 6, 7]. As bold as they may sound, they are supported by the unprecedented series of advancements achieved in the last decade by artificial intelligence and in particular by machine learning (ML) and deep learning (DL). These advancements impacted multiple fields like computer vision, machine-translation, natural language processing, and representation learning [8, 9, 10, 11]. This, along with the lower cost of storing and processing of data, has marked a shift in many industries from first-principle based models to heavily data-based ones [12].

No straightforward reasons seem to obstacle the adoption of such techniques in the health care world. In fact, various advancement of computer vision and ML have been introduced in the medical domain to successfully solve problems of very diversified nature like disease detection, diagnosis, work-flow management, and, more generally, medical imaging problems [13, 14, 15, 16, 17, 18, 19].

These results are a promising starting point for a wider adoption of ML techniques in the health care sector. However, the unique challenges of healthcare data management - like an inherent heterogeneity and the need to cope with legal and ethical constraints [20] - brought the medical ML community to a condition known as *Data Starvation*; that is "*[...] an urgent need to find better ways to collect, annotate, and reuse medical imaging data*" [21]. This situation spurred the necessity for a more data-savvy approach towards a scenario "*[...]in which the best treatment decisions are computationally learned from electronic health record data*" [22].

As naïve as it may sound, data-based approaches are only as good as the

data they have been fed. In particular, modern medical data sets are composed only by few hundreds patients cases.[1] This is in stark contrast with the incredible variety and size of well established computer vision data sets like ImageNet [23], CIFAR [24] or MSCoCo [25].

The consequence of having such a restricted ground truth has an effect on the models' ability to be employed in real-world scenarios, because of their reduced generalization capabilities [21, 26]. In addition to the limited sample size, clinical annotations compose the most important building block of medical imaging ground truth[2] and they are mainly inserted by medical experts in free-text form. Although standards exist, they are seldom followed by the parties involved, resulting in an important heterogeneity of labeled data [21].

This condition may come as a surprise, considering that medical imaging, and radiology in particular, was one of the first sectors to introduce a widely adopted standard for Digital Imaging and Communications in Medicine (DICOM) [27, 28]. The application of the DICOM standard effectively enabled radiologist and medical experts to store, retrieve, and exchange medical data safely and reliably. Unfortunately, this standard is not designed to cope with the data quality constraint typical of an ML data set since, for example, many data fields may be filled incorrectly or not be filled at all [21, 29].[3] While these issues are not an impediment in daily medical practice, they may constitute a major hassle when attempting to merge or federate data sets while trying to create a more comprehensive ground truth for the ML task at hand [30].

On the other hand, the wide adoption of the DICOM standard has fostered the storage and aggregation of large sets of medical data in specialized archiving systems, the so-called PACS (Picture Archiving and Communication System). Thus, a controversial situation is originated: there is, in theory, a great abundance of medical data, but the lack of standardization renders them unusable for any ML application [21, 22].

Consequently, this work focuses on the investigation of automatic standardization techniques of medical data to streamline their usage in ML applications. In particular, we take into consideration the standardization of radiation oncology data to address the heterogeneity generated by "*Organs At Risk*" (OARs) labels. The next section further explains the problem and the

[1] Multiple cases may belong to the same patient, being visited multiple times.

[2] Annotations are a special kind of meta-data that pertain to particular regions contained in a medical image (being it the diagnoses, anatomical or pathological). An example in the radiology field could be the area of an X-RAY image delimiting the left lung of the patient, labeled as "`LT_LUNG`".

[3] Some with standard and others with proprietary format

data structures used in radiation oncology.

## 1.1 Problem Formulation

Radiation therapy (or radiotherapy) (RT) is a kind of medical therapy that uses ionizing radiation as part of a cancer treatment procedure [31]. More precisely, "*[it] aims to sculpt the optimal isodose on the tumour volume while sparing normal tissues*" [32]. To achieve this goal medical experts identify a series of important regions within the patient's body. This is done on a 3D representation of the patient obtained from medical images (see section 2.1.1). These regions are eponymously called *Regions of Interest* (ROIs).[4] These regions define precise 3D volumes inside the patient and are of different types. In some regions, referred to as *target regions*, the radiation dose must be maximized in order to treat the tumor and lower the chances of re-appearance of the disease. In other regions, the dose must be minimized as much as possible. This is especially the case for organs, which are indeed called *Organs at Risk*. If the target region partially includes an organ, then not all the volume of the organ should receive minimum radiation. For these cases, a special volume containing the part of the organ outside the target region is created and commonly named an *avoidance region*. The process of delineating all the relevant ROIs for the treatment is called *patient modeling*. When it is terminated, a single patient's representation can contain from ten to thirty ROIs, and to each of them the medical expert assigns a free text label, normally referred to as the *ROI name*.

Between all the ROIs, OARs have particular importance when it comes to daily clinical practice. Their location is identified with a process known as *multi-organ segmentation* (MOS), during which medical images (usually from a CT or MRI[5]) are analyzed and the exact 3D position of the organs is delineated. Performing MOS manually is a lengthy and time-consuming process, hence a variety of automatic and semi-automatic methods have been developed, the vast majority of these being statistically based [33, 34, 35, 36, 37, 38, 39]. As anticipated in section 1, an extensive and high-quality ground truth data set is needed for these methods to be robust [21]. MOS is only one example of data-intensive problems in radiation oncology requiring such a high quality and quantity of data, as a matter of fact all ML application share

[4] Section 2.1.2 explains in great detail how ROIs are defined and the taxonomy used to categorize them.

[5] Plese refer to section 2.1 for further details.

the same constraints (for more on the application of ML techniques in the medical domain see section 2.5).

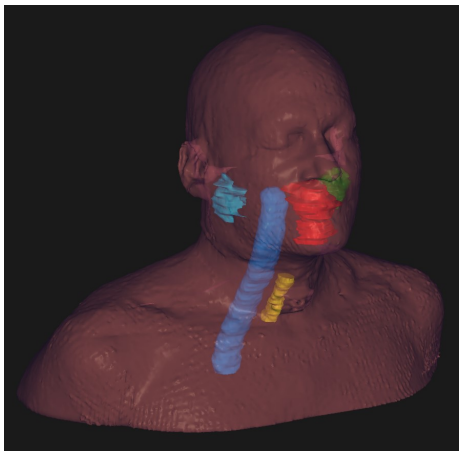


*Figure 1.1: 3D Multi-organ segmentation performed on an head-and-neck disease site from a CT scan of the cetuximab data set [40]. Segmented organs are: spinal cord (blue), larynx (yellow), oral cavity (red), right parotid gland (turquoise), left parotid gland (green).*

Luckily, many medical institutions possess and maintain large storage systems called PACS where patients folders containing already segmented CT images are stored. These archiving systems have the upside of containing ground truth segmentation that has already been verified by a medical expert, which, in theory, perfectly suits an ML task. Unfortunately though, the regulatory and technical constraints concerning health-care data management [20] result in two major obstacles to ML applications: the data sets usually contain few hundreds patients, and the free-text nature of the labels associated to organs are often heterogeneous and inconsistent in many different levels. Different institutions, hospitals and even medical experts within the same institution may use different naming conventions, and lexicographical errors and abbreviations may be recurrent [41] [42], as well as highly locale-dependent dictionaries.

Using standard data-warehouse terminology we can classify the different sources of heterogeneity as "*single-source instance level*" and "*multi-source schema level*" data quality problems [43]. The former refers to "*errors and inconsistencies that cannot be prevented at the schema level*" [43], like misspellings, duplicates, and contradictory values. The latter concerns problems that arise when multiple different data sources are integrated. An example that particularly fits to our case is constituted by naming conflicts: the same

*Table 1.1: Examples of names associated to the same anatomic structure in twelve different clinics [41].*

| Structure | Examples |
|---|---|
| Left Optic Nerve | Lt Optic Nerve, OPTICN_L, OPTNRV_L, optic_nrv_l, L_optic_nerve, OPTIC_NRV_L, OpticNerve_L, LOPTIC, OpticNerve_L (3), Lef Optic Nerve, ON_L |
| Left Lung | Lt Lung, Lung_L(4), LUNG_L(3), lung_l, L_lung, LLUNG, L Lung |
| Both Lungs | Lungs(2), LUNGs, LUNG_TOTAL, lung_total, combined_lung, LUNG, LUNGS(2), Lung, BilatLung, Lung_Both |

organ called differently in different patient's folders. To give a practical example, table 1.1 reports some examples of different names associated to the same OAR, based on the work of the task group 263 of the *American Association of Physicist in Medicine (AAPM) [41]*.

To rephrase the problem in a rather pragmatic way: for many institutions it is not possible nowadays to formulate a simple query like this: "*Retrieve all the contours of the left lung of all our patients*", as the left lung may have rather different names. Let alone trying to perform such a query on two institutions' PACS located in two different countries with different languages, for example a Dutch and American clinic.

## 1.2 Purpose

The focus of this work is to compare different supervised ML-based approach to classify OAR volumes based only on their geometric representation. This would enable the association of the volume with a unique OAR label, thus performing the standardization of the OAR data set. The complete process is taken under consideration: the initial analysis of a real-world clinical data set, the partition of the data set in a divide-and-conquer-like fashion (using reliable patients for training and more complex cases for testing), the training of the model along with its performance evaluation, and, most notably, the assessment of the quality of the obtained inferences. Despite its importance, this last aspect is seldom analyzed, resulting in excessive attention posed to the

performance of the model, and not considering how much the model inference could be trusted in a production environment.

We decided to restrict our efforts only to OARs labels because they are central in many RT operations, like MOS or treatment planning. But also because once standardized they can be easily used to identify the disease site and detect the kind of tumor being treated by comparing the position of target volume with the one of the OARs in its proximity.[6] A complete list of possible benefits is reported in section 1.7. Moreover, concentrating on OARs reduces sensibly the scope of the project and allows for a comprehensive study of different approaches.

## 1.3 Goal

The goal is to identify a viable strategy for automatic data cleaning of OAR labels in oncology data sets. Also, to suggest possible directions of development of more comprehensive systems concerning data-quality assessment in radiation oncology like anomaly and outlier detection, target and avoidance structure standardization (see section 2.1.2), and oncology information retrieval systems.

## 1.4 Research Questions and Hypothesis

If consistency is lacking at annotation level, we cannot say the same for human bodies. In fact, a certain level of consistency is to be expected. We can safely say that, excluding peculiar clinical cases, organs look alike in different patients. Most importantly, they belong to a well-defined spatial context: that is, their respective position is the same in all human bodies. As an example, the rectum is always below the bowel, the bladder is always between the two femurs, the left parotid gland is always on the left of the spinal cord. This, together with the assumption that the MOS is performed consistently by a medical expert, opens two tightly coupled questions:

**Research Question 1**

Is it possible to exploit OARs' contours spatial consistency to enable unique classification through an ML algorithm, hence enforce unambiguous labeling?

[6] It may seem odd, but the actual information of the kind of tumor being treated is rarely contained in the patient treatment plan.

Considering the label standardization aspect is not enough though. Any obtained model is subject to the risk of errors at inference time. Hence, the result obtained by such a system must go hand-in-hand with an estimate of its confidence. To put it more clearly, how "trustworthy" are the system's inferences? Is the model able to discriminate between a risky prediction and an extremely confident one?

In our particular case:

**Research Question 2**

Is it possible to use the inference probability yielded by the ML algorithm as a measure of confidence?

Or, posed more pragmatically: can we establish an inference probability threshold above which all the inferences of the model can be safely trusted and below which they have to be rejected? Hence signaling the need for human expert intervention to discriminate doubtful cases. It must be stated that the target of this study is not to reach full automatic standardization, but to reduce human intervention only to cases that necessitate it.

Concerning the first question, our hypothesis is that it will be possible to build an ML model capable of correctly classifying the OARs based on their unique features and their positioning in the spatial context.

While, for the second question, in case the aforementioned model - or set of models - is able to produce inference probability, we hypothesize that it will be possible to estimate such a threshold or, at least, obtain its qualitative behavior.

## 1.5 Research Methodology

For the nature of the formulated hypotheses and the resources at our disposal, a qualitative approach is selected. At the same time, it must be also considered that the performed experiments use measurable quantities and, as such, have a non-negligible quantitative aspect. The collected data derive from a real-world phenomenon, which the author has no power to control or influence. For these reasons, an inductive research strategy is preferred, where propositions are derived directly from observations, thus giving the practitioner more freedom in terms of altering the path and direction of the research process [44]. This choice was also driven by the need for solving a practical problem with real-world data. As such, the work was organized in an iterative fashion. This allowed us to obtain results and insights into the complexity of the problem at

an early stage, inductively driving our choices for the next iterations towards a more promising strategy. A total of three iterations were performed in the time at our disposal. Please refer to chapter 3 for an in-depth explanation of the method followed.

Two main principles were used as a reference when selecting an approach: simplicity and pragmatic stance. For the former, special care was put in avoiding un-necessary steps and complications in the methodology. Rather than selecting exotic and complex models, we preferred to start from very simple solutions and add complexity gradually, always justifying the choice following the principle that *"plurality should not be posited without necessity"* [45], thus striving for a *"less moving parts"* solution. For the latter, we made sure that the proposed approach was implementable easily using well tested industry standard open source tools.

The author is a master thesis student belonging to the ML department of Raysearch Laboratories AB, based in Stockholm. His main responsibilities are to build and compare different classification models for OAR geometries, as well as assessing the quality of the obtained inferences. This included various steps: literature review, data collection and encoding, model selection, model training and hyperparameter tuning, model evaluation. The author received feedback and guidance not only from the assigned supervisors, but from the whole ML department, enjoying and appreciating the open mindset and extreme curiosity shown by its members. The author also had the opportunity to work in earnest as a member of the company, participating to various company business meetings and social events, training and seminaries; experiencing the full spectrum of activities and responsibilities the ML engineer position demands. For these reasons, when explaining and justifying the decisions taken during the development of the work, the author prefers to use the *"we"* personal pronoun, instead of referring only to himself as the sole contributor to the decision making process.

## 1.6 The Importance of a Geometric Approach

As explained above, the problem revolves around the heterogeneity in OAR labels, which are encoded as text. A more direct approach would have been to consider only the labels and use ad-hoc mappings to correct and standardize clinical data sets.

Although being more straightforward and less computational intensive, such a solution have numerous issues. First, it has to take into account local dependency, meaning that it has to cope with different languages, used in

an extremely technical fashion. Second, it may be impossible to establish rules valid even for a single clinic, given that different conventions may be followed by different experts (even in a single patient case!). Third, rules do not offer any measure of confidence. They either match or do not match, hence they require to be trusted in an agnostic fashion, with the risk of obtaining incorrect labels that will be blindly considered as correct. Finally, by the words of Santanam et al. [46] *"variability of free-text structure names limits the reliability of such heuristic methods for mapping structure names, thus requiring a great deal of manual quality assurance"* [46].

On the other hand, a geometric approach is immune to language specific aspects and removes from the equation the labels themselves. It builds on the sole assumption that the organs are segmented with a fair degree of coherence. As a plus, an ML approach based on geometries also yields an inference probability that may be used to discard dubious cases, which is one of the hypotheses under investigation in this work.

## 1.7 Benefits, Ethics and Sustainability

Various are the benefits that could arise from OAR label standardization:

- The company will directly benefit from a more reliable ML pipeline, with an increased quality of the ground truth. This is particularly applicable on OAR segmentation tasks which are under constant development and improvement.
- OAR label standardization is a key enabler for treatment planning automation [46, 47, 41, 48], which is already saving a noticeable amount of resources in various clinics and is allowing doctors, physicians, and oncologist to spend more time at direct contact with the patients. When health care is managed mostly with public resources (i.e. in most of Europe), this directly translates in saving tax-payers money while increasing service quality, throughput, and consistency.
- Label standardization will render communication more reliable, which has already shown to be a key factor in reducing the occurrence of incidents and mistreatment in clinical operations [49, 50]
- Facilitation of report generation, information retrieval, plan benchmarking and quality assurance [51, 52, 53].

- If privacy preserving distributed ML will become a reality in the health care field, it would have to be developed on the assumption that the data set is standardized and of high quality. This is because by definition the access to the raw data will be incapacitated.

The ethical aspects concerning this work revolve around the concept of automation and its use. In particular, job loss and accountability may be a major concern. For the former, we think that this is a non-existent issue given the direct outcomes of this work. The standardization of OAR labels is currently stealing precious time from highly trained medical experts that should dedicate their efforts to more important causes, like impacting the life and improving the care of patients. The latter is a much more complicated issue. Automation is not a panacea for all the problems and there are no guarantees that it will be fully fool proof. Errors have to be expected, as some labels will still be incorrect after automatic standardization, that is why a thorough risk-benefit analysis should be performed before employing any automatic solution. We should ask ourselves how many mistreatment cases will be caused by incorrect automatic labeling compared to the ones already generated by the lack of standardization.[7] The author believes the benefits will overcome the risks, and this is exactly why this work focuses also on finding strategies to ensure high quality of inference rather than just good classification scores.

Finally, to evaluate the environmental impact of this work we must have a broader view of possible future outcomes. Good and robust label standardization has the main potential of saving precious man-time. This will also mean that the computational and energy resources that are nowadays allocated to manual operations will be freed and dedicated to more important tasks. The direct consequence of this scenario is a wiser use of energy and resources, but also the reduction of e-waste material. From a more practical standpoint, RaySearch Laboratories AB enforces policies to prevent the waste of natural resources and energy, which contribute to long term environmental sustainability by complying with or exceed all applicable environmental legislation, standards, and industry codes.

## 1.8 Limitations

We focus on a data set containing only prostate cancer cases coming from a single medical institution. This implies that the conclusions drawn from this

[7] To make a parallel with self-driving cars: how many car accidents are generated by automated driving compared to the ones generated by incorrect human driving behaviors.

work may apply only to data sets where the disease site is known beforehand. However, this work delineates a procedure that can be easily extended to other data sets, provided that they contain only one disease site. The nature of the features engineered in section 3.4.2, in particular the relative position towards the body center of mass, suggests the necessity for a dedicated model per disease site. It must be considered that the disease site may be easily inferred from other information contained in the patient folder, like the plan name and the ROI names themselves, or even by classifying the raw CT scan transverse slices [54, 26].

The data set used contains geometric representations that are consistently oriented in the same direction. Patient orientation is encoded in the DICOM standard[8] and is an important and commonly used information in modern clinical operations. Cases of incorrect recording of patient orientation are extremely rare, given that specific medical protocols have been designed to avoid this eventuality [55, 56]. Moreover, automatic techniques exist to detect patient orientation directly from CT images [57].

Finally, as it is explained in detail in chapter 3.4.2, we assume that the segmentation of the OAR is coherent. However, it must be noted that close to the surface of the organ different segmentation protocols may be followed. For this reason, we opt for selecting features that are robust in this regard and that describe the global shape of the organ rather than relying on its surface.

## 1.9 Outline

Chapter 2 gives the reader the necessary background and nomenclature needed to understand the content of this work. In particular, we suggest to dedicate particular attention at section 2.1, in order to acquire the much-needed context and terminology proper of radiation oncology. When possible, surveys and other materials are reported as a support for further studies. Chapter 3 explains the method followed to answer the research questions, while chapter 4 contains all the results collected during the performed experiments. Finally, chapter 5 is dedicated to our final considerations and to the directions that future works should follow.

[8] Field `PatientOrientation` code (0020,0020). See section 2.1.3.

# Chapter 2

# Background

## 2.1 Data in Radiation Therapy

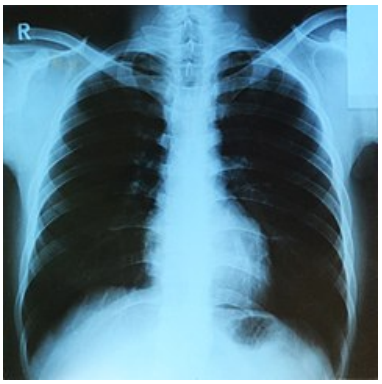
This section covers the most common data encoding and protocols that are used in RT. First, in section 2.1.1, Computed Tomography (CT) medical images are introduced. Basic terminology and conventions are reported, with the aim of helping the reader to orient themselves in the coordinate system used to model the patient. Then, section 2.1.2 explains how CT images are converted into a 3D geometrical representation of the patient and how important structures inside the patient are encoded. Particular attention should be posed to this section as all the data used in this work follows the aforementioned structure. Finally, section 2.1.3 gives a brief introduction to the DICOM standard, which is the *de facto* standard for storage, communication, and retrieval of medical data in the RT domain.

### 2.1.1 CT Scans

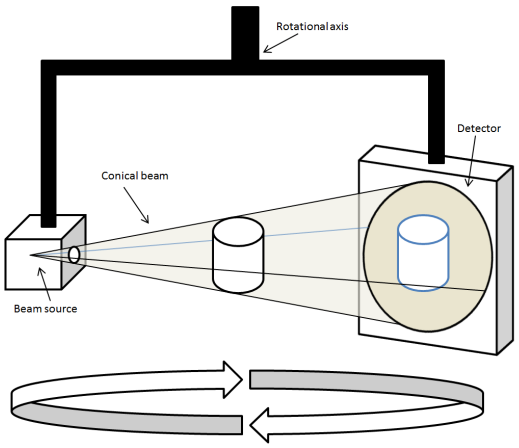
*Computed Tomography* images are the basic diagnostic tool used in *3D Conformal Radiotherapy* (3DCRT), a medical procedure that takes under consideration the full 3D representation of the treatment region within the patient [31]. CT images are central in the diagnosis and localization of a tumor, as well as the delineation of all the structures required to plan the treatment (see section 2.1.2). CT images are generated using X-ray radiation, a type of electromagnetic waves with photon energies in the order of 100 eV to 100 keV.[1] When X-rays pass through a material (for example biological tissue) a certain amount of radiation is absorbed or scattered, which may vary depending on

[1] Which correspond to a wavelength range within $0.1$ and $10$ nanometers.

the kind of material. The amount of radiation that passes through a specific material is described by the linear attenuation and can be measured to obtain a *radiograph*, which is a 2D projection of the internals of the irradiated object (see figure 2.1a).




*(a) Radiograph of human lungs [58].*




*(b) Principle of a CT scan, where the generator and the detector rotates around the object [59].*

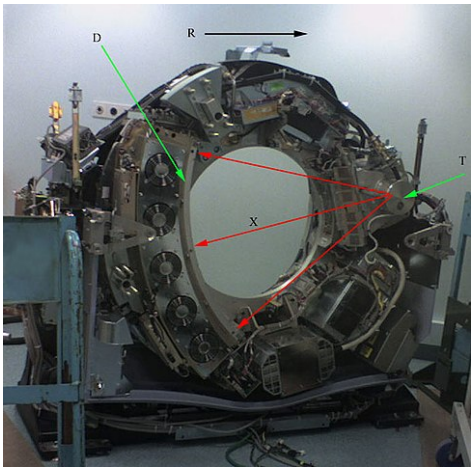
*Figure 2.1*

When generating a CT scan, an X-ray generator rotates around an object on a predefined axis, together with a detector on the opposite side (see figure 2.1b). The object is then translated along the rotation axis in order to cover the region subject to diagnosis, commonly referred to as the *field of view*.

The raw data obtained from the detector at different angles are then combined by a process called *tomographic reconstruction* into a 3D representation of the imaged object. The end result is stored as a series of 2D cross-sectional images called *CT slices* (see figure 2.4a), where each pixel contains a scalar value from $+3071$ (most attenuating) to $-1024$ (least attenuating). These values are obtained after a linear transformation of the attenuation coefficients obtained while scanning and are expressed on the *Hounsfield scale* [60].

By stacking a series of 2D CT slices and factoring in the thickness of each slice, a discretized patient representation is built by defining a 3D volumetric grid[2] (see figure 2.4b). Each element of the grid is called a *voxel* and can be thought of as a 3D generalization of an image pixel (see figure 2.3). Each voxel

[2]To be more precise, the original data source is natively a 3D representation of the per-pixel attenuation of the imaged volume. But it is always stored as a series of 2D slices, which is what every medical visualization tool uses, and this work as well. The author preferred the reported formulation in order to ease the interpretation.




*(a) Open CT machine [61]*


*(b) CT machine by SIEMENS [62]*

*Figure 2.2: (a) Internals of a CT machine, called a* CT gantry. *The letter "T" points to the X-ray generator, while the detector is with the letter "D". The letter "R" with the adjiacent arrow indicates that the apparatus rotates clockwise with respect to the observer. (b) A CT gantry with a patient and a technician;*

represents an element of volume inside the patient body in which an estimated amount of radiation was absorbed (see figure 2.4b).

An important technical aspect to consider is the spatial definition of a voxel, i.e. what are its actual dimensions. In most of the CT scans used for RT, the distance between two subsequent CT slices is much higher than the size of a pixel on the slice itself. As a result, the actual CT voxels have a box shape rather than a cubic shape (much higher than wider). On transversal slices typical pixel dimensions are around $1.25mm \times 1.25mm$, while the distance between two slices[3] can be up to $3mm$.[4]

Moreover, the field of view may vary between patients, depending on the goal of the diagnostic procedure. As a result, the number of transverse CT slices differs from patient to patient, as well as the final CT 3D.

[3] i.e. the height of the voxel.

[4] The exact pixel dimensions in practice depend on a series of factors. The technician usually sets only the number of pixels on the transverse slice, which is usually $512 \times 512$ pixels. But the limiting factors defining the vertical resolution are time, storage and the field of view. Using modern machines and small fields of view, the voxels dimension can reach $0.5 \times 0.5 \times 0.5mm^3$.

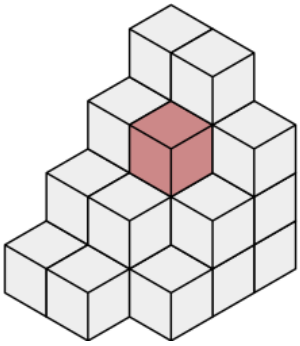
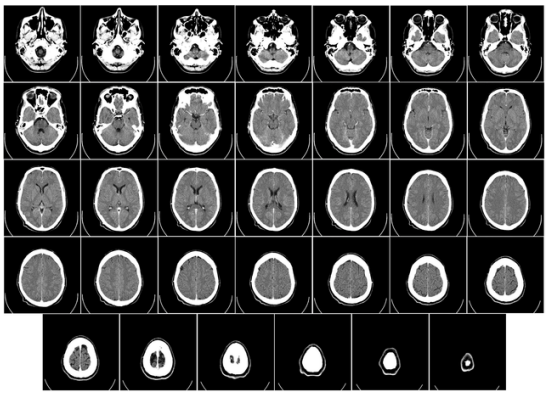
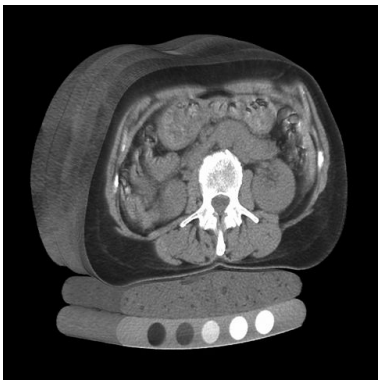


*Figure 2.3: A series of voxels in a stack with a single voxel shaded [63].*



*(a) CT slices of a human brain [64].*



*(b) Volume created by stacking CT slices [65].*

*Figure 2.4: (a) Series of subsequent CT slices of a human brain, from the lower part to the upper part. (b) Volume created by stacking subsequent CT slices in the abdominal region.*

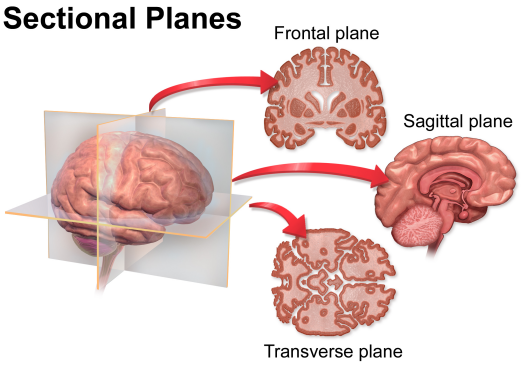
### Anatomic Planes and Coordinates Systems

During diagnostic and visualization procedures, the 3D representation of the patient is commonly sliced in three perpendicular planes called *anatomical planes* (see figure 2.5):

- *Axial* or *transverse* plane: an horizontal plane that divides the patient's body into superior and inferior parts (considering the patient as standing). It is perpendicular to the axis of rotation used to perform the scan. The slices obtained on this plane coincide with the original CT slices generated by the tomographic process.[5]

[5]For this reason, transverse slices are considered to contain the most information and are used as a base for diagnosis.

- *Coronal* or *Frontal* plane: a vertical plane that divides the patient's body into ventral and dorsal sections.[6]
- *Sagittal* or *Longitudinal* plane: a vertical plane that divides the patient's body into right and left parts, always defined from the point of view of the patient.




*(a) Depiction of the anatomical planes of a human brain [66]*

*(b) Sagittal (top left), frontal (top right) and transverse (bottom) views of the abdominal region [65].*

*Figure 2.5*

At this point we can introduce the spatial coordinate system that is used in this work. We will use a Cartesian reference system composed of three orthonormal vectors: $\{\bar{x}, \bar{y}, \bar{z}\}$. Each of these vectors is perpendicular respectively to the frontal, sagittal, and transverse plane. A depiction of the coordinate system in respect to a patient body can be found in figure 2.6.

### 2.1.2 RT Patient Modeling

The 3D model obtained by stacking CT slices is a great tool for diagnostic purposes, but there is no explicit information on where the tumor and the sensible tissues are. It is just a 3D grid of scalar values which are proportional to

[6] That is, dividing the patient's body into the belly and back sections.

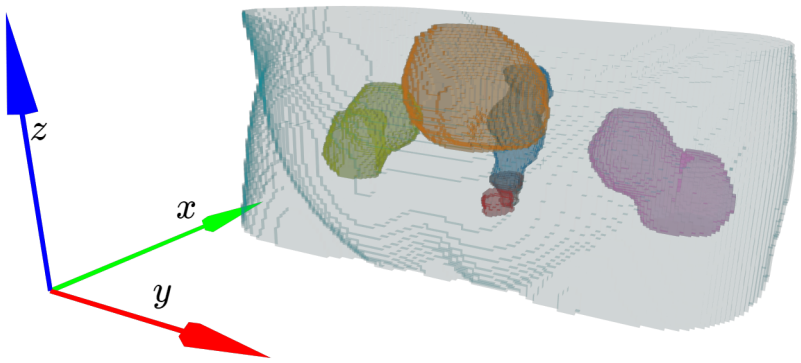



*Figure 2.6: Voxel geometric representation of OARs and external ROI of pelvic disease site. Left femur in violet, right femur in green, bladder in orange, rectum in blue, pelvic bulbus in red, anal canal in brown, external ROI in light blue. Created with [67]*

.

the amount of X-ray radiation absorbed and scattered by each voxel. To plan and proceed with the treatment it is necessary to identify at which *Region of Interest* (ROI) each voxel belongs to.

Depending on the part of the body contained in the region and their role in the treatment planning, ROIs can be categorized in different types:

**Target Volumes:** represent a series of encapsulated volumes that will receive a maximal dose of radiation, for this reason they are also known as *targets*. The innermost one is the *Gross Tumor Volume* (GTV). It is then extended with a margin, in order to treat microscopic tumor extension, forming the *Clinical Target Volume* (CTV). The CTV is further expanded by an anisotropic margin to accommodate uncertainties deriving from setup variation. This last and outermost volume is called the *Planning Target Volume* (PTV) [31]. A graphical representation of all the target volumes is reported in figure 2.7.

**Organs At Risk (OARs):** all the organs that are at risk of receiving a radiation dose. The goal of treatment planning is to minimize the radiation received by OARs while maximizing it for PTVs.

**Markers:** also called *fiducial markers* are small metal objects that are surgically placed in or near a tumor in preparation for RT. Their goal is

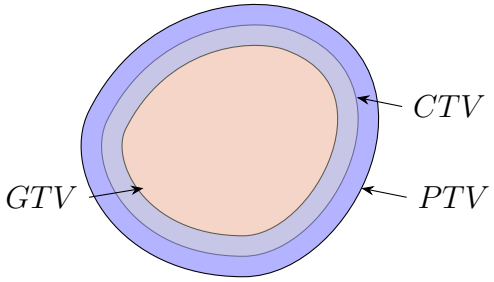



*Figure 2.7: Schematic of the different volumes irradiated in RT. Image reproduced with the permission of the author (Cecilia Battinelli)*

to identify the tumor's location with greater accuracy and help deliver the maximum radiation dose to the tumor [68].

**Avoidance:** generic volume in which the radiation dosage should be minimized. They are frequently composed of so-called *Algebraic ROIs*, which are ROIs generated when the PTV partially includes one OAR. In this case, it will be incorrect to minimize the dosage on the totality of the organ, hence an ROI is calculated by subtracting the PTV from the organ (see figure 2.9). Other types of avoidance ROIs are the *No Treatment* (NT) ROIs, which are an isometric margin around the PTV used to guide the optimization process to avoid irradiation to all the tissue around the PTV, regardless if it is an organ or not. NT can be represented as a hollow container encapsulating the PTV.

**Helper ROIs:** all the ROIs that are useful to tune the treatment planning, register the image or position the patient.

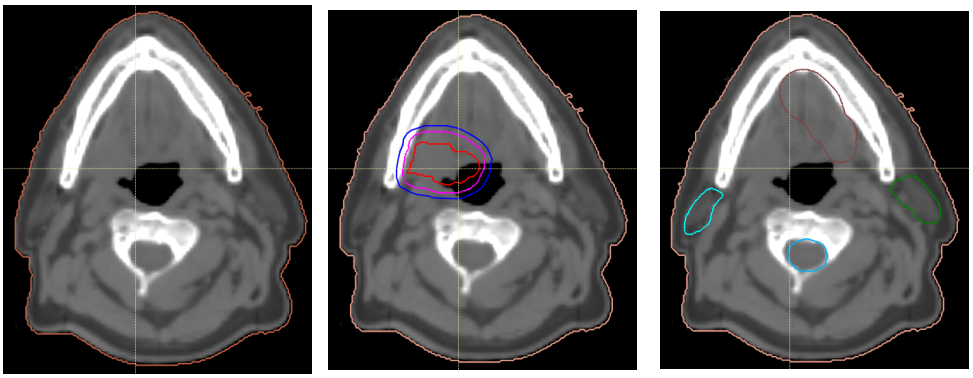
**External:** this ROI represents the body of the patient and should enclose all other ROIs.

Note that each voxel, or group of voxels, can belong to multiple ROIs. For example, a voxel in the GTV is included both in the CTV and PTV (see figure 2.7).

From an operative perspective, depending on the institution and the regulation, ROIs may be delineated by different medical experts (usually oncologists) and RT technicians. The process may be manual, automatic or semi-automatic[7] (as reported in section 1.1), but the end result is always encoded

[7]The medical expert will always review, verify and sign the result.

as an overlay of the CT transverse slice (see figure 2.8). For this reason, the operation of segmenting the patient geometry may also be called *contouring*, as *contours* are drawn to delineate the segmented ROI. In figure 2.6 we can see the final geometric representation of a pelvic disease site obtained by stacking all the transverse contours of the OARs.



*(a) CT image* *(b) Targets* *(c) OAR*

*Figure 2.8: (a) 2D CT image and external ROI in pink; (b) contours of targets: GTV in red, CTV in purple, PTV in blue; (c) contours of OARs: spinal cord in light blue, left parotid gland in turquoise, right parotid gland in green, oral cavity in brown; Patient 0522c0766 from [40].*

It must be noted that the above list covers only the ROIs types that are relevant for this work, the complete list specified by the DICOM standard (see section 2.1.3) can be found in [69].

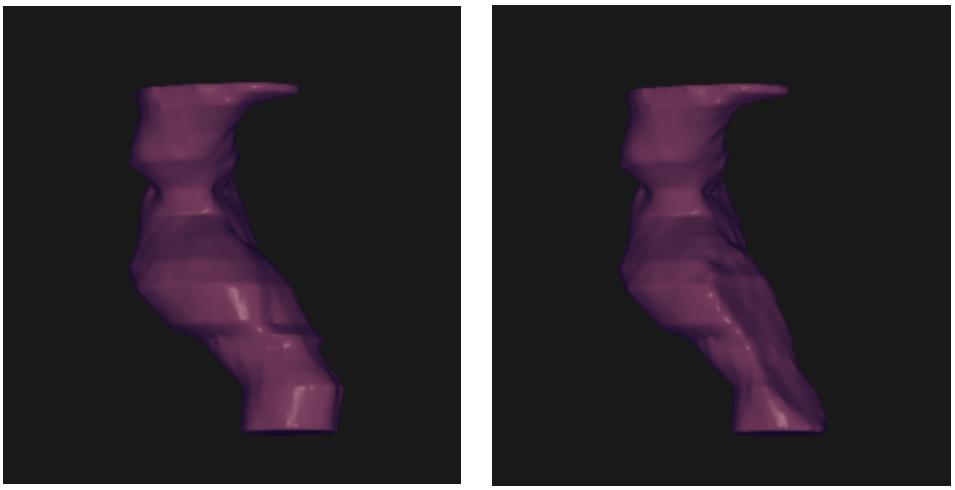
### 2.1.3 The DICOM Standard

The DICOM standard (*Digital Imaging Communication in Medicine*) is the *de facto* standard for transmission, storage and retrieval of digital images in the medical field [70].

The central components of the DICOM data structures are called *Information Object Definitions* (IODs). They may be considered as a well-defined schema of attributes associated with each object. Examples of objects are: CT images, RT plan specifics, voice audio recordings, PDF documents and many more.[8]

Every attribute inside this schema has a unique identifier composed of two hexadecimal numbers: the group number and the item number. For example,

[8] For a complete description of the whole DICOM standard please see `https://dicom.innolitics.com/ciods`



*(a) Rectum OAR.* *(b) Rectum OAR minus PTV*

*Figure 2.9: (a) A normal contoured rectum; (b) An avoidance ROI obtained by subtracting the PTV from the OAR in figure (a);*

the ROI name attribute has code (3006, 0026). Every single image[9] encoded using DICOM has its respective IOD [71], which at filesystem level is stored as a file header.

ROIs are considered an attribute of the *RT structures set* IOD [72], which is called *RT ROI Observation*. For this reason, the terms "ROI" and "RT structure" (or just "structure") are commonly used interchangeably.

We do not enter into the details of the DICOM standard specifications, but it is important to understand how ROIs are encoded and represented in DICOM.

Each ROI defined in a transverse CT slice contains also a series of attributes, the most important for this work being:

- `ROIName` (3006,0026): user-defined name for the ROI.
- `RTROIInterpretedType` (3006, 00A4): type of ROI (PTV, CTV, GTV, AVOIDANCE, ORGAN, etc.)
- `ROIContourSequence` (3006,0039): series of points on the CT transverse slice representing an open or closed polygon.

As clearly stated in the DICOM specifications, there is no enforcement or check on the content of the `ROIName` field, as it is a free text field completely in the hands of the user. The field `RTROIInterpretedType` may

[9] A single transverse CT slice

be optionally used to define the type of ROI, as explained in section 2.1.2. The `ROIContourSequence` contains the actual polygon that defines the ROI on the transverse CT slice. By stacking the polygons on all the slices and interpolating them, the 3D *"voxelized"* representation in figure 2.6 is obtained. Note that in DICOM it is totally possible for an ROI to not define any contour, resulting in an empty ROI (i.e. without any associated voxel). It is also possible to have single-point ROI (actually called *Points of Interest* (POI)). Common uses of POI are for patient and image registration as well as identification of fiducial markers.

## 2.2 Naming Standards

The current lack of standardization in ROI naming should not be associated to the actual inexistence of standards. In fact, multiple standards have been proposed and the general attention of the clinical world to the need of naming standardization is increasing. In this section we discuss the standards and ontologies used in medical practice and RT.

### 2.2.1 Ontologies

Ontologies offer a rich framework for defining concepts and inter-relationships among them. Ontologies have been extensively used in the medical domain and represent an important component in interoperability and integration into health care informatics systems. The BioPortal [73] website maintained by the National Center for Biomedical Ontology (based in Stanford, California) contains a wide variety of medical ontologies that are publicly accessible.

#### Foundational Model of Anatomy

The *Foundational Model of Anatomy* (FMA) [74] defines anatomic structures and interrelationships necessary for a phenotypic representation of the human body. The intent of the FMA is to accommodate all current naming conventions, rather than attempting to standardize terminology. The FMA is often used by other ontologies and other naming standards as an important reference to define concepts with a very high degree of precision. For its inherent complexity, the FMA is seldom used in daily medical practice. This is particularly the case for radiation oncology, where the DICOM standard does not provide for any explicit reference to the FMA.

#### SNOMED CT

The *Systematized Nomenclature of Medicine–Clinical Terms* (SNOMED CT) is a standardized terminology owned and licensed by the International Health Terminology Standards Development Organization (based in London UK). According to the authors, it is *"the most comprehensive, multilingual clinical healthcare terminology in the world"* [75], actively used in eight countries. It is particularly aimed to store and organize electronic health records in the wide sector of health care. As such, it lacks the simplicity and pragmatic aspects required to be proficiently used in daily RT practice. Moreover, the labels of the concepts in SNOMED CT contains special characters that are not supported by all the vendors providing solutions to the RT field [41].

### 2.2.2 AAPM TG-263

The *American Association of Physicists in Medicine* (AAPM) is an established organization that focuses on advancing patient care by providing education, improving safety and efficacy of radiation oncology and medical imaging procedures through research. At the beginning of 2018, AAPM released the final report of its task group number 263 (TG-263) [48] having as a sole goal the identification of a comprehensive nomenclature standard for RT that could be easily and proficiently used in every medical institution in the United States.

After reviewing the ontologies reported above and the recent development in standards for nomenclature in RT [46, 76, 77, 78], the task group developed a comprehensive nomenclature system of all the concepts used in RT. Special attention was posed to practical limitations (like characters supported by vendors' solutions) and to the utilization of names that minimizes the chance of communication errors. As a result, TG-263 names are short but easy to understand and interpret, even without a strong background in anatomy. Important concepts of RT that were not reported in medical ontologies (like algebraic ROIs and target structures) are covered in great detail. TG-263 is not an ontology and does not aspire to be one. It can be considered as a set of simple naming guidelines and conventions. On the other hand, when possible (i.e. for OARs) the FMA identifier that most closely match the represented ROI is provided, thus enabling direct linking with the FMA structure. This latter aspect is not to be underestimated; a properly standardized TG-263 clinical data set isn't just more usable for medical purposes, but it also allows for the use of semantic web technologies thanks to the integration with the FMA ontology.

The list of standardized OAR names is publicly accessible and constantly updated [79]. Given its straightforward architecture and its growing adoption,

we decided to use TG-263 in the course of this work to label the OARs in our data set (see section 3.3).

## 2.3 Shape Descriptor and Normalized Central Moments

Shape descriptors are a class of features used in computer vision that are based on the shape of an object rather than on other, maybe richer, sources of information (like the intensity of color in an image). They are an important tool used in content based image retrieval, image search, and image classification.

Following the taxonomy outlined in [80], shape descriptors can be divided into two main categories: *contour-based* and *region-based* descriptors. As the name may suggest, contour-based descriptors extract features only from the contour of the shape (i.e its border). Instead, in region-based descriptors, the features are extracted from the whole region occupied by the object.

As already stated in section 1.8, given that the contouring protocol used by the medical expert in the proximity of the organ surface can vary from institution to institution, in this work we preferred to use region-based shape descriptors[10].

In the case of 3D shapes, simple shapes descriptors are the *volume* intended as the number of voxels contained in the shape, the *surface* as the number of voxels on the surface of the shape, the *circularity* as the square of the surface over the volume and the *major axis orientation*.

A particular family of region-based shape descriptors is composed of *moments invariants*. The first and most simple moment invariants are the *geometric moment invariants* [82]. Their definition for the 2D case is

$$\begin{aligned} M_{pq} &= \sum_x \sum_y x^p y^q f(x,y) \\ p, q &= 0, 1, 2, ..., \end{aligned} \tag{2.1}$$

where $x$ and $y$ are the coordinates of the pixel in the image, $f(x, y)$ is the intensity of the pixel with coordinates $(x, y)$, and $p$ and $q$ are the parameters dictating the order of the moment. Geometric moments are not translation invariant in this simple formulation, because changing the position of the object offsets the $x$ and $y$ coordinates. To render them translation invariant and construct the *central moments*, it is sufficient to subtract to $x$ and $y$ the coordinates

[10] For an in-depth review of contour-based shape descriptors we suggest the reading of [80] and the excellent survey by Zhang and Lu [81].

of the centroid of the shape:

$$\begin{aligned}\mu_{pq} &= \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y), \\ \bar{x} &= \frac{M_{10}}{M_{11}} = \frac{\sum_x \sum_y x f(x,y)}{\sum_x \sum_y f(x,y)}, \\ \bar{y} &= \frac{M_{01}}{M_{11}} = \frac{\sum_x \sum_y y f(x,y)}{\sum_x \sum_y f(x,y)}.\end{aligned} \tag{2.2}$$

The centroid is nothing more than the 2D discrete form of the center of mass of an object, a concept borrowed from physics:

$$\begin{aligned}\bar{R} &= \frac{1}{M} \iiint_V \bar{r}\rho(\bar{r})d\bar{r} = \frac{1}{M} \begin{pmatrix} \iiint_V x\rho(\bar{r})d\bar{r} \\ \iiint_V y\rho(\bar{r})d\bar{r} \\ \iiint_V z\rho(\bar{r})d\bar{r} \end{pmatrix}, \\ M &= \iiint_V \rho(\bar{r})d\bar{V}.\end{aligned} \tag{2.3}$$

Where $\bar{r}$ is the coordinate vector used to integrate in the whole space $V$, $\bar{dr}$ is the infinitesimal volume in position $\bar{r}$, $\rho(\bar{r})$ is the density of the object in $\bar{r}$, and $M$ is the total mass of the object (not to be confounded with the central moment of the shape).

As shown by Hu [82], translation and scale invariance can be achieved by further diving the central moments by a properly scaled zero-th central moment:

$$\begin{aligned}\nu_{ij} &= \frac{\mu_{ij}}{\mu_{00}^{1+\frac{i+j}{2}}}, \\ i+j &\geq 2.\end{aligned} \tag{2.4}$$

Thus constructing the *Normalized Central Moments* (NCMs), which will be extensively used in this work (see section 3.4.2).

An evolution of geometric moments are the *orthogonal moments*, where the kernel $x^p y^q$ is replaced by a generic kernel $P_p(x)P_q(y)$, where $P$ can be a Legendre or a Zernike polynomial [83].[11] Orthogonal moments require to inscribe the shape into a unity sphere in order to be transition and scale invariant, which implies a form of up/down interpolation of the object.

Other types of region-based shape descriptors are the *generic Fourier shape descriptors* [84], in which a 2D Fourier transform is applied on the polar-raster

[11] The name "orthogonal" comes from the fact that the Zernike and Lagrange polynomials are orthogonal.

of the surface of the object. The Fourier coefficients are then used as feature vector.

In this work, we opted to employ only NCMs as they perform well on contour based shapes without interior content [80], their implementation is supported by major open source libraries and they are sensibly less computational expensive than orthogonal moments, where the calculation of Zernike or Lagrange polynomial is not trivial. Following the principles exposed in 1.5, to avoid the re-implementation of the NCMs in the 3D form, we preferred to use already existing tools on the projections of the OAR on each anatomic axis. The projection was calculated by summing the voxels along each of the directions in figure 2.6, therefore obtaining one projected 2D image per anatomic plane.

More formally, if $I(x, y, z)$ is the binary intensity of the 3D OAR (i.e. the value of the voxel in position $x, y, z$), then for each direction we calculate the 2D projections

$$\begin{aligned} f(x', y')_x &= \sum_x I(x, x', y'), \\ f(x', y')_y &= \sum_y I(x', y, y'), \\ f(x', y')_z &= \sum_z I(x', y', z), \end{aligned} \tag{2.5}$$

where $(x', y')$ are the coordinates on the 2D projection, and $f(x', y')_x$ is the intensity of the 2D projection along direction $x$ of the 3D OAR. For each projection we calculate a series of NCMs up to a defined order, constructing a feature vector of scalar values that is used to describe the shape of the OAR.

## 2.4 Machine Learning Algorithms

In this section, we explain the concept of *supervised learning*, the learning framework used in this work, and we report a brief explanation of the ML algorithms employed.

### 2.4.1 Supervised Learning and Classification

As the name might suggest, in the framework of supervised statistical learning a strong supervising signal is required to train a model to produce predictions. This supervised signal is composed of a set of $K$ examples $\mathbf{D} = \{\bar{x}_k, y_k\}_{k=1}^{K}$,

also known as the *ground truth*, where $\bar{x}_k \in \mathbb{R}^N$ is a single data *instance* or *predictor* and $y_k$ is the *response variable*. The goal of the ML algorithm is to learn the entailing function $f : \bar{x} \rightarrow y$ from the supervising signal, in order to predict the response variable of an unknown data instance. The response variable can take various forms, it could be a categorical variable $y_k \in \{1, ..., C\}$, or it could be a real value $y_k \in \mathbb{R}$. In the former case, the task of learning is called *classification*, in the latter is called *regression* [85]. In this work we are interested only in supervised classification, more precisely in *mono-label* classification, meaning that the response variable is composed of one and only one value, while in *multi-label* classification the response variable can have multiple coexisting values.

### 2.4.2 Training Process and Classification Metrics

In order to train an ML model and evaluate its performance, the ground truth is divided into three sets:

**Train:** it contains the instances and the labels used to train the model.

**Validation:** also known as *"development"* or *"evaluation"* set, it is used to calculate the performance of the trained model and perform parameter tuning.

**Test:** also known as *"held out"* set, it is composed of instances that neither the model nor the practitioner has ever used. It is used to evaluate the performance of the model in a real-world scenario.

The performance of the model is evaluated by comparing the model prediction $\hat{y}_k$ with the corresponding response variable in the ground truth $y_k$. In order to ease the explanation, we are going to use as an example a binary classification, meaning that the response variable can only have two values $y_k \in \{0, 1\}$, respectively called the *negative* and *positive* value; for example an ML algorithm able to tell if a patient is affected (positive) or not (negative) by a disease. The single model prediction can then be categorized in the following outcomes:

**True Positive (TP)**: meaning that the model predicted a positive and the response variable in the ground truth is positive.

**True Negative (TN)**: meaning that the model predicted a negative and the response variable in the ground truth is negative.

**False Positive (FP)**: meaning that the model predicted a positive, but the response variable in the ground truth is negative.

**False Negative (FN)**: meaning that the model predicted a negative, but the response variable in the ground truth is positive.

In case of TP and TN we have that $\hat{y}_k = y_k$, while for FP and FN we have that $\hat{y}_k \neq y_k$. When considering the full set of predictions of the model, the number of TP, TN, FP, and FN can be summarized in a *confusion matrix*

*Table 2.1: Confusion matrix of a binary classifier.*

| | | **Predicted Value** | | |
|---|---|---|---|---|
| | | **p** | **n** | **total** |
| **True value** | **p′** | True Positives | False Negatives | P′ |
| | **n′** | False Positives | True Negatives | N′ |
| | **total** | P | N | |

If the classification is perfect, all the examples will be on the diagonal of the confusion matrix.

Based on the confusion matrix, a series of scalar metrics comprised in the range $[0, 1]$ can be built. *Precision* is defined as $Prec = \frac{TP}{TP+FP}$, and represents (in a frequentist fashion) how much the model is correct when predicting that the instance is a positive. *Recall* is defined as $Rec = \frac{TP}{TP+FN}$, and represents how many of the total number of positive cases in the data set have been actually retrieved.

Following the example of the diagnosis of a disease, having an high precision and low recall means that if the model predicts a positive than the chances of the patient to be ill are high and actions should be taken; while the low recall implies that many of the actually ill patients have not been identified as such.

To better represent the trade-off between precision and recall in a single

scalar value, the *f1 score* is commonly used

$$f_1 = \frac{2}{\frac{1}{Prec} + \frac{1}{Rec}} = 2\frac{Prec \cdot Rec}{Prec + Rec}, \tag{2.6}$$

which is the harmonic mean of the two values and it is also bound in the interval $[0, 1]$.

Another global metric commonly used is the *accuracy*

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}. \tag{2.7}$$

Accuracy is valuable only in case the number of examples per class is balanced. If the condition is not met, accuracy is bound to be biased towards the more frequent class. Again following the diagnosis example, if only $1\%$ of patients is actually ill, a classifier that always predicts the patient to be fine has a $0.99$ accuracy, which is indeed a misleading number. On the other hand, the precision is undefined and the recall is $0.00$, signaling that the classifier is not functioning properly.

The extension of the confusion matrix to *multi-class* cases (i.e. not binary), is immediate. The true and predicted labels have $C$ possible values, and the matrix is of size $C \times C$, like the ones in figure 4.1. The precision and recall metrics are then calculated by treating each class as a one-vs-rest binary problem. Then, the per-class metrics are combined with a weighted average, taking into account the frequency of each class in the data set[12].

### 2.4.3 Decision Trees

*Decision Tree* (DT) classifiers are among the most simple ML classification algorithms.[13] They are structured as a tree-like graph, where each node corresponds to a particular condition applied to one of the features of the instance. The effect of the node is to split the data set,[14] with the end goal of producing leaves that contain only one class of examples, a situation named *pure leaf*. The conditions in each node are selected during the training phase based on how effective they are in splitting the training data set. This effectiveness is measured either in terms of *information gain* or of *Gini impurity*. The information gain is calculated as the difference between the entropy of current tree and the weighted sum of the entropy of the candidate child condition. The

[12]The explicit formulas for the problem at hand are reported in section 3.5.

[13]They can be adapted to regression tasks.

[14]For the scope of this work, each node can only perform a binary split

Gini impurity is calculated as the sum of the probability of an instance with a specified target label to be picked, multiplied by the probability of an incorrect classification of that instance.

### 2.4.4 Random Forest

Decision trees can be combined together forming a well renowned ensemble model called *Random Forest* (RF) [86]. In RF, the trees are called *estimators* and they are independent of one another. Each tree is trained from a set of randomly sampled training examples, a procedure called *bagging*. Moreover, the candidate features for the split in each node are randomly sampled from the $N$ total features. Compared to a single DT, the feature sampling lowers the correlation between the trees and avoid that only the strongest features in terms of splitting criterion are always picked, forcing the whole model to make a more extensive use of the feature set. Once a set of $T$ estimators $t_1, t_2, ..., t_T$ is trained, the prediction on new instances is performed by taking the majority vote of all the estimators. The bagging mechanism is instrumental in achieving better predictions, as it reduces the variance without increasing the bias of the model. The set of parameters used for tuning are the number of estimators, the maximum number of features sampled in each split, the minimum number of samples a node can have to be considered a leaf, and the maximum depth of each tree.[15]

### 2.4.5 3D Convolutional Neural Networks

*3D Convolutional Neural Networks* (3DCNN) are a particular kind of convolutional neural network that operates on 3D voxelized volumes rather than on 2D images or on 1D signals. For this reason, the kernels of a 3DCNN are 3D tensors that slides through the input volume on the $x, y, z$ axis. To each direction the practitioner can associate a different stride. Besides this aspect, all the mechanics of the network and of the training process are exactly the same as in 2D and 1D CNNs.

A thorough explanation of neural networks models is outside the scope of this work. We suggest the read of [87] for a complete overview of the topic. For a more succinct read, the initial sections of [19] and [88] are an excellent starting point, especially for readers having a medical background.

In this work we selected a particular 3DCNN called *VOXNet* [89], initially developed for point-cloud object classification, but still capable to achieve ex-

[15] See appendix A.6 for an in-depth report of the tuning process.

tremely good classifications results also on voxel-based data sets like the modelnet40 [90], this despite its rather simple architecture of just two convolutional layers:

**input**: $[32 \times 32 \times 32 \times 1]$ voxelized occupation grid with only one channel.

**Conv1**: 32 filters, kernels of size $[5 \times 5 \times 5]$, strides of $(2, 2, 2)$ for the $(x, y, z)$ directions, ReLU activations. The resulting input for the next layer is $[14 \times 14 \times 14]$.

**Conv2**: 32 filters, kernels of size $[3 \times 3 \times 3]$, strides of $(1, 1, 1)$, ReLU activations.

**Max Pooling**: max pooling with a kernel of size $[2 \times 2 \times 2]$, the resulting input size after Conv2 and max pooling is $[6 \times 6 \times 6]$ that is then flattened in a feature vector of 128 scalar values.

**Fully connected**: fully connected layer of K units, where K is the umber of classes, using as input the 128 features above. Relu activations.

The justification of this choice can be found in section 3.4.3.

## 2.5 Machine Learning in Medical Imaging

In this section we briefly report the main areas of medical imaging in which ML algorithms have been used to solve medical problems. Following the classification outlined in [19], we can categorize these areas as follows:

**Feature Representation**: features are either automatically learned or manually engineered to describe raw medical images. These features are then used for image clustering or content based medical image retrieval (CBMIR) to ease the search for similar images.

**Computer Aided Detection (CADe)**: CADe aims to find and/or localize suspicious and abnormal regions, with the intent of automatically alerting clinicians. The goal is to increase the detection rate of disease regions with a particular focus on reducing false-negatives that can be caused by excessive fatigue of the clinician or by human error.

**Computer Aided Diagnosis (CADx)**: in CADx, an automatic system provides an opinion over the nature of a possible disease in order to help

the human expert. The typical application involves the discrimination between malignant or benign lesions.

**Segmentation**: in segmentation, area of the patient body corresponding to a well-defined entity are identified and contoured. Segmentation of medical images is not only applied to MOS (as explained in chapter 1), but also to brain imaging, neonatal imaging, and histopathology.

**Clinical Outcome Prediction**: the goal is to predict if a patient will or will not develop (or re-develop) a disease after a certain period of time from when the imaging procedure was performed. Typical examples of application in this area are to predict the risk of a patient to be hospitalized again after a treatment. For its nature, clinical outcome prediction requires models to be interpretable, and, when possible, allow for casual inference. This is to identify the causes of risk and apply countermeasures to decrease the probability of re-hospitalization.

**Anatomical Structures Detection and Classification**: in this area ML algorithms are used to detect important anatomic regions (organs, body parts, or even cells). After the detection process, which usually consists in identifying promising bounding boxes in the image, a classification task is performed to identify which kind of anatomic region is represented.

Covering all these areas in details is far from the scope of this work, but we point the reader to [19, 88, 16, 91, 92] for a series of well-curated surveys on the use of ML in medical imaging.

We now focus on a series of works in the area of anatomical structure detection and classification that are of particular interest for this work.

Roth et al. [54] used a Deep Learning (DL) fully supervised approach to classify and map transverse CT slices to one of five body parts they belong (neck, lung, liver, pelvis, and legs). The architecture used was a CNN with five convolutional layers for feature extraction and a dual layer 4096 units fully connected part for the classification task. The training set contained roughly 4K slices, obtaining a classification accuracy of $94.1\%$. However, for this strategy to be effective in real-world scenarios, a higher grade of discrimination than just five body parts may be needed.

This limitation is overcome by Yan et al. [93] by using a CNN trained in a multi-stage fashion (pre-training and boosting) with multi-label classification objective (given that each CT slice may contain multiple organs). The final stage was based on the classification of sliding windows on the CT scans. The

network showed peaky response only to the windows containing organs, that were then used for classification purposes. The data set used contained 12 organs; 2413 slices (225 patients) were used for training, 656 slices (56 patients) for validation, and 4043 slices (394 patients) for testing.

In [26], Chow et al. investigated the learning curve associated to the same procedure used in [54], suggesting that the high quality and low variability of medical images do not require excessive amounts of data to obtain good classification results, provided that peculiar and anomalous cases are excluded from the training.

In [94], Yan et al. demonstrated the validity of mining CT scan annotation contained in hospitals PACS to create a large scale data set (32K lesions from 10K studies) for lesion detection. Although their approach was simply based on a specific kind of annotation used only for lesion identification, they were able to generate such a data set with minimal manual work. They then demonstrated it's validity by successfully training a lesion detection (CADe) DL model.

All the works reported above suggests that the classification of organs from CT slices through an ML approach is a viable solution. Moreover, for the classification task, huge data sets may not be required, while they can indeed be extracted for other tasks from already existing clinical PACS.

Finally, the work that most closely resembles our problem is the one from Rozario et al. [95], in which a CNN is used to classify OARs on 2D slices and then assign them a unique TG-263 label. Instead of working on raw CT images (like all the works above), Rozario used only the OAR contours annotated by the medical experts. The approach followed was to represent in each data instance both the OAR to classify and its spatial context, which was assumed to be coherent in every patient. Instead of using the full OAR geometry (like in figure 3.2), each data instance was derived from a 2D transverse slice. More precisely, the whole 3D patient's representation containing all the OARs (see section 2.1.2 and figure 3.4), was sliced on the horizontal plane and each transverse slice was treated separately. Then, for each OAR present in the slice, a 2D data instance was created with the following encoding: the pixels[16] pertaining to the OAR to classify were encoded with a value of $1$; the pixels belonging to the other OARs in the slice had a value of $0.5$; while all other pixels had a value of $0$. The end result was a 2D matrix of float values that can be plotted as an image. The mechanics of this method imply that if a slice contained $n$ OARs, then $n$ instances would be created. In fact, by taking into

[16]It would be more correct to refer to them as voxels, but the author prefers the term pixel to ease the explanation, given that they are located on a 2D transverse slice.

account the number of horizontal slices per patient, Rozario generated $40K$ instances from a data set containing only $100$ patients. The results sensationally reported on a prostate and head-and-neck data sets were of $100\%$ accuracy. However, a series of aspects were not treated by this study. First of all, the whole data set was manually cleaned before training the model, thus removing the automation goal from the scope of the work. Then, no baseline method was attempted or reported to establish a measure of comparison. Moreover, the OARs in the data set did not show any sign of overlapping, a condition that is not met in our data set. No consideration on the confidence of the prediction was performed, either silently implying that the data set did not contain any peculiar case or that the inference of the model should be trusted blindly. This is not the case for our data set, were, for example, prosthetic femurs may be present. We would like to take this eventuality under consideration. Finally, the study suggested as a future line of investigation to use the whole 3D representation rather than just the 2D slices, an aspect on which we would like to focus.

We can say that the main contribution of the work by Rozario et al. is that it proves that using an ML approach to classify OAR contours is possible and that in this effort the encoding of the spatial context at which each OAR belongs is likely to be important.

# Chapter 3

# Method

Our goal was to implement an automatic standardization approach of OAR labels based on their geometries by exploiting their spatial consistency. The underlying hypothesis was that it would be possible to do achieve this result through the application of an ML model. As such, we formulated the problem outlined in section 1.1 as a multi-class mono-label classification task [85], where a unique OAR label is associated to a generic ROI geometry encoded as a 3D binary tensor (like in figure 3.2). Then, a fully supervised approach is followed to train an ML classifier. This approach was chosen as being the most established and successful methodology in object classification tasks [85, 11].

The data set at hand was a real-world clinical database of prostate cancer cases containing annotated ROIs geometries. The raw format of the data is reported in section 3.1, and we advise the reader to familiarize with it early on. The train, evaluation and testing set were constructed in a semi-automatic fashion by following a divide-and-conquer-like approach. First, only OARs were selected, and second, patients containing the highest number of OAR were used in the train and evaluation set, while less representative patients and potentially more peculiar cases were confined in the test set. This with the goal of simulating the application of the final model to a real-world clinical database, which may contain complex cases. Through three iterations, we were able to use different data encoding upon which various ML models were trained and their performance was evaluated with standard classification metrics. In order to answer our second research question, that the inference probability could be used as a measure of confidence, we implemented a reject class mechanism and evaluated the thread-off between coverage of the solution and its final accuracy.

The goal of this section is to ease the reproduction of the results, by ex-

plaining in great detail the process followed, which can be divided into the following steps:

1. **Raw data set:** (section 3.2) the data set at hand is analyzed to underline the frequency of each ROI label, the number of patients and the number of ROI per patient.

2. **Creation of ground truth**: (section 3.3) based on the results of the previous step, three sets of patients are created: training, evaluation, and testing. For each patient only ROIs that are considered to belong to OAR are taken under consideration.

3. **Input encoding and model training:** (section 3.4) each ROI in the ground truth is encoded as a vector.[1] A classifier is selected and trained on the encoded ground truth.

4. **Classification evaluation:** (section 3.5) the weighted precision and recall, as well as the confusion matrix and accuracy are calculated on the inferences obtained on the evaluation set.

5. **Inference Probability Evaluation:** (section 3.6) the distribution of the inference probability obtained on the evaluation set is used to establish a probability threshold below which each inference is rejected. The test set is used to calculate the reject rate of OAR and, when possible, of all ROI in the data set. The presence of incorrect classifications above threshold is considered a particularly negative factor in the model performance.

As stated above, steps 3 to 5 were repeated in an iterative fashion and the insights obtained in each iteration were used to inductively select the more suitable strategy for the following one. In order to assure a fair comparison ground for all the developed solutions, only step 3 was changed in each iteration. Finally, a summary of the methodology in the form of a flux diagram can be found in figure 3.1.

[1] Being it binary or a feature vector, depending on the method used.

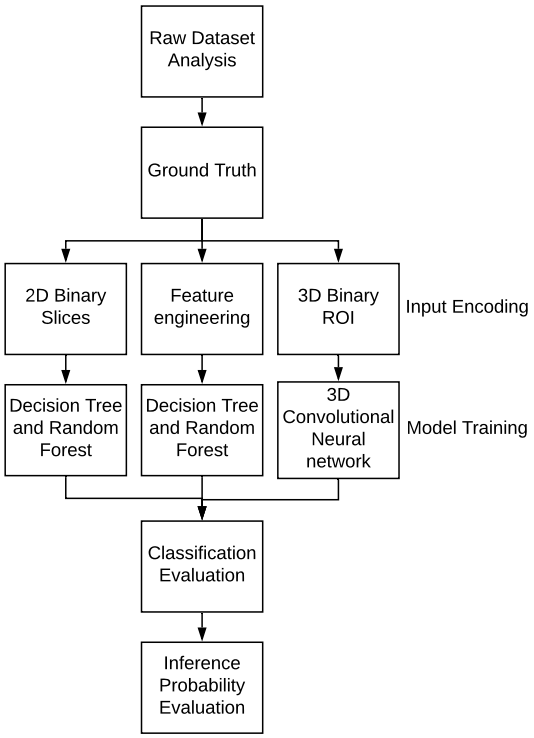



*Figure 3.1: Methodology diagram.*

## 3.1 Raw Data Encoding

In the totality of this work, the fundamental data instance is the ROI, which is composed of a set of three attributes:

| **Attribute** | ROI Name | Patient ID | Geometry |
|---|---|---|---|
| **Format** | alphanumeric string | alphanumeric string | voxels |
| **Example** | “RECTUM” | “anonymized_01” | 3D tensor binary mask |

The “ROI name” is a user defined text label associated to the ROI; examples could be: “RECTUM”, “BLADDER”, “marker”, “PTV”, “BLADDER-PTV”. The “Patient ID” is a unique identifier of the patient and it carries no information with regards to the personal identity.[2] The ROI geometry is encoded as a 3D tensor (or grid) covering the whole patient where each voxel in the tensor has value 0 if it is outside of the ROI or value 1 if it is contained in the ROI.

[2] All the data in this study are fully anonymous. Although, given the nature of the disease, only male patients are present.

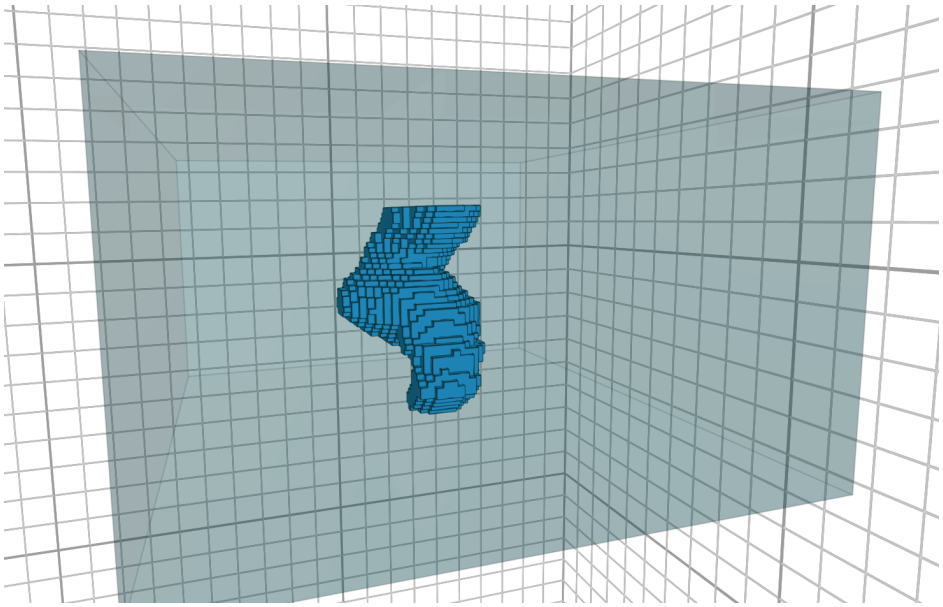


*Figure 3.2: 3D tensor representation of a rectum (OAR). The voxels in dark blue have value 1, while the voxels in shaded blue have value 0. The tensor covers the whole pelvic area of the patient.*

Effectively, this encoding can be thought to as a binary mask identifying the voxels contained in the ROI. Taking as an example the rectum OAR of figure 3.2, the dark blue voxels have value 1 and represent the voxels inside the rectum, while the light shaded voxels have value 0 because they are outside it, but still covering the whole patient body section.

"ROI name" and "Patient ID" derives directly for their DICOM equivalents reported in section 2.1.3, while the geometry is obtained by interpolating the DICOM polygon on each transverse slice and considering as belonging to the ROI only the voxels internal to the polygon.

The combination of "ROI name" and "Patient ID" is a unique identifier of the ROI as the data set contains only one full CT scan per patient.

As explained in section 2.1.2, the ROI geometry is based on the CT 3D representation of the patient, which has a well-defined spatial definition. In the specific case of the data set at hand, the original CT voxels have a dimension of $1.25 \times 1.25 \times 3.0$ millimeters respectively for the $x$, $y$, and $z$ direction[3]), resulting in a $512 \times 512$ pixels for each transverse CT slice.[4] In order to obtain

[3]See section 2.1.1

[4]The number of transversal slices is not the same for every patient, as such the number of voxels in the $z$ direction.

an isotropic representation (i.e. having voxels of the same spatial definition in all the three dimensions), the original CT volume is downsampled and interpolated to obtain cubic voxels of dimensions $2.5 \times 2.5 \times 2.5$ millimeters. The resulting number of voxels on the transverse plane is $256 \times 256$, while on the vertical axis it depends on how many 2D CT slices have been acquired and then stacked.[5]

In order to reduce the memory footprint of the data set, in case a top or bottom set of transverse slices does not contain any ROI other than the external one (i.e. the body), the slices are removed. This can be though as similar to removing leading or trailing spaces in a string.

## 3.2 Raw Data Set Analysis

The anonymized patient data set used for this study was received from Iridium Kankernetwerk (Belgium) with whom RaySearch has a data transfer agreement covering this project. It is composed of 181 male patients that have been diagnosed with prostate cancer. As such, all the ROIs included in the data set belongs to the *"pelvic"* region [48]. The ROI names are written in Dutch language.

The total number of ROIs is $3144$, with an average of $17.3 \pm 1.3$ ROIs per patient. The difference in the number of ROI is mainly caused by the number of avoidance structures, which depends on the size and location of the PTV dictated by the clinical condition of the patient. Also, the presence of different fiducial markers can influence the total number of ROIs.

Although the names are not in the English language, some ROI types can be inferred: target structures (*"CTV_high", "CTV_low", "GTV_high", "PTV_TOTAAL", "Dose 103", "Dose 104"*, ...); fiducial markers (*"marker", "markers", "merkers"*); avoidance structures ( *"rectum - PTV", "rectum Overlap", "rectum Post", "rectum in PTV", "rectum in ptv", "rectum overl"*, ...); the external ROI ("BODY"); the OARs ("rectum", "blass", "heup re", "heup li", ...).

More details about the raw data set composition can be found in appendix A.2.

[5] As explained in section 2.1.1, it means that the third dimension of the ROI tensor may vary sensibly from patient to patient. This characteristic is particularly important when the voxels are directly used as features for a classification task (typical in DL pipelines), because the input must always have the same dimensions.

## 3.3 Ground Truth

Following the established ML methodology reported in 2.4, the raw data set was partitioned in a training, evaluation and testing set. In addition to regular parameter tuning, the evaluation set was also used to establish the rejection threshold. While the test was used to simulate a real-world, uncleaned, clinical data set.

The building of the ground truth was performed in a semi-automatic fashion based on the results of the previous step. The rationale behind this choice is to limit human intervention as far as possible. Moreover, no extensive and time-consuming data-cleaning should be performed beforehand, otherwise the purpose of this study would be completely defeated.

Particular attention was given in taking into account what is the information known at inference time. For instance, we must consider that every ROI pertains to a specific patient and no one else. Hence it is very important that different input instances pertaining to the same patient must not be included in the train, validation and test data sets at the same time. For example, let's suppose that we are classifying OARs contained in a 2D transverse slice of a CT scan (as outlined in section 3.4.1). Given the resolution of the CT scan being of some millimeters on the $z$ dimension,[6] two adjacent slices coming from the same patient will appear to be very similar. If most of these slices are in the train set and some in the validation set, the model will have very little difficulty in correctly classifying the "unknown" instances, because they closely resemble the one used in training as they belong to the same individual. This effect is commonly referred to as a source of *"data leakage"* [96] and its consequences are that the obtained performance metrics will be sensibly over-optimistic compared to the the ones expected in real life applications.

For the reasons above, the train, validation and test set are composed of separate groups of patients that are mutually exclusive. This approach also grants the independence of the encoding strategy, as long as the obtained encoded input contains information pertaining to the same patient.

That said, to build the ground truth the following issues must be addressed:

- Which ROI names should be considered as belonging to OARs.
- Which patients should be included in each set.

[6] See section 2.1.1

### OAR Names identification

To address the first issue, one could simply argue that the DICOM attribute `RTROIInterpretedType` introduced in section 2.1.3 could be used to identify ROIs that belong to OARs.[7] On the other hand, this attribute may not be used in all data sets as it is not mandatory to specify it.[8]

Although this information is present in our data set, a different strategy was used in order to have a more general approach. First of all, common names that belonged to target and helper ROIs where excluded.[9] Then, based on the fact that clinical guidelines often require the expert to contour all the organs in the disease site (even if some OARs may not be needed to plan the treatment), we assumed that OARs names were the most frequently encountered ones.[10] We then counted the frequency (or support) of each ROI label in the data set (which at most can coincide with the total number of patients) and sorted them in descending order having the most frequent in the first position and the least frequent as last. For each ROI in position "$i$" we calculated the cumulative sum of the frequencies of the ROIs in positions $[0, ..., i]$ included. By dividing this number by the total number of ROIs names in the data set, we obtained the percentage coverage of ROI names corresponding to taking the first $i$ most frequent ones. The results are reported in table 3.1. For example, if we take the first three most frequent ROI names (*"rectum", "blaas", "bulbus"*), we would include in the ground truth $540$ ROI over a total of $1515$ not-target ROIs, corresponding to a $35.6\%$ of coverage.

Finally, we plotted in figure 3.3 the frequency and the cumulative coverage in function of the ROI name. We observed a sharp drop in frequency between the 5th and 7th most popular ROI names, as well as a long tail of fairly infrequent names. The decision was to include the first 6th of those names, which granted a coverage of $1097$ ROIs ($72\%$) that was deemed to be *"good enough"* for our purposes.

[7] As one of the possible values is literally "ORGAN".

[8] It is common for open data sets to avoid this attribute. On the other hand, the data sets coming directly from clinical practice use it more consistently as it is employed as a parameter during the treatment planning process.

[9] This was done using a simple regular expression that can be found in appendix A.3.

[10] This assumption is based on expert clinical knowledge.

*Table 3.1: non-target ROI names sorted by frequency in descending order. The "cumulative sum" of row $i$ is obtained by summing to its frequency all the previous (higher) frequencies. The "coverage" is calculated by dividing the cumulative sum by* $1515$*, which is the total number of non-target ROIs.*

| ROI name | frequency | cumulative sum | coverage |
|---|---|---|---|
| rectum | 181 | 181 | 11.94 |
| blaas | 181 | 362 | 23.89 |
| bulbus | 178 | 540 | 35.64 |
| anaal kanaal | 173 | 713 | 47.06 |
| heup li | 157 | 870 | 57.42 |
| heup re | 156 | 1026 | 67.72 |
| dunne darm | 71 | 1097 | 72.40 |
| blaas-PTV | 36 | 1133 | 74.78 |
| rectum-ptv | 35 | 1168 | 77.09 |
| rectum-PTV | 35 | 1203 | 79.40 |
| blaas-ptv | 34 | 1237 | 81.65 |
| RectumOverlap | 22 | 1259 | 83.10 |
| NS_Fiducial | 19 | 1278 | 84.35 |
| rectum overlap | 14 | 1292 | 85.28 |
| rectum in PTV | 14 | 1306 | 86.20 |

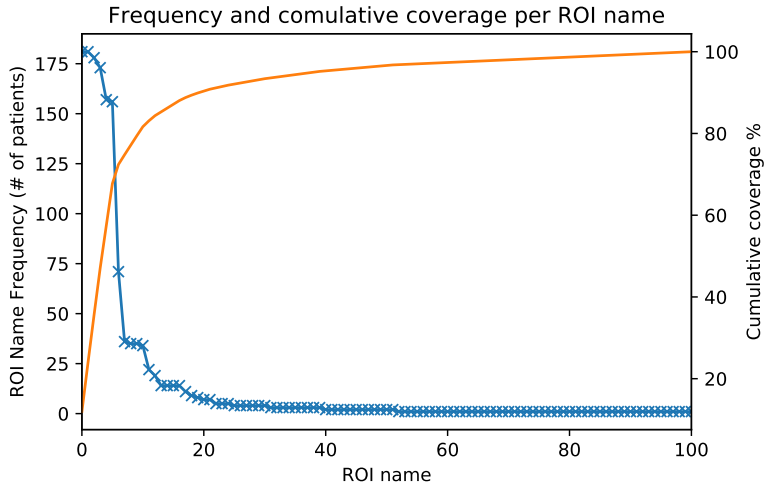



*Figure 3.3: Every entry on the x axis is an ROI name; on the left y axis, in blue, we have the frequency of each ROI name (i.e. the total number of patients in which that ROI name appears); on the right y axis, in orange, we have the cumulative coverage corresponding to taking the first x ROI names.*

As such, the final list of OAR names is: *"rectum", "blaas", "bulbus", "anaal kanaal", "heup li", "heup re", "dunne darm"*.

In order to help the reader to better understand the meaning of each name, from now on we will only use their translation obtained by applying the TG-263 guideline (see section 2.2.2). The result is reported in table 3.2. A representation of all the OAR geometries for a patient having all the ROI listed in table 3.2 can be seen in figure 3.4.

*Table 3.2: Translation from original OAR names to TG-263 compliant ones*

| Index | Original Name | TG-263 Name |
|---|---|---|
| 0 | rectum | Rectum |
| 1 | blaas | Bladder |
| 2 | bulbus | PenileBulb |
| 3 | anaal kanaal | Canal_Anal |
| 4 | heup li | Femur_L |
| 5 | heup re | Femur_R |
| 6 | dunne darm | Bowel |
| 7 | BODY | Body |

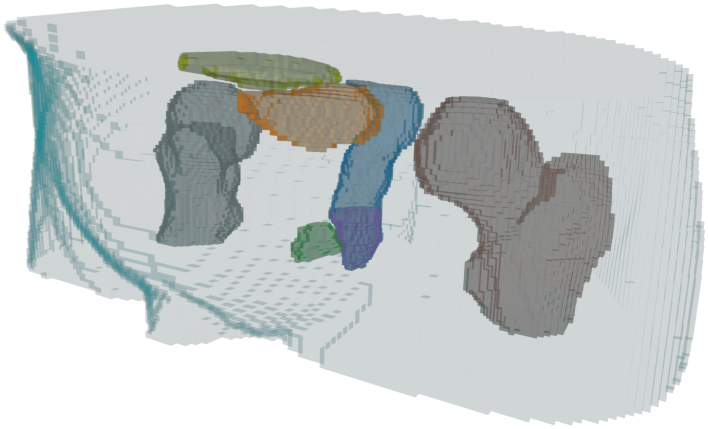


*Figure 3.4: OAR geometries contained in a patient. Rectum in dark blue, Bladder in orange, PenileBulb in green, Canal_Anal in violet, Femur_L in brown, Femur_R in grey, Bowel in yellow, Body in light blue.*

An alternative method based on frequent item sets mining yielded the same results as the one explained above (see appendix A.1).

### Patients Partitioning

After having selected the list of OAR names to be included in the ground truth, it was necessary to distribute each patient either in the train, evaluation or test data set.

In order to make this decision, we considered how many of the selected OARs in table 3.2 were actually contained in each patient. In figure 3.5 we can see that roughly one hundred patients have at least 6, and we reach 152 total patients if we factor in also the patients with seven of the selected OARs, while a total of 29 patients have less than 6 covered ROIs. We decided to compose the train and evaluation data set with a total of 100 patients that had at least 6 OARs (70 for train and 30 for evaluation, with random splitting), with the assumption that this will relegate peculiar cases to the test set. The rationale behind this decision was that it would allow us to verify how the model behaves in a sub-optimal scenario once deployed. In particular, to see if the inference probability could be used as a measure of confidence to discard such cases. Moreover, this setup left the test set with 81 patients, which is close to the average size of medical research data sets that contains RT structures [97].

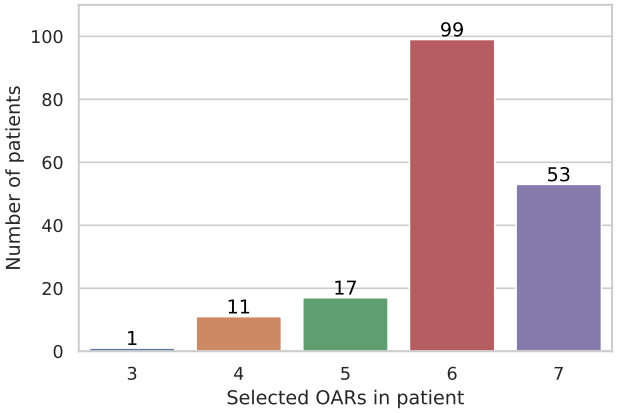



*Figure 3.5: Histogram of number of patients in function of the number of selected OARs contained in the patient.*

For reference, table 3.3 reports the number of patient and OARs per each set, while table 3.4 contains the per OAR break-down.

*Table 3.3: Number of patients and number of OARs contained in each split of the ground truth.*

| | Patients | OARs |
|---|---|---|
| train | 70 | 450 |
| evaluation | 30 | 186 |
| test | 81 | 461 |

*Table 3.4: Number of OARs contained in each split of the ground truth.*

| | Rectum | Bladder | PenileBulb | Canal_Anal | Femur_L | Femur_R | Bowel |
|---|---|---|---|---|---|---|---|
| train | 70 | 70 | 70 | 70 | 70 | 69 | 31 |
| evaluation | 30 | 30 | 30 | 30 | 30 | 29 | 7 |
| test | 81 | 81 | 79 | 73 | 57 | 57 | 33 |

## 3.4 Encoding and Model Training

### 3.4.1 2D Transverse Slices

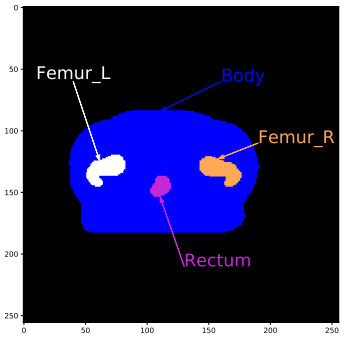
Rather than already providing final results, the goal of this section was twofold: first to familiarize with the data set, and second to obtain an estimate of the

difficulty of the problem from a classification standpoint. The most closely related work found in the literature was the study from Rozario [95], in which the ROI classification was done using a 2D CNN. As already noted in section 2.5, no comparison with different and more traditional methods was provided, and the sensational results obtained suggest the fact that the problem may be easier than expected. For this reason, we decided to use Rozario's work as a starting point and borrow the idea of the 2D encoding (with some modifications) of the OAR, but using simpler classification models.

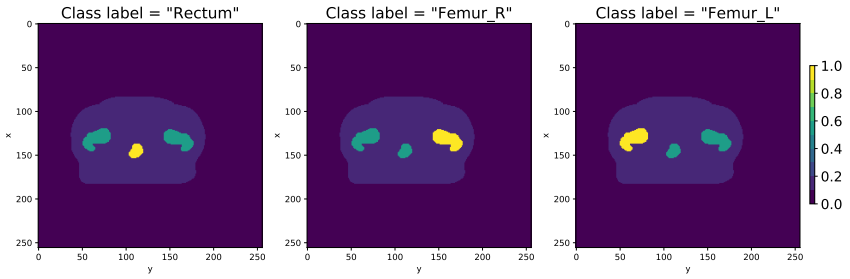
### Encoding

Our decision was to modify slightly the encoding scheme used in [95] (explained in detail in section 2.5) by adding also the pixels of the body to the 2D data instance. In our case, the final values of a pixel could be: $1.0$ if it was in the OAR we want to classify, $0.5$ if it was contained in one of the other OAR in the slice, $0.1$ if it was in the body, $0$ if it was outside the body.

Following the example in figure 3.6, there is one *"original"* transverse slice containing the following OARs: Rectum, Femur_L, and Femur_R. Starting from the Rectum, a new transverse slice of size $256 \times 256$ is generated with all its values set to $0$ (in violet). Then, the pixels belonging to the rectum are set to value $1$ (in yellow); the pixels of the Femur_L and Femur_R are set to value $0.5$ (in green); finally, the pixels of the body are set with value $0.1$ (in light blue). The resulting 2D instance is the leftmost one in figure 3.6b, and its associated class label is "Rectum". The same procedure is repeated for Femur_L and Femur_R, thus creating three input classification instances from a single transverse slice.




*(a) Original CT slice*




*(b) Generated instances for training, validation and testing.*

*Figure 3.6: (a) The original transverse slice. Each OAR is depicted in a different color but is effectively a separate tensor; (b) Three instances generated from the slice above. The OAR that we aim to classify is yellow, the other OARs in green, the body in light purple. The class associated to the instance is reported on top.*

As stated in section 2.5, this kind of encoding alters the number of instances belonging to each class. Simply put: the "taller" (longer on the z direction) an OAR, the more instances will be generated. This effect on the distribution of the number of instances per class is clearly visible in figure 3.7, where the PenileBulb and the Bowel are sensibly less represented than the others. As we can see in figure 3.4, the former is a rather small OAR, while the latter is already less frequent but also tends to be present in a small amount of slices because it is far from the treatment volume.

### ML Algorithms

To establish a baseline we decided to train first a simple DT model. The rationale behind this choice is that the DT is one of the simpler classification models, and it has the upside of being able to deal with multiclass classification problems in a native fashion. Having used another classifier, for example

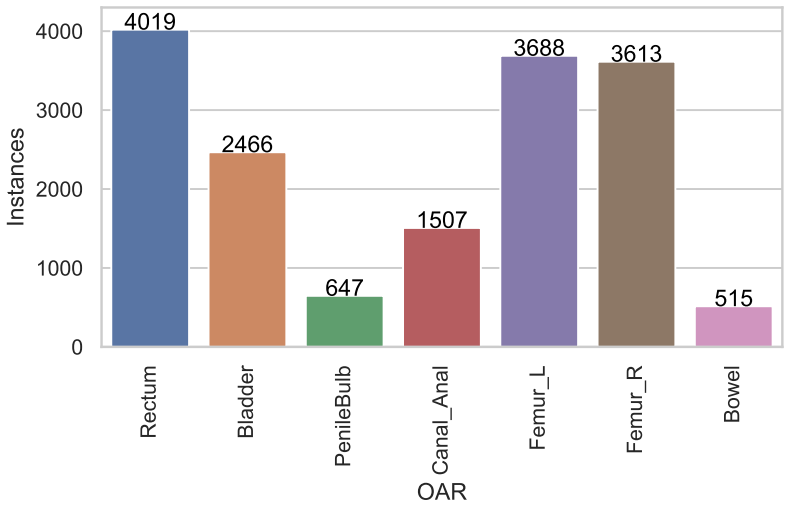



*Figure 3.7: 2D encoding; number if instances per each class on the whole data set.*

a logistic regressor, would have requested to train a one-vs-rest model for each class. Considered the class imbalance reported above, we wanted to avoid any up/downsampling of the data set, thus maintaining a simple and *"less moving parts"* approach, as declared in section 1.5.

The input is composed of a 2D matrix of floats that can be interpreted as a simple image of $256 \times 256$ pixels. To be used by the classifier, the matrix is flattened into a $65536$ elements vector. Hence, from the point of view of the model, each element of the matrix (a pixel) is a different and independent feature.

After obtaining the first results, we selected a tree ensemble classifier, namely RF, as being the natural evolution of the DT, while still conserving the desired characteristic of supporting natively multi-class classification tasks.

With this encoding based on [95], every slice was considered as a separate independent instance. This is an excessively constraining framework, given that at inference time we are aware of which set of instances belong to the same OAR in a specified patient; what we don't know is which OAR it actually is. For example, for a given unknown OAR $j$ made of 100 slices, we may have 80 slices which are classified as a Rectum and 20 slices as an Canal_Anal, but the OAR can be only one of the two. We exploited the information contained in the inferences of the model and implemented a majority voting strategy[11]

[11] To not confound with majority voting used in ensemble methods. Here the voters are not the learners but the inference probability obtained on each 2D instance.

saying, for example, that the OAR above is a rectum rather than a Canal_Anal. This approach does not require to modify the trained model, as it is based on the already obtained classification probability distributions.

More formally, consider $X_{ij} = \{x_1, ..., x_K\}_{ij}$ being the set of all the $K$ slices belonging to patient $i$ and unknown OAR $j$. After classifying a certain $x_k$, the model yields a discrete probability distribution expressed as a vector $\bar{p}_k = [p_{k1}, .., p_{k7}]$ with $k \in [1, .., K]$ where $\sum_{c=1}^{7} p_{kc} = 1$ and each $p_{kc}$ represent the probability of the instance $x_k$ to belong to class $c$. Classifying all the instances in $X_{ij}$ we obtain the set of probability distributions $P_{ij} = \{\bar{p}_1, ..., \bar{p}_k\}$ that we can express as a $k \times 7$ matrix of element $p_{kc}$. We then sum the components of this matrix column-wise (i.e. class-wise) to obtain a $1 \times 7$ vector that we normalize by the sum of its component to have an overall probability of 1 as in

$$\bar{p}_{ij} = \frac{1}{\sum_{c=1}^{7} \sum_{k=1}^{K} p_{ck}} [\sum_{k=1}^{K} p_{k1}, ..., \sum_{k=1}^{K} p_{k7}].$$

In this framework, $\bar{p}_{ij}$ is the discrete probability distribution associated to the OAR $i$ in patient $j$ obtained by majority voting. As usual, the position of the element in the vector with the highest value corresponds to the predicted class

$$\hat{y}_{ij} = argmax(\bar{p}_{ij}).$$

This moves the problem from a per OAR per slice classification, to just a per OAR classification, reducing the number of instances and the class distribution to the ones in table 3.3 and 3.4.

The results obtained by this majority voting approach should suggest if considering the whole 3D OAR structure is better than separate the problem in a set of independent 2D instances.

The results obtained in this iteration gave us important insights on which feature and encoding may work best in the future iteration. This process is discussed in greater details in the results section.

### 3.4.2 3D Feature Engineering

The results obtained with the 2D encoding highlighted the importance of considering the full volumetric geometry of the ROI, while still relying on information of the spatial context. This led us to develop a set of hand-engineered features comprehensive enough to offer good separation between the classes. In order to compare the results with the ones obtained in the previous step, we

decided to use the same classification models. Using a *"raw-like"* 3D encoding, having each voxel as a feature, would have resulted in high memory consumption given that the ML library we used [98] had reduced support for large data sets.[12] Coping with this constraint would have requested to implement a great deal of custom data pipelines, which, besides adding extra complexity, would have introduced the risk of creating hard-to-detect bugs in the process. Having a small set of hand generated features allowed us to train models efficiently, which gave us the opportunity to do extensive hyperparameter tuning by means of cross validation.

### Encoding

We can divide the engineered features into two groups: OAR specific features and spatial context features.

OAR specific features are features that derive from a single OAR as an independent entity. That is, regardless of its location with respect to the patient body and all the other ROIs in the patient. The features belonging to this group are the following:

- **volume**: the total number of voxels contained in the OAR.[13] Intuitively, OARs like the Bladder are sensibly bigger than the PenileBulb, hence will have higher volume.
- **surface**: the total number voxels on the surface of the OAR. The surface was obtained by applying a 3D Sobel-Feldman operator[14] [99]. OARs with more complex geometries, like the Rectum and the Femurs should have higher surface than others.
- **bounding box dimensions**: the three dimensions $(x,y,z)$ of the bounding box enclosing the OAR. This features should highlight OARs that have a prominent extension in one direction, for example Femurs.
- **maximum intensity of the projection**: the maximum intensity of a pixel after projecting the OAR on one of the anatomic planes.[15] The projection is calculated by summing all the voxels values on the direction perpendicular to the anatomic plane considered.

[12] Using data-intensive frameworks like Spark was not possible due to technical limitations and the constraint that the data could not leave RaySearch premises.

[13] That is, the total number of voxels set to 1 in the tensor representing the OAR geometry (see section 3.1)

[14] Also known as *"edge detection filter"*.

[15] Section 2.5

- **surface over volume**: the surface itself may introduce some confusion. The Rectum and the Bladder may have similar surface but sensibly different volumes. Dividing the surface by the volume should allow to distinguish between more spherical OARs than elongated ones. This feature is also known as *circularity* [80].

- **Normalized Central Moments**: NCMs calculated on the projection of the ROI on anatomic axes, as explained in section2.3. NCMs aim to identify asymmetry in the mass distribution of the shape. They were also chosen because they are invariant to translation and scale transformations. This is important because the features introduced above are already aimed to represent the different scale of the OARs. Translation invariance is also important given that we decided to encode the spatial context with dedicated independent features. We also preferred to avoid local surface descriptors [81] as they solely rely on the surface in the proximity of key points. This is because while the delineation of most of the OAR shape is generally coherent and well-defined, deciding how the proximity of the surface should be segmented may depend on different factors and can sensibly vary between medical experts [100]. The maximum computed order for NCMs was 3 (included). Meaning that for each projection we have 13 NCMs[16], for a total of 39 NCMs.

Except for NCMs, all the features listed above are considered *"simple shape descriptors"* [80], and are often used in combination with other descriptors because they are not deemed to be enough discriminative when employed alone.

To encode the spatial context we wanted to express the position of the OAR with respect to other points of reference. To have an easy to compute and simply understandable representation of the position of the OAR, we used its center of mass (section 2.3), which is a simple 3D vector. We then decided to compare it to the body center of mass, thus rendering this approach independent from any choice of system of reference.[17] The comparison is done by simply calculating the difference vector between the two points as explained

[16]For each projection, $p$ and $q$ have value $[0, 1, 2, 3]$, which means $4 \times 4 = 16$ moments, but NCMs are defined only for $p + q \geq 2$ so we have to subtract 3 moments that are not defined, hence 13 moments per projection.

[17]Provided that the orientation of the patient is always the same (see section 1.8).

in the following equation:

$$\bar{r} = \bar{o} - \bar{b} = \begin{pmatrix} o_x - b_x \\ o_y - b_y \\ o_z - b_z \end{pmatrix} = \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix}.$$

Where $\bar{r}$ is the obtained difference vector, $\bar{o}$ is the OAR center of mass and $\bar{b}$ is the body center of mass. Figure 3.8 has a graphical representation of the obtained vector for the case of the Femur_R, while in figure 3.9 we can see the 3D representation of the centers of mass of each OAR in the body. Finally

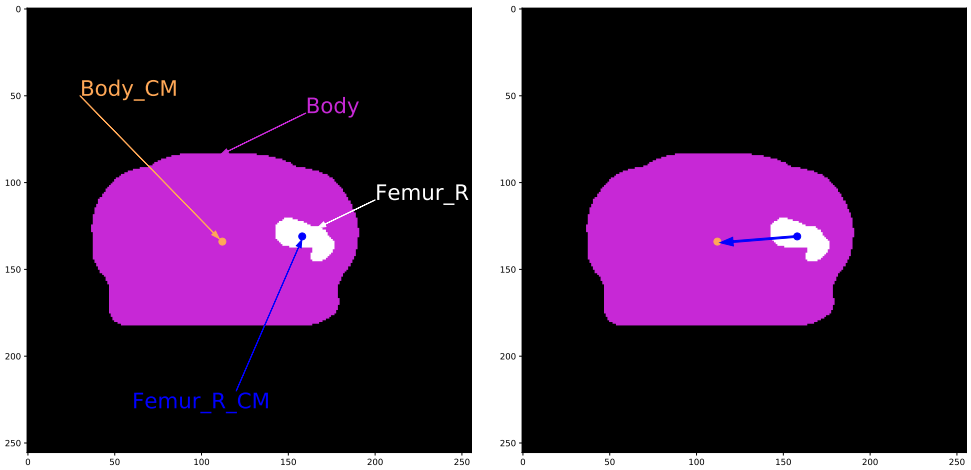



*Figure 3.8: On the left, the Body and Femur_R with their respective centers of mass. On the right, the difference vector between the two centers of mass.*

we decided to use as features only the components on the transversal plane ($r_x$ and $r_y$), because series of transver slices may be missing in clinical data sets, thus representing incorrectly the distance on the $z$ axis. Even though our data set did not present this problem, we preferred to follow a more robust approach. Moreover, we decided to compare only to the body center of mass rather than other possible reference points (like the compounded mass of all the other ROIs) because the body ROI is always segmented and manifest a very high level of consistency.

### ML Algorithms

We decided to select the same model architectures used in the previous iteration: DT and RF. We also decided to train with and without NCMs to assess their effect on the overall performance. For both cases, when RF was used,

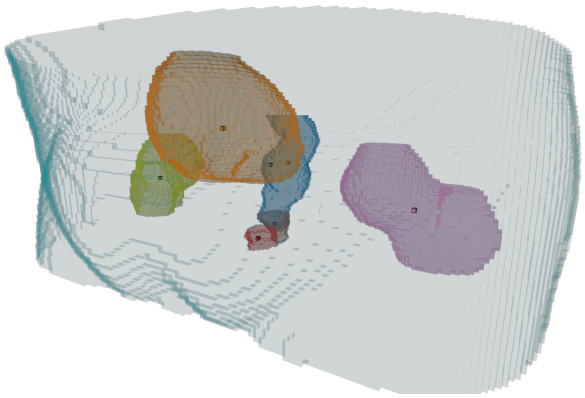


*Figure 3.9: OAR in body with their respective centers of mass highlighted as a single voxel*

grid search cross validation was performed to find the best set of hyperparameters. For more details about the tuning process, please refer to appendix A.6.

### 3.4.3 3DCNN

The recent advancements in computer vision have been mainly driven by the adoption of deep CNNs [11]. In order to have a comparison with a more modern approach, we decided to select a 3DCNN to perform automatic feature learning.

#### Encoding

The advantage of using a DL approach is to avoid the manual feature engineering. For this reason, the input was used in a form as close as possible to the original 3D geometric encoding of the OAR (see section 3.1). To represent the spatial context we worked on the same line of what was done with the 2D encoding of section 3.4.1, but instead of using a different value to distinguish between the OARs,[18] we used a separate channel for each one. Each of the three channels contained a 3D binary tensor, one with the OAR we aim to classify, one with all the other OARs in the patient,[19] and one with the Body. A representation of the final input instance can be found in figure 3.10.

CNNs require all the input samples to be of the same size, unfortunately in

[18]The one we want to classify, the other OARs, and the body

[19]Effectively obtained with a 3D "or" operation.

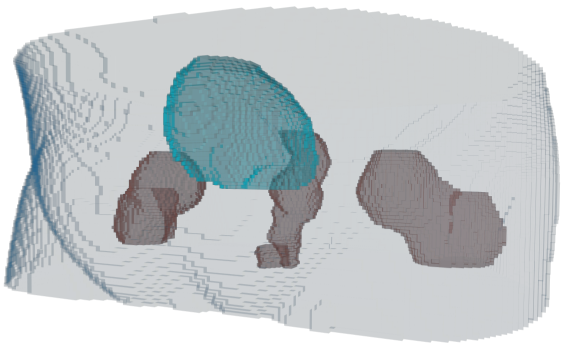


*Figure 3.10: Input of the 3DCNN. The OAR we aim to classify (the Bladder in this case) is highlighted in blue. The other OARs are in brown while the patient body is in light grey. Note that the value contained in each voxel is always 1, as the three groups above are in separate channels.*

our case the number of transversal slices in the representation may vary from patient to patient (see section 2.1.1). To solve this issue we zero-padded each input instance on the z direction to reach 128 voxels,[20] adding the same amount of empty voxels above and below the patient representation. The final shape of the input tensor after padding was $256 \times 256 \times 128 \times 3$ respectively for the x, y, z directions and channels.

### ML Algorithms

The network chosen was a VOXNet [89] which architecture is discussed in details in section 2.4.5. Such an architecture was chosen for a series of reasons. First, it was a well documented and published model with over 260 citations at the time of the writing. Second, it contained a low number of parameters compared to more complex 3DCNNs which allowed us to complete the training process in roughly one hour (see appendix A.8). Third, despite its simplicity, it managed to obtain discrete results on the modelnet40 data set [90]. We thought this to be particularly important because the encoding of modelnet40 resembles the one used in this work (an example can be seen in figure 3.11).

The model was trained from scratch (no fine tuning) and the only difference

[20] 128 was the smallest power of 2 that allowed to include all the patients without discarding slices.

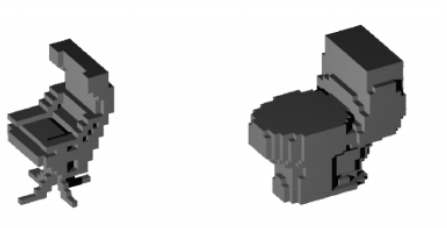


*Figure 3.11: Two instances in the modelnet40 data set.*

with a vanilla VOXNet architecture is the last fully connected layer, which in our case was composed of seven units (one per each target class). The decision to avoid the down-sampling of the input resulted in instances with a large memory footprint. For this reason, the batch size was only consisting of 8 examples, requiring 58 batches to complete an epoch.

## 3.5 Classification Evaluation

The problem is formulated as a multi-class mono-label classification task [85]. As such, for each instance of the evaluation set, the ML model yields a vector of seven real values in the interval $[0, 1]$ corresponding to the probability of the instance belonging to one of the classes reported in table 3.2. The labels are mutually exclusive, implying that the sum of the element in the probability vector is 1. The instance is considered to belong to the class with the highest probability value.[21]

Two metrics were mainly considered for evaluating the performance of the model: weighted precision and recall. They are calculated in the following way. First, the precision and recall of each class is computed, by considering each class as a binary classification problem in a one-vs-rest fashion.[22] Then the values obtained are averaged with weights corresponding to the class support (i.e. the number of instances for each class in the ground truth labels),

$$Prec = \frac{\sum_{i=1}^{7} w_i Prec_i}{\sum_{i=1}^{7} w_i}; Rec = \frac{\sum_{i=1}^{7} w_i Rec_i}{\sum_{i=1}^{7} w_i}. \tag{3.1}$$

Where $w_i$, $Prec_i$, and $Rec_i$ are respectively the support, precision and recall of each class. Using the weighted average accommodates for the class imbalance concerning the under-representation of the Bowel class.

[21] Probability thresholds are discussed in the next section.

[22] For class $i$, what is labeled to be of class $i$ is a positive, what is label with a class different than $i$ is a negative.

We also made extensive use of the confusion matrix to better highlight the behavior of the model with respect to specific classes.

ROC curves[23] [70] were initially considered but later discarded due to the lack of a uniquely accepted formulation for multi-class mono-label classification tasks [101, 102, 103, 104]. Although in principle the same approach used for weighted precision and recall could have been used, it would have required a great deal of custom implementation and testing of the metric, which we preferred to avoid. Moreover, it would have not been representative of the actual area (hyper-volume in this case) under the curve, which is what it is actually interesting to consider.

The implementation of the metrics was provided by the *scikit-learn* library [98], please refer to appendix A.7 for further details.

## 3.6 Inference Probability Evaluation

This step was completely dedicated to answering the second research question: if it is possible to use the inference probability as a measure of confidence.

Keeping a pragmatic approach, we decided to answer it by implementing the concept of a reject class. The basic idea is rather simple: given an instance, if the maximum probability returned by the classifier is lower than a threshold, the instance is rejected [105]. This translates into assigning to the instance a "reject" class label, which does not exist in the ground truth but has the sole goal of signaling that the inference of the model on that instance is not "trustworthy" and, as such, the instance should be manually checked by a human expert. Keeping in mind that, as stated in section 1.4, we are not aiming for full automation, but manual labor reduction in front of reliable predictions.

If the classifier behaves correctly (i.e. all the incorrect classifications have a low inference probability), by increasing the threshold the precision is expected to increase[24], while the recall should decrease (as instances are effectively "lost" by refusing to classify them).

To better understand the behavior of the model one can look at the rejection ratio (i.e the number of rejected instances over the total number of instances) in function of the probability threshold, as well as the trend of the precision and recall. It is also helpful to look at the histogram of the obtained probabilities and locate where the misclassification cases are.

[23] Namely the area under the ROC curve

[24] Small fluctuations are possible due to the different weights assigned to each class. Note that the "reject" class does not exist in the ground truth, hence it does not have an associated precision or recall, but its effect is to alter the classification metrics of the other classes.

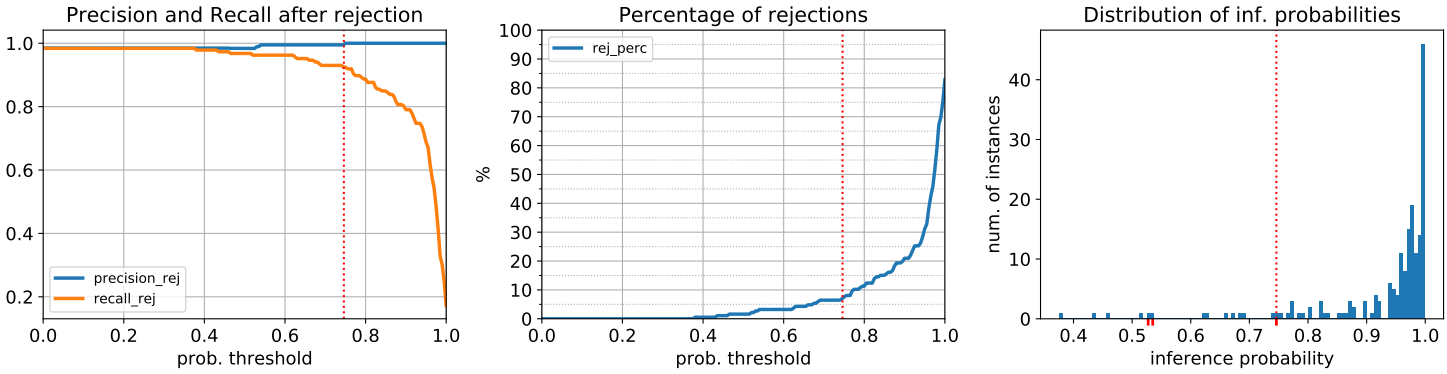



*Figure 3.12: 3D engineered features, RF rejection curves.*

Figure 3.12 shows the rejection curves on the evaluation set for the RF model trained with the 3D engineered features without NCMs. In the leftmost plot, we have the weighted average precision (blue line) and recall (orange line) on the vertical axis, in function of the probability threshold on the horizontal axis. For example, if we select a rejection threshold of $0.9$, the precision will rise to 1 but the recall will go down to $0.8$. The vertical dotted line, in all three of the plots, corresponds to the highest probability of a misclassification case. The central plot represents the reject rate on the vertical axis in function of the same probability threshold as above. Looking at the plot we can see that a $10\%$ rejection rate correspond to a probability threshold of around $0.78$. Also, by increasing the rejection threshold we increase the rejection rate (obviously) up to a tipping point where only the inferences with probability equal to 1 remain. The third and last plot is a histogram of inference probabilities. The red ticks on the horizontal axis correspond to the probabilities of all cases of misclassification in the evaluation set. We can see that the distribution is skewed to the right and that all the misclassification cases have a lower probability compared to most of the cases. Please note that the horizontal scale of the rightmost plot does not start from zero.

The final probability threshold is inferred by looking at the rejection curves on the evaluation set. To be effective, the threshold must be greater than the highest misclassification case. As an educated guess, we used a probability threshold corresponding to a $10\%$ rejection rate, this because it coincides with reducing the time needed to clean the data set by one order of magnitude compared to a complete manual cleaning.

Finally, the combination of the classifier and the rejection class (with its probability threshold) are applied to the test set. Then two aspects are assessed: first that all misclassifications have been rejected, i.e. the accuracy on the not-

rejected instances is 100%; second, the percentage of rejected instances over the total.

The goal of these two aspects is to estimate the compromise between model accuracy and the need for manual checking. That is, even if a model is 100% accurate there is no point if it is rejecting 90% of the data set, because expert intervention will be required in nine over ten cases.

# Chapter 4

# Results

This chapter contains the results obtained following the method outlined in chapter 3. The results are arranged in six main sections respectively aimed to expose the following aspects: *i*) establishing a baseline with a simple DT model based on the 2D slices encoding, and comparing it with a random classifier (section 4.1); *ii*) evaluating the results of a more complex tree-based ensemble model on the same encoding (section 4.2); *iii*) investigating the importance of considering the full 3D OAR volume by implementing a slice based majority voting (section 4.3); *iv*) evaluating the performance of tree-based classifiers and CNNs on 3D based features (section 4.4); *v*) evaluating if the inference probability could be used as a measure of confidence through the implementation of a reject class mechanism (section 4.5); *vi*) analyzing the rejected inferences to qualitatively evaluate the results of the reject class mechanism (section 4.6);

Finally, section 4.7 contains a summarizing table of all metrics collected during the performed experiments.

## 4.1 Baseline: DT on 2D slices

As stated in the method section, the first iteration was aimed at obtaining an estimate of the problem complexity. It was also used to collect pointers on the kind of encoding and features that were more effective in the classification effort, in order to further refine our strategy in the next iterations.

Recall that the encoding used for the ground truth[1] was reported in section 3.4.1, where each data instance was a transverse 2D slice containing the OAR we wanted to classify, the nearby OARs and the body, all with a different

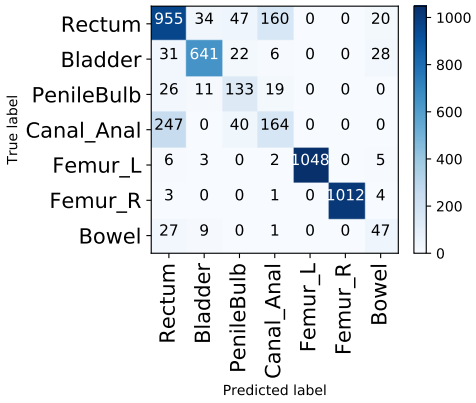
[1] Which was a modification of the encoding used in [95].

pixel value. Please refer to figure 3.6 at page 47 for a clear representation of the input. Given this encoding (each slice per OAR was a different instance), the number of instances was much higher than the total number of OARs, as clearly reported in figure 3.7.

In figure 4.1 we report the classification results on the evaluation set obtained by a vanilla DT. The parameters used[2] were the following: the criterion for splitting was the Gini criterion; the depth was not restricted, meaning that the tree was free to grow until a leaf node was pure or no more splits were possible; the minimum number of samples per leaf was one, again not restricting the growth of the tree. All the features (i.e. the $256 \times 256$ pixels) were used.

*Figure 4.1: 2D transverse slices encoding, DT results on the evaluation set.*

| | f1-score | precision | recall | support |
|---|---|---|---|---|
| weighted avg | 0.8418 | 0.8419 | 0.8418 | 4752 |




The model was particularly efficient in distinguishing the slices pertaining to the Femur_L and Femur_R (which were the most displaced), but there was a great deal of confusion around the Canal_Anal and the Rectum. This was due to the fact that the Canal_Anal was usually contoured as a subset of the Rectum, being its terminal part. This implied that the two OARs overlap for most of the slices where the Canal_Anal was defined, an aspect not considered by [95]. Nevertheless, the Bowel and the PenileBulb, two OARs with different horizontal extensions but similar number of slices per OAR, had sensibly lower performances. The overall f1 score achieved by this simple classifier was $84.18\%$. Compared to a random baseline classifier[3], that would achieve roughly a $14.8\%$ score[4], it was already a very interesting result.

[2]The default ones offered by the implementation [98].
[3]Picking uniformly at random one of the seven labels.
[4]Not taking into account class imbalance.

## 4.2 Random Forest: an ensemble of decision trees

| | f1-score | precision | recall | support |
|---|---|---|---|---|
| weighted avg | 0.8796 | 0.8800 | 0.8792 | 4752 |

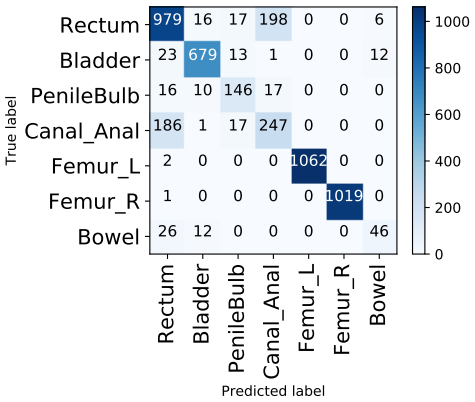
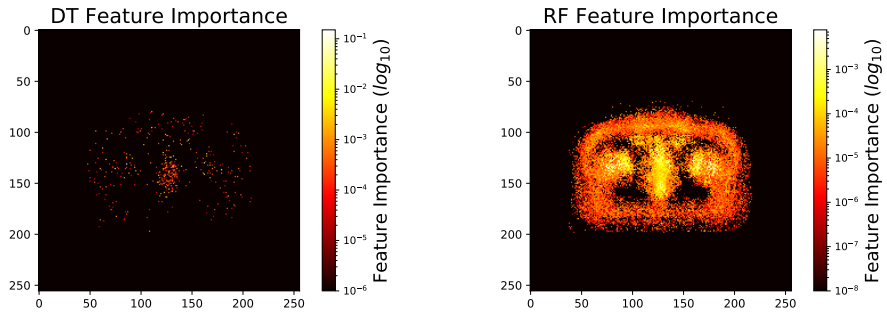



*Figure 4.2: 2D transverse slices encoding, RF results on the evaluation set.*

Once we increased the complexity of the model to its natural evolution, a RF model, we could see in figure 4.2 a significant improvement on the overall score, as well as the performance on Bowel and PenileBulb. The number of Canal_Anal slices correctly classified went from 164 to 247, but there was still a good amount of confusion. The model was made of 50 decision trees (also known as *estimators*), each of them with the same parameters as the ones used before. The maximum number of features for each split was the square root of the total number of features[5].

[5] Again the default values offered by [98].



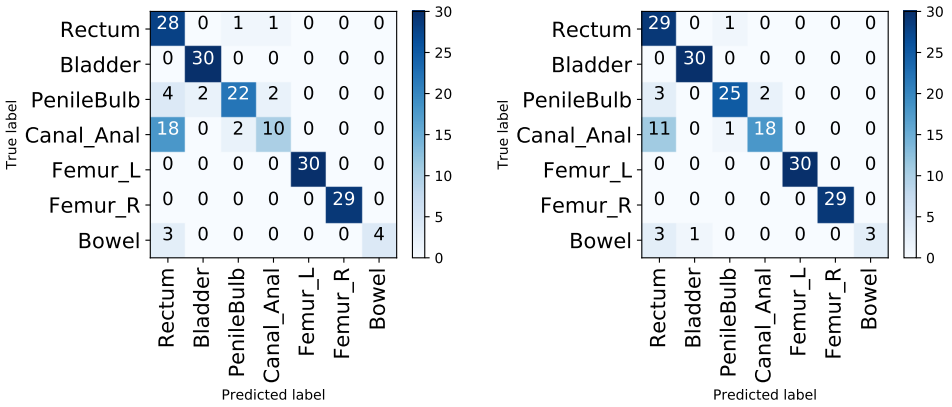

*(a) DT* *(b) RF*

*Figure 4.3: Feature importance of the DT and the RF, respectively the left and right plot. Mind the logarithmic color scale.*

Figure 4.3 represents the feature importance of the two models. In both cases it was calculated as the sum of information gain brought by every feature used for a split, normalizing the total importance to 1. Keeping in mind that each independent feature was a pixel of the 2D slice in figure 3.6b, it is clear that both the models were looking at the whole content of the patient, thus exploiting the spatial context on the 2D representation. As expected, due to the random sampling of features and instances performed during the training, the RF model had a more complete view of the patient with a finer distribution of feature importance. Nevertheless, the highest features were in the proximity of where the OARs are usually located. Interestingly enough, the shape of the body and its borders seemed to play a non-negligible role.

## 4.3 Slice Based Majority Voting: towards 3D features

As explained in section 3.4.1, we also used the inference probability distributions obtained by the two models to implement a majority voting mechanism based on the 2D slices, as a mean to suggest if considering the whole 3D structure of the OAR could improve classification performance. The results of this approach for the DT and the RF are reported in figure 4.4.




*(a) DT* *(b) RF*

| | DT | | | RF | | | |
|---|---|---|---|---|---|---|---|
| | f1-score | precision | recall | f1-score | precision | recall | support |
| weighted avg | 0.8396 | 0.8573 | 0.8226 | 0.8942 | 0.9071 | 0.8817 | 186 |

*Figure 4.4: 2D transverse slice encoding, majority voting results.*

Looking at both the confusion matrices, we can immediately see that the misclassification of the Rectum (true label) in Canal_Anal (predicted label) was almost totally removed (totally in the RF case). This is due to the fact that the Rectum was much taller than the Canal_Anal (on the $z$ direction), hence it had more slices that were likely to be classified correctly, thus shifting the majority vote to its correct classification. For the same reason, the majority voting had little effect on the opposite case, Canal_Anal (true label) classified as Rectum (predicted label). The Canal_Anal was short and most of its volume was shared with the bottom part of the Rectum. This latter aspect was particularly penalizing in the DT case, which obtained a lower overall f1 score. On the other hand, the RF f1 score improved by $1.5\%$. The change in class distribution might have also accounted for the DT performance.

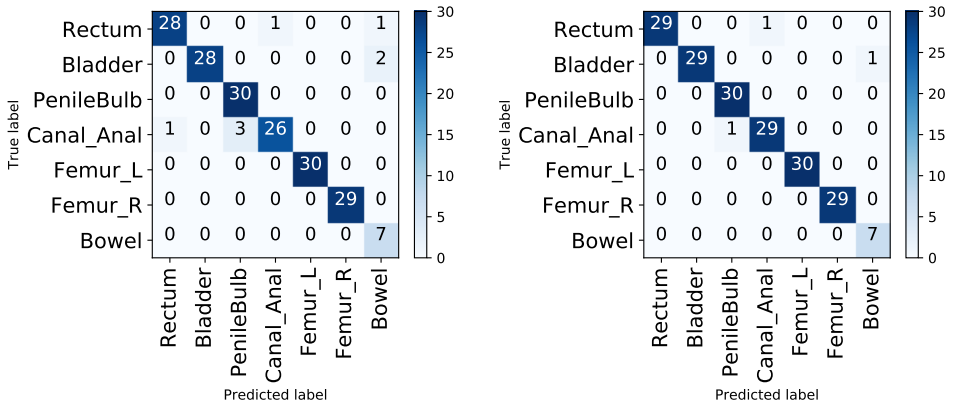
*Table 4.1: Classification results on the evaluation set for DT and RF approach based on 2D encoding (section 3.4.1).*

| enconding | model | evaluation set | | |
|---|---|---|---|---|
| | | prec | rec | f1 |
| 2D slices | DT | .8419 | .8418 | .8418 |
| | RF | .8800 | .8792 | .8796 |
| 2D maj | DT | .8573 | .8226 | .8396 |
| | RF | .9071 | .8817 | .8942 |

In light of these results and the feature importance plots, we decided to focus our efforts on encodings that kept considering the whole OAR volume, as well as its spatial context. This is also the reason why we did not perform any experiment on the testing set and the reject class implementation, as we found more promising to focus our resources on 3D approaches.

## 4.4 Classification of 3D Volumes

Following the results exposed above, the two encoding and classification strategies reported in section 3.4.2 and 3.4.3 were designed. The first one was based on manual feature engineering and used the same ML algorithms as before (DT and RF), while in the second one the features were automatically learned from the data during the training of a 3DCNN. Moreover, regarding the manually engineered features, we trained our classifiers with and without NCMs in the feature set. For completeness, we report both results with the aim of highlighting which role they play in the classification performance as well as the inference probability evaluation that is reported in the next section.



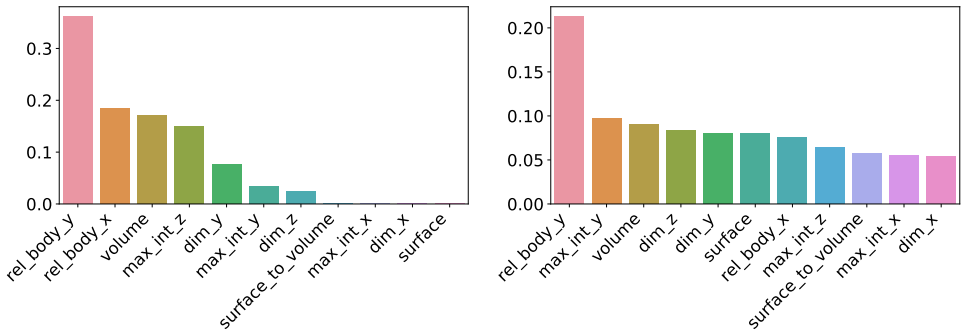

*(a) DT* *(b) RF*

| | DT | | | RF | | | |
|---|---|---|---|---|---|---|---|
| | f1-score | precision | recall | f1-score | precision | recall | support |
| weighted avg | 0.9597 | 0.9625 | 0.9570 | 0.9843 | 0.9847 | 0.9839 | 186 |

*Figure 4.5: 3D engineered features without NCMs, DT and RF results on the evaluation set.*

Figure 4.5 contains the results of the DT and the RF on the manually engineered features without NCMs. The DT had the same parameters used in the previous section, while for RF they were selected after performing five fold cross validation (as reported in appendix A.6) and were the following: 256 estimators, 3 max features, 10 maximum depth and 1 minimum samples per leaf.

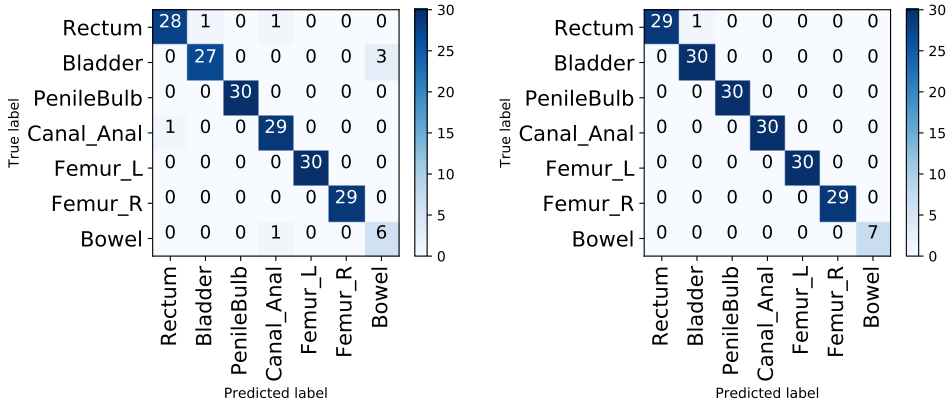
Compared to the ones obtained in the previous section, the improvement is clear. The confusion surrounding Rectum and Canal_Anal was almost totally removed and even small OARs were classified correctly. On the quantitative side, the f1 score improved by $12.01\%$ and $9.01\%$ respectively for the DT and the RF, compared to the majority voting results.




*(a) DT* *(b) RF*

*Figure 4.6: Feature importance of the DT and the RF, respectively the left and right plot.*

If we take a look at the feature importance plot in figure 4.6, we can see how the spatial context played a major role in both models. In particular, the position of the OARs center of mass in respect to the body center of mass on the $y$ direction was the major contributor in increasing the information gain. If we look at the PCA plots in appendix A.4, we can see how easily this feature can be used to separate the two Femurs.



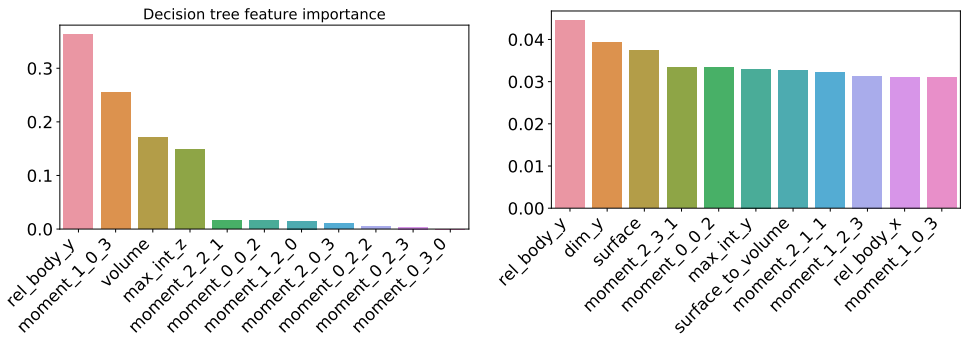

*(a) DT* *(b) RF*

| | DT | | | RF | | | |
|---|---|---|---|---|---|---|---|
| | f1-score | precision | recall | f1-score | precision | recall | support |
| weighted avg | 0.9639 | 0.9654 | 0.9624 | 0.9947 | 0.9948 | 0.9946 | 186 |

*Figure 4.7: 3D engineered features with NCMs, DT and RF results on the evaluation set.*

When the NCMs were added to the feature set, as in figure 4.7, the results sensibly improved and RF achieved almost perfect classification.

From figure 4.8, we can see that the spatial context kept playing an important role, even in this enriched set of features. However, in the RF case the moments were more important than in the DT. This effect was probably due to the fact that the number of moments was higher than the engineered features, thus when sampling the features in a node the probability of having only moments was higher, forcing the estimator to use them.




*(a) DT* *(b) RF*

*Figure 4.8: Feature importance of the DT and the RF, respectively the left and right plot.*

| | f1-score | precision | recall | support |
|---|---|---|---|---|
| weighted avg | 0.9788 | 0.9790 | 0.9785 | 186 |

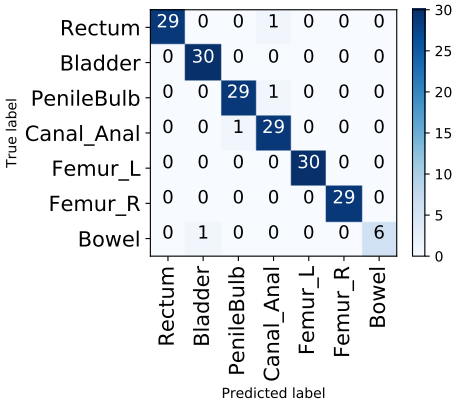



*Figure 4.9: 3DCNN, VOXNet results on the evaluation set.*

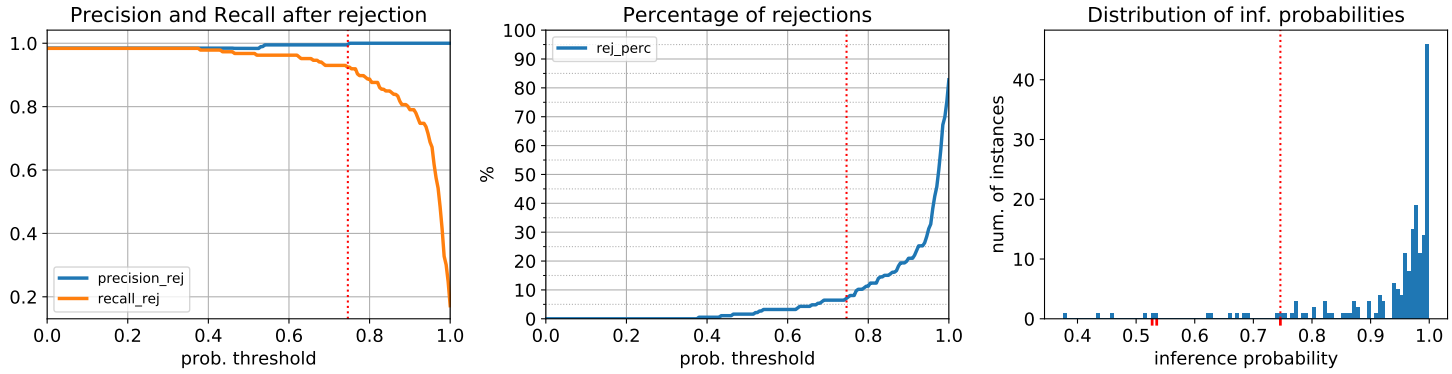
Finally, figure 4.9 contains the results obtained by the VOXNet 3DCNN architecture (see section 3.4.3). The details concerning its training process are reported in appendix A.8. The confusion matrix showed results comparable

to the ones of the RF model without NCMs, suggesting that also DL methods might be effective in the classification task.

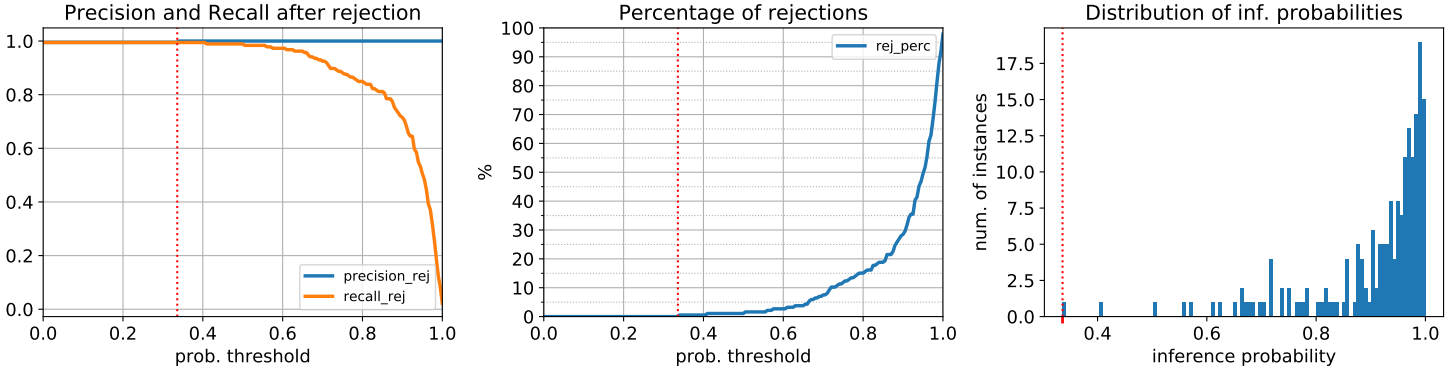
## 4.5 Inference Probability as Confidence Measure

In section 3.6, we explained in detail how the reject class mechanism was implemented and how it was used to evaluate if the inference probability yielded by the classifier could be used as a measure of confidence. Please refer to the aforementioned section to have an explanation on how to interpret the rejection curves plots.

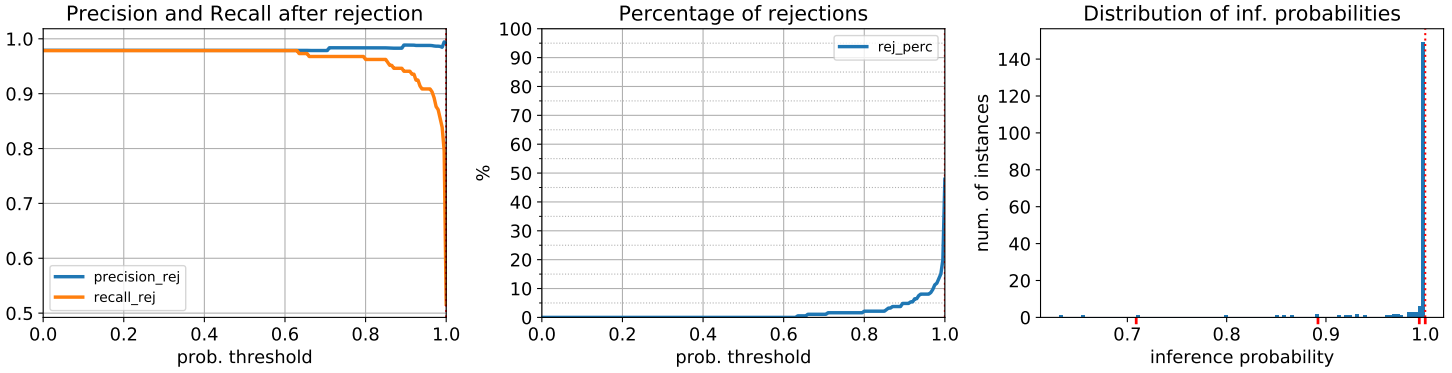
We remind the reader that we wanted to find the threshold corresponding to a rejection rate of 10% on the evaluation set, in order to then apply it to the test set and reject all the inference probabilities below it. We then measured the precision and recall after rejection, as well as the rejection rate on the test set.




*(a) RF*




*(b) RF with NCMs*




*(c) VOXNet 3DCNN*

*Figure 4.10: (a) Rejection curves of the RF model on 3D features without NCMs. (b) Rejection curves of the RF model on 3D features with NCMs. (c) Rejection curves of the VOXNet 3DCNN on 3D OAR geometries 4.10c.*

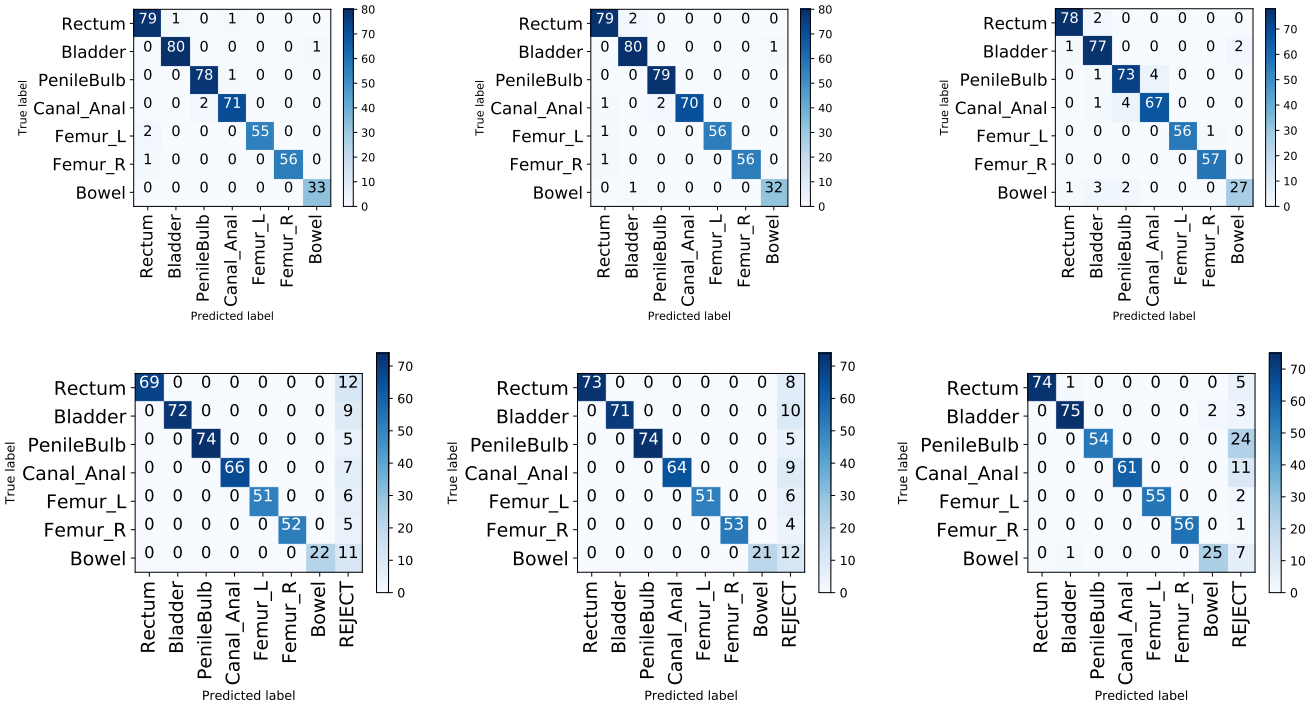
Figure 4.10 shows the rejection curves of the three models employed in the previous section: the RF without and with NCMs, and the VOXNet 3DCNN architecture.[6]

Comparing the RF curves without and with NCMs (figure 4.10a and 4.10b) we can see that in the latter case there were fewer instances with inference probability of 1.[7] Moreover, there was only one misclassification case (as seen in the previous section) which had a very low probability compared to the case without NCMs. On the contrary, the VOXNet architecture showed a very different behavior. Most of the inferences had probability of 1, and between them there were various misclassifications. For this reason, the rejection rate curve was much flatter than the others.

The probability threshold at 10% reject rate on the evaluation set were $0.785$, $.730$, and $.975$, respectively for the RF without NCMs, the RF with NCMs and the VOXNet 3DCNN.

[6] As reported in 3.6, decision trees are not suited for this approach as they mostly return very high inference probabilities.

[7] For the RF to yield a probability of 1, all the trees must agree in the classification of the instance.




*(a) RF* *(b) RF with moments* *(c) VOXNet*

*Figure 4.11: The top row: confusion matrices on the test set before applying the rejection mechanism. The bottom row: confusion matrices on the test set after applying the rejection mechanism; rejected instances are collected in the "REJECT" columns, there is no "REJECT" row because the class does not exist in the ground truth.*

Figure 4.11 shows the effect of the application of the reject threshold (bottom row) on the test set predictions (top row). The first two columns from the left contain the results obtained on the RF without and with NCMs. It is apparent that the rejection policy removed all the misclassifications by assigning them the reject class labels. Thus obtaining a perfect classification on the remaining inferences, which meant a global precision of 1. This happened at the cost of discarding also some correct, although not confident, predictions that could be quantified simply by calculating the reject rate, which for these two models was respectively of $11.93\%$ and $11.71\%$.

The third and rightmost row contains the results of the rejection mechanism applied to the predictions of the VOXNet model. Various differences can be noted compared to the first two approaches. First, there were still misclassifications after applying the rejection policy. Second, the per-class rejection rate was less homogeneous compared to the one of the RF approach: for the Penile_Bulb and Canal_Anal, the two smallest OARs, it was much higher than

the others, and all the other OARs have a lower number of rejections. Third, the global rejection rate was $11.59\%$, slightly lower than the previous two.

*Table 4.2: Classification metrics of RF without NCMs, RF with NCMs, and VOXNet 3DCNN architecture.*

| enconding | model | evaluation set | | | test set - no rej | | | test set - with rejection | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | prec | rec | f1 | prec | rec | f1 | thr | prec | rec | f1 | rej rate |
| 3D | RF | .9847 | .9839 | .9843 | .9807 | .9805 | .9806 | .785 | 1.000 | .8807 | .9365 | .1193 |
| **3D + NCM** | **RF** | **.9948** | **.9946** | **.9947** | **.9808** | **.9805** | **.9806** | **.730** | **1.000** | **.8828** | **.9377** | **.1171** |
| DL | VoxNet | .9790 | .9758 | .9788 | .9521 | .9519 | .9520 | .975 | .9901 | .8752 | .9291 | .1159 |

Table 4.2 contains the classification metrics of the three approaches on the evaluation and test set, without and with rejection mechanism. It is clear that only the RF approaches reached perfect precision after rejection, while the VOXNet architecture had consistently lower metrics on all the sets (excluding rejection rate).

Finally, the resources at hand allowed us to test the two RF approaches on all the avoidance ROIs present in the data set (see section 2.1.2).[8] The rationale behind this choice was to have a rough estimate of the suitability of this method in a scenario where the assumption of having only OARs in the data set was not valid. Ideally, the classifier should reject most (hopefully all) of these not-OARs ROIs. The resulting rejection rate for the RF without and with NCMs was respectively of $84.06\%$ and $86.57\%$.

## 4.6 Rejected Cases

In this section we report some of the most interesting cases of OAR rejection for the RF approach with NCMs.

We first focus on the misclassification cases reported in the confusion matrix in the top row and center column of figure 4.11. We report a 3D rendering of the patient OARs in figure 4.12.

[8] Dealing with extracted feature is much easier than with the 3D encoding of the VOXNet approach, especially when factoring in memory occupation.

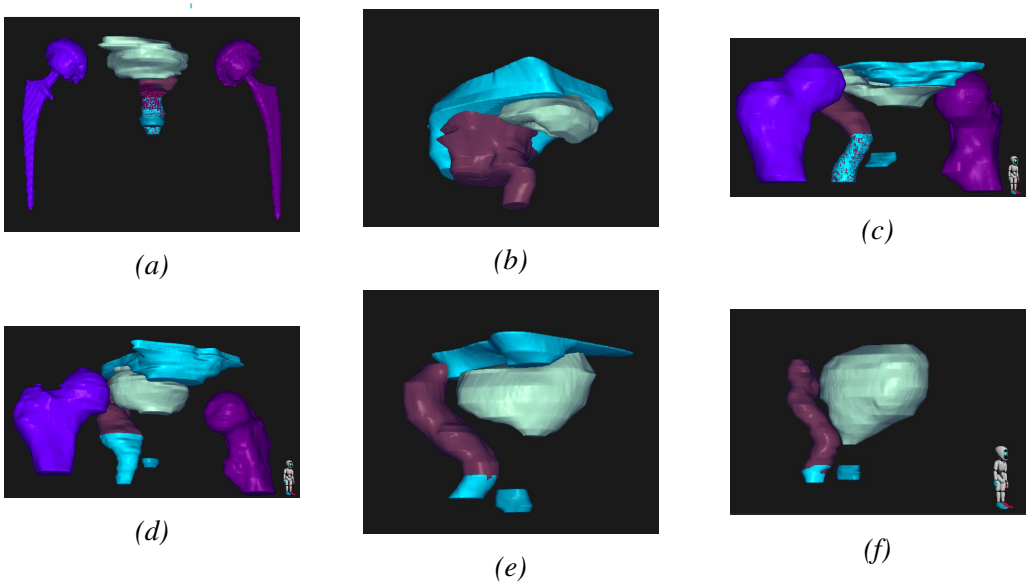
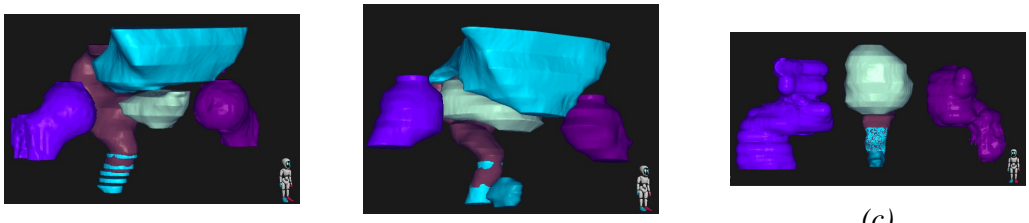


*Figure 4.12: (a) both Femurs as Rectum, inf. prob.* 0.60 *and* 0.58*. (b) Rectum as Bladder, inf. prob.* 0.44*. (c) Bladder as Bowel, inf. prob.* 0.48*. (d) Canal_Anal as Rectum, inf. prob.* 0.40*. (e) Canal_Anal as PenileBulb, inf. prob.* 0.69*. (f) Canal_Anal as PenileBulb, inf. prob.* 0.64*.*

In patient *(a)* both the Femurs were classified as Rectum with an inference probability of around 0.60, but as we can see the Femurs had actually been substituted by two prosthetics. In patient *(b)* the Rectum and the Bowel showed a very abnormal segmentation. In fact, the Rectum should end before (see the other patients). The bladder of patient *(c)* was classified as Bowel and appeared to be deflated. This was interesting because as a diagnostic procedure the patients are usually invited to drink a good amount of liquids before the CT scan is performed[9]. Instead, most of the other patients had more spherical Bladders. Patient *(d)* had an abnormally wide Canal_Anal, and between the six patients above, this was the inference with the lowest probability of 0.40. Finally patient *(e)* and *(f)* had the same misclassification, Canal_Anal classified as a PenileBulb. Initially, there seemed to be no particular reason or anomaly in their Canal_Anal OARs. The interesting fact was that they were the two inferences with the highest probability, signaling that indeed the Canal_Anal and the PenileBulb usually look very similar in most of the patients, but maybe in this case the Canal_Anal should be taller. The common trait of the misclassifications patients *(a),(b),(c),*and *(d)*, was that they were the most abnormal

[9]Water absorbs radiations and stretches the bowel wall thus lowering the chance of damage during treatment.

ones. At the same time, their inference probability was far below the rejection threshold of $0.730$. On the other hand, for patients *(e) and (f)* the probability was close to threshold.



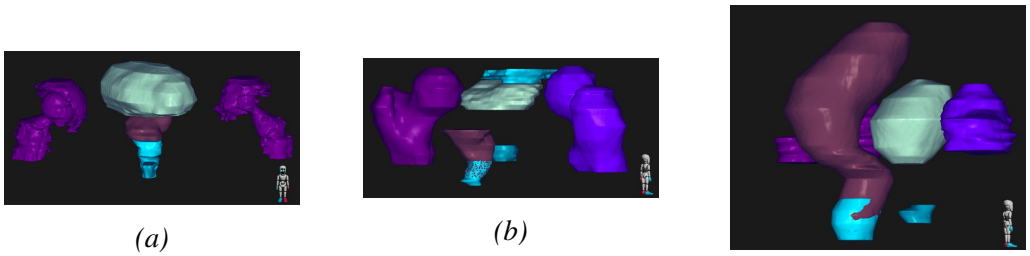
*(a)* *(b)* *(c)*

*Figure 4.13:*
*(a) Rejected: Femur_L* ($0.55$)*, Femur_R* ($0.63$)*, Bowel* ($0.52$)*, Bladder* ($0.69$)*.*
*(b) Rejected: Femur_L* ($0.58$)*, Femur_R* ($0.43$)*, Bowel* ($0.62$)*, PenileBulb* ($0.68$)*.*
*(c) Rejected: Femur_L* ($0.65$)*, Femur_R* ($0.59$)*.*

Figure 4.13 contains the patients with more rejected OARs, regardless of whether the classification was successful or not.[10] In patient *(a)*, the segmentation of the Canal_Anal was peculiar: probably the medical expert forgot to interpolate the slices. However, the Canal_Anal was not rejected[11], but other four OARs were. Although the Femurs were not prosthetics, their segmentation was peculiar because they have hollow internals. Both patients *(a)* and *(b)* had a very large Bowel. In patient *(b)* the Femurs were rather short compared to other contours. Patient *(c)* had two prosthetics Femurs that have both been segmented in a very anomalous *"bumpy"* way.

[10] In this list we should also include patient *(a)* and *(b)* of figure 4.12, but we omitted them to avoid repetition.

[11] Maybe signaling the need for a feature counting the number of connected components in the OAR



*(a)* *(b)* *(c)*

*Figure 4.14:*
*(a) Rejected: Both femurs are prosthetics but in the same OAR as Femur_L* (0.43)*.*
*(b) Rejected: Rectum* (0.44)*.*
*(c) Rejected: Rectum* (0.39)*.*

Finally, the patients in figure 4.14 have one rejected OAR with particularly low inference probabilities. In patient *(a)*, both prosthetic Femurs had been contoured rigorously. The anomalous thing was that they had been included in the same OAR, like if they were belonging to the same organ. For this reason, they appeared to have the same color. This aspect might be the main reason why the inference had such a low probability of $0.43$. In patients *(b)* and *(c)* the Rectums were rejected. In the former, it was rather short; in the latter, it was too inflated. Moreover, although patient *(c)* had two quite peculiar and short Femurs, they were both classified correctly with an inference probability of $0.78$.

## 4.7 Summary

*Table 4.3: Comparative table of all the experiments performed*

| enconding | model | evaluation set | | | test set - no rej [f] | | | test set - with rejection [g] | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | prec | rec | f1 | prec | rec | f1 | thr | prec | rec | f1 | rej rate |
| 2D [a] | DT | .8419 | .8418 | .8418 | - | - | - | - | - | - | - | - |
| | RF | .8800 | .8792 | .8796 | - | - | - | - | - | - | - | - |
| 2D maj [b] | DT | .8573 | .8226 | .8396 | - | - | - | - | - | - | - | - |
| | RF | .9071 | .8817 | .8942 | - | - | - | - | - | - | - | - |
| 3D [c] | DT | .9625 | .9570 | .9597 | .9837 | .9826 | .9832 | - | - | - | - | - |
| | RF | .9847 | .9839 | .9843 | .9807 | .9805 | .9806 | .785 | 1.000 | .8807 | .9365 | .1193 |
| 3D + NCM [d] | DT | .9654 | .9624 | .9639 | .9712 | .9696 | .9704 | - | - | - | - | - |
| | **RF** | **.9948** | **.9946** | **.9947** | **.9808** | **.9805** | **.9806** | **.730** | **1.000** | **.8828** | **.9377** | **.1171** |
| DL [e] | VoxNet | .9790 | .9758 | .9788 | .9521 | .9519 | .9520 | .975 | .9901 | .8752 | .9291 | .1159 |

[a] 2D slice encoding (section 3.4.1).
[b] 2D slice encoding (section 3.4.1) with slice-based majority voting.
[c] 3D engineered features (section 3.4.2) without NCMs.
[d] 3D engineered features (section 3.4.2) with NCMs.
[e] 3D voxelized encoding (section 3.4.3).
[f] Before the application of the reject class (section 3.6).
[g] After the application of the reject class (section 3.6).

In order to summarize all the results obtained and allow for an easier comparison, table 4.3 reports the classification metrics used for all the approaches and experiments outlined in chapter 3.

# Chapter 5

# Discussion

This work was aimed to tackle the problem of OAR label standardization in RT oncology data. We were particularly interested in investigating if the spatial consistency of OAR's contours could be exploited by an ML classification algorithm, and enforce unambiguous labeling. Such a problem had already been investigated in a work by Rozario et al [95] using a DL approach based on an encoding of 2D slices. Although the reported results were sensationally good, multiple aspects were not investigated, leaving gaps open for further studies and discussions.

This work was able to fill those gaps. Our results highlight that even a simple DT model adopted on a similar encoding strategy achieves an f1-score of $0.88$, which is already far better than the one achieved by a trivial random classifier. This signals that simpler models may be employed to solve the problem. In addition, we observe that using a 2D encoding renders very difficult for this type of model to correctly classify OARs in case of overlapping (an instance of an aspect not taken under consideration in [95]).

The feature importance analysis highlights that the spatial context has a major contribution to increasing model performance; this is valid both for 2D and 3D input encodings. In particular, the location of the OAR with respect to the body (and not only in respect to other OARs like in [95]) plays an important role in the classification process.

Our findings also suggest that considering the whole 3D structure of the OAR, either by implementing a slice-based majority voting or by engineering 3D features, has a positive impact on the classification performance. In particular, it significantly helps to reduce the confusion when two OAR overlaps, like in the case of Rectum and Canal_Anal.

Continuing the discussion on 3D engineered features, the results and the

feature importance analysis suggest that NCMs, a set of global shape descriptors, improves the outcome of the classification task. Moreover, an RF approach based on manually engineered 3D features obtains very high classification performance with an f1 score of $0.9947$ on the evaluation set. This result is not matched by an automated feature learning approach (i.e. DL), like the 3DCNN VOXNet architecture, with an f1 score of $0.9788$. However, had we performed a more thorough and comprehensive optimization of the VOXNet hyperparameters, we are positive we would have achieved similar classification performance.

The second aspect touched by this work was evaluating if the inference probability yielded by the ML algorithm could be used as a measure of confidence to discard a not-confident prediction. The rationale behind this research question lies in the fact that we are not interested in the full automation of the standardization process, but in reducing the need for manual check to only peculiar cases (like prosthetic arts or incoherence in the contouring phase), while standardizing most of the remaining labels in an automatic and trustworthy fashion. We decided to answer this question by implementing a reject class mechanism on the test set, based on a probability threshold obtained from the 10% reject rate of the evaluation set[1].

This approach obtained promising results when applied to the test set inferences of the RF model based on 3D engineered features. By rejecting around 12% of the classifications, the remaining 88% of the OARs were classified perfectly. The same cannot be said for the DL approach, which contained misclassifications even after the application of the reject class. DL models are known to yield confident but yet incorrect predictions, which can be exploited in so-called *adversarial attacks*. This aspect is an active area of research and an excellent survey on the matter can be found in [106].

As an unexpected result, the analysis of the rejections of the RF model included cases of extremely peculiar segmentation, prosthetic femurs and multiple OARs in the same contour. This suggests that explicit techniques for outliers and anomaly detection can be successfully employed on OAR geometries. On the other hand, the reject class mechanism had included many correctly classified and apparently normal OAR contours, implying that there is still a wide margin of improvement.

We can conclude that the RF approach based on manually engineered 3D features may be a viable solution for the problem of OAR label standardization based on OAR geometries. Compared to a DL approach it has several advantages: lower computational cost; lower overhead when integrating the

[1] Please see section 3.6 for a justification on this particular approach and reject rate value

solution in existing software (compared to GPU based solutions); higher interpretability of the model; and support for simple but effective reject class mechanism.

The validity of our conclusions is framed by the set of limitations outlined in section 1.8. Tackling these aspects and broadening the impact of this work should be prioritized when planning for future works.

The most pertinent limitation from an implementation point of view is that we restricted our approach only to one data set of a well-defined disease site, coming from a single medical institution. This limitation was originated by the choice of offering a more complete comparison of different methods, namely testing also the 3DCNN approach, as well as performing a robust optimization of the random forest approach. Introducing another data set while comparing all three models would have required efforts exceeding the time framework allocated for this work. For this reason, future works should focus also on other disease sites, most notably the head and neck, abdominal and thorax regions. Moreover, the assumption that all the patients' cases in the data set pertain to the same disease site is excessively restrictive and the viability of a disease-site agnostic model should be investigated with high priority. Nevertheless, the standardization of a single data set is already a relevant result; given that it constitutes an important resource in scientific research projects involving RaySearch Laboratories AB and the Iridium Kankernetwerk clinic.

Another important assumption is that the data set is OAR only, a condition that was achieved with a statistically based divide-and-conquer-like approach outlined in section 3.3. However, there is no guarantee that this approach is viable on all data sets. In future, the research should focus on obtaining a model able to cope with all the type of OARs used in RT oncology (see section 2.1.2), possibly by performing a pre-classification, although it might be particularly difficult in case of avoidance ROIs.

In addition, the ML algorithms used were trained on a data set made of 70 patients. In the future, it would be very interesting to assess if a smaller training set may achieve similar results. This can be done with learning curve estimation, and the results contained in [26] show the viability of this approach with medical imaging data. The end goal could be to obtain a one-shot-learning system or few-shot-learning system [107] capable of standardizing a whole data set from just a handful of very well-curated and cleaned patients' cases that have been manually controlled.

Moreover, the aim of this work was not to find the perfect and best solution to the OAR label standardization problem, but rather to identify a promising path to explore towards and industry-grade robust solution. Many different

approaches and try-outs could have been attempted, if more time and resources were available.

As already reported above, hyperparameter tuning of the VOXNet architecture may give better results. Also, it has been shown in [108] that simultaneously predicting the class label and the object orientation with a slightly modified VOXNet architecture sensibly improves the classification performance. This resembles closely the very interesting approach used in self-supervised learning, where a data set without annotation is used to automatically learn features that can boost the classification task.

Also, using an unsupervised learning approach could bypass the need for a divide-and-conquer strategy. Given that the 3D engineered features used in this work seem to be effective, we could more quickly and more easily attempt a classic clustering method (like K-means), or density based clustering that does not require to know beforehand the number of clusters (i.e. the number of OARs).

As we were interested only to evaluate the feasibility of the approach, the reject class implementation in this work was rather classic and simple as well. Nevertheless, given the promising results obtained, future works can focus on more refined rejection strategies, like the ones based on the distance from the decision boundaries of the model and use per-class local thresholds like in [109] and [110]. Or, more simply, they can focus on how to perform probability calibration of the model inferences to find an optimal rejection threshold [111, 112].

Finally, it could be interesting to experiment with other classifiers like logistic regression; or cast the problem to a series of one-vs-rest binary classification problems, while properly dealing with the inherent class unbalance.

# Appendix A

# Appendix

## A.1 Frequent Item Sets for OAR Names Detection

We attempted an alternative approach to the one explained in section 3.3 to obtain the names of the OARs in the ROI data set.

Given the assumption that OARs are frequent and they frequently appear together, we can consider every patient as a transaction and every ROI name as an item in that transaction. Then, using the apriori algorithm [113] ( implementation offered by [114]), we can mine frequent item sets. We can select the item set with the highest support and the length equal to the expected number of OARs a patient should have[1], which in our case is seven.

In table A.1 we report the item set with the highest support for a given size. The minimum item set support was set to $0.2$. We can clearly see that if we are looking for seven OARs in a patient the resulting ROI names list coincides with the one obtained in section 3.3. Its support is $0.29$ meaning that $29\%$ of the patients contain that item set.

[1] It is a very reasonable assumption that this number is known beforehand.

*Table A.1: Frequent item sets with highest support per set size.*

| support | itemsets | size |
|---|---|---|
| 1.000000 | (rectum) | 1 |
| 1.000000 | (blaas, rectum) | 2 |
| 0.983425 | (bulbus, blaas, rectum) | 3 |
| 0.944751 | (bulbus, blaas, anaal kanaal, rectum) | 4 |
| 0.839779 | (blaas, heup re, rectum, bulbus, heup li) | 5 |
| 0.823204 | (blaas, heup re, rectum, anaal kanaal, bulbus, heup li) | 6 |
| 0.292818 | (blaas, heup re, dunne darm, rectum, anaal kanaal, bulbus, heup li) | 7 |

If we look closely to all the item sets with length equal or higher than seven and we sort then by support (higher first), we can see in table A.2 that the selected item set has almost double the support of the subsequent one, which contains the first avoidance algebraic ROI.

*Table A.2: Frequent item sets of length 7 or 8, sorted by support (only first 9).*

| support | itemsets | size |
|---|---|---|
| 0.292818 | (blaas, heup re, dunne darm, rectum, anaal kanaal, bulbus, heup li) | 7 |
| 0.165746 | (blaas-PTV, blaas, heup re, rectum, anaal kanaal, bulbus, heup li) | 7 |
| 0.160221 | (blaas-PTV, blaas, rectum-PTV, heup re, rectum, bulbus, heup li) | 7 |
| 0.160221 | (blaas, rectum-PTV, heup re, rectum, anaal kanaal, bulbus, heup li) | 7 |
| 0.149171 | (blaas-PTV, blaas, rectum-PTV, heup re, anaal kanaal, bulbus, heup li) | 7 |
| 0.149171 | (blaas-PTV, blaas, rectum-PTV, rectum, anaal kanaal, bulbus, heup li) | 7 |
| 0.149171 | (blaas-PTV, blaas, rectum-PTV, heup re, rectum, anaal kanaal, bulbus) | 7 |
| 0.149171 | (blaas-PTV, blaas, rectum-PTV, heup re, rectum, anaal kanaal, heup li) | 7 |
| 0.149171 | (blaas-PTV, blaas, rectum-PTV, heup re, rectum, anaal kanaal, bulbus, heup li) | 8 |

## A.2 Raw Data Set Additionals

The data set contained $1165$ empty ROIs. For most of them, their origin derives from the fact that some structures where imported from a series of MRI scans and their contour was not replicated on the CT based representation. MRI images are more detailed but at the same time more expensive to acquire, for this reason their are mostly used to detect the GTV [115].

Other ROIs were empty because they were automatically added but never segmented, this typically happens to the lower Bowel (*"dunne darm"* in Dutch) as it is usually far from the PTV and outside the irradiation plane where the beam lays (so not subject to any radiation and not needed for treatment planning). For this reason, the expert will avoid spending precious time in seg-

menting it.

Figure A.1 shows the distribution of the not-empty ROIs per patients. Low outliers were manually checked, the conclusion is that most of the contained ROIs were empty but do not involved OARs.

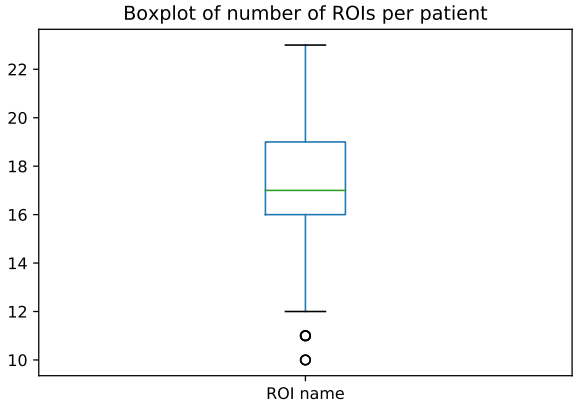



*Figure A.1: Boxplot of number of ROI names per patient*

Figure A.2 shows the average number of slices per OAR. We can clearly see that the Bowel presents the highest relative error. This is probably due to the fact that the CT field of view is reduced only to the proximity of the prostate, or by partial segmentation of the Bowel due to its distance from the treatment site.

## A.3 Target ROI Names Identification

ROI names belonging to targets were identified using the following code python code:

```
import re
pattern = r'(^CTV)|(^PTV)|(^GTV)|(^Dose)|(BODY)'+\
    r'|(^m[a,e]rker[s]?)|(NT)|(^Couch)'
not_targets = []
for i in unique_not_empty_ROI_names:
    is_match=re.match(pattern, i, re.IGNORECASE)
    print("{}{}".format(">>> " if is_match else "",i))
    if is_match is None:
        not_targets.append(i).
```

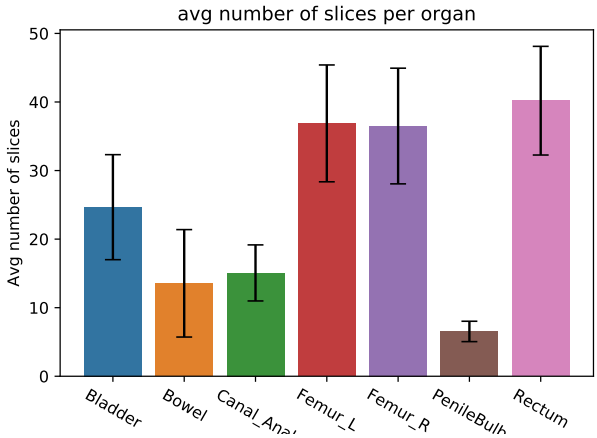



*Figure A.2: Average number of slices per OAR, error bar is the standard deviation.*

## A.4 3D Engineered Features

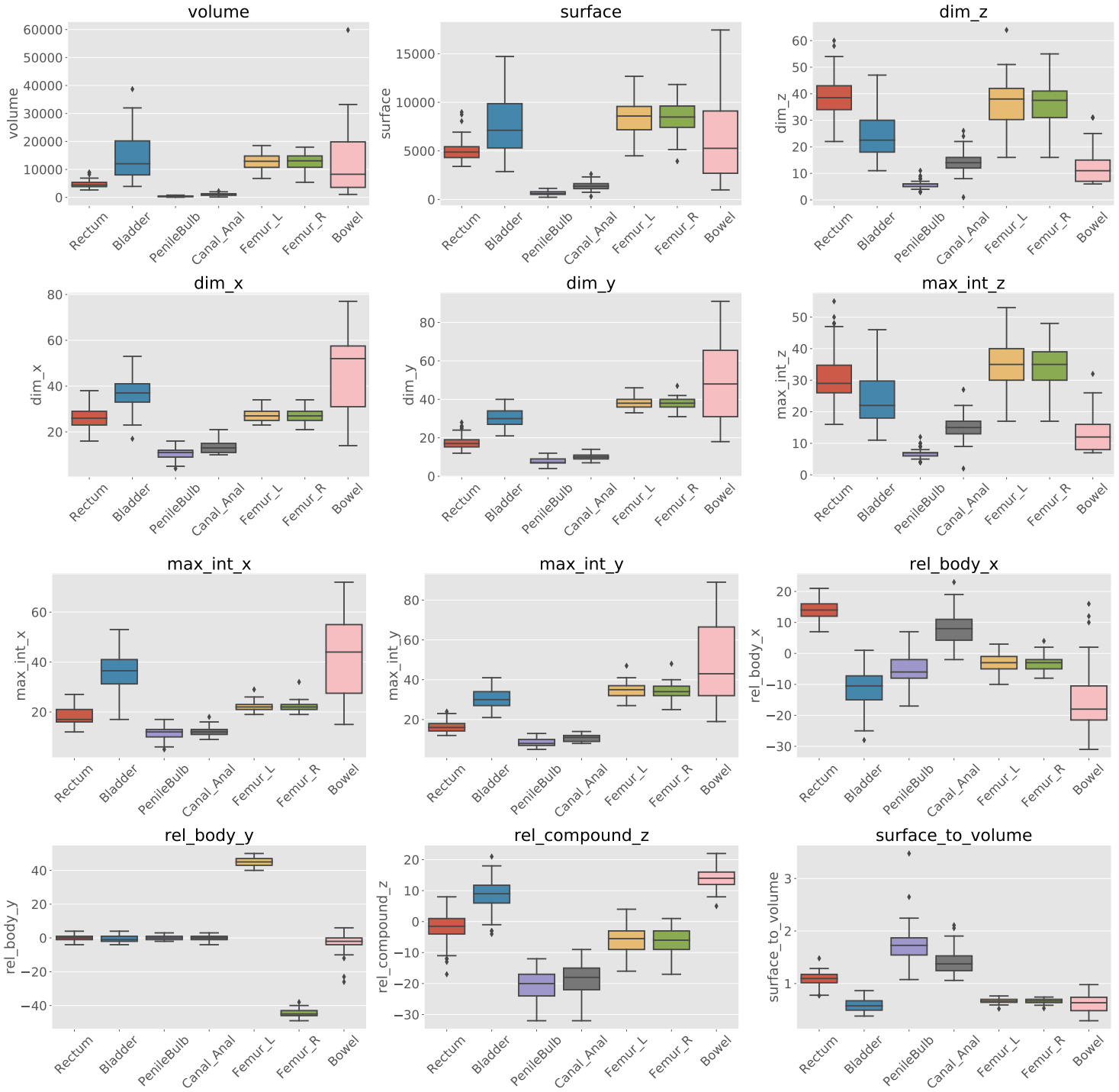



*Figure A.3: Box plots of the engineered feature on the train data set*

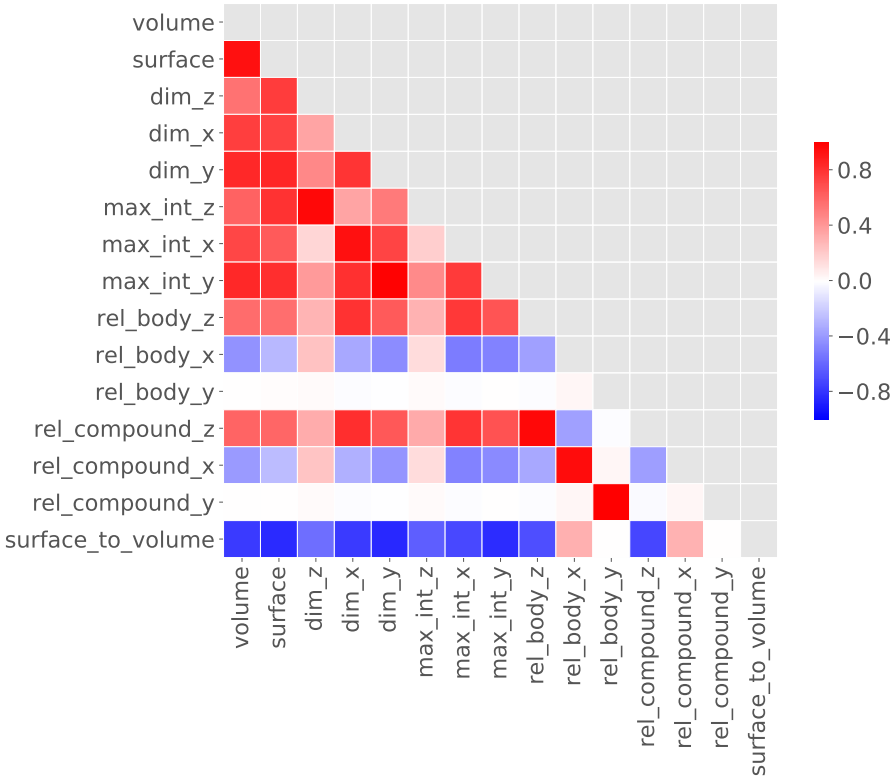



*Figure A.4: Correlation plot of features in train data set*

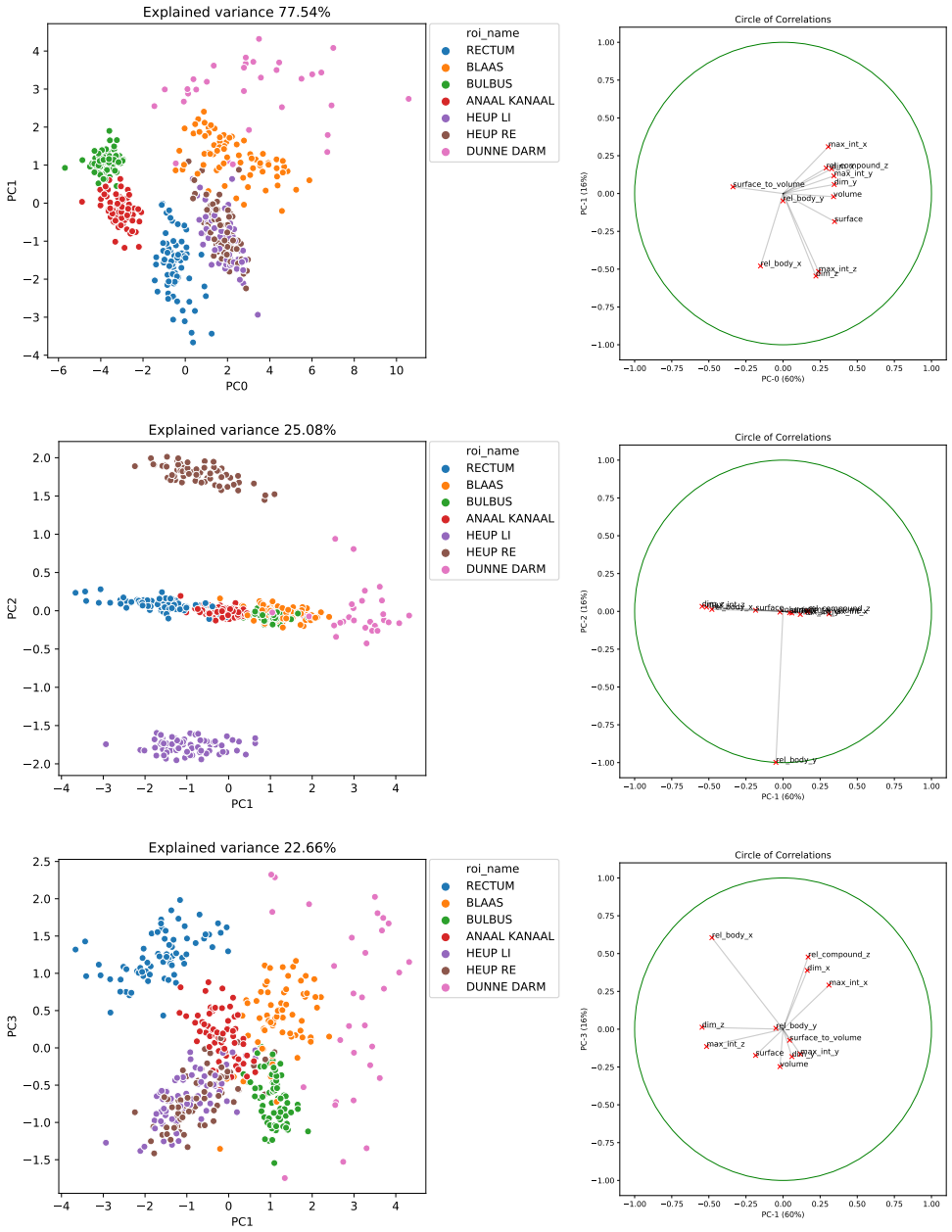



*Figure A.5: left column: projection of the training data on the first three principal components. right column: corresponding correlation circle plot.*

## A.5 Extended Results

| | f1-score | precision | recall | support |
|---|---|---|---|---|
| Rectum | 0.9747 | 1.0000 | 0.9506 | 81 |
| Bladder | 0.9938 | 1.0000 | 0.9877 | 81 |
| PenileBulb | 0.9875 | 0.9753 | 1.0000 | 79 |
| Canal_Anal | 0.9726 | 0.9726 | 0.9726 | 73 |
| Femur_L | 0.9912 | 1.0000 | 0.9825 | 57 |
| Femur_R | 1.0000 | 1.0000 | 1.0000 | 57 |
| Bowel | 0.9429 | 0.8919 | 1.0000 | 33 |
| weighted avg | 0.9832 | 0.9837 | 0.9826 | 461 |

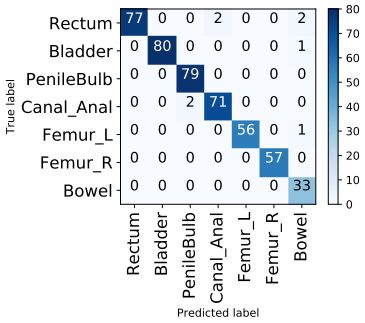



*Figure A.6: 3D engineered features, DT results on the test set.*

| | f1-score | precision | recall | support |
|---|---|---|---|---|
| Rectum | 0.9693 | 0.9634 | 0.9753 | 81 |
| Bladder | 0.9877 | 0.9877 | 0.9877 | 81 |
| PenileBulb | 0.9811 | 0.9750 | 0.9873 | 79 |
| Canal_Anal | 0.9726 | 0.9726 | 0.9726 | 73 |
| Femur_L | 0.9821 | 1.0000 | 0.9649 | 57 |
| Femur_R | 0.9912 | 1.0000 | 0.9825 | 57 |
| Bowel | 0.9851 | 0.9706 | 1.0000 | 33 |
| weighted avg | 0.9806 | 0.9807 | 0.9805 | 461 |

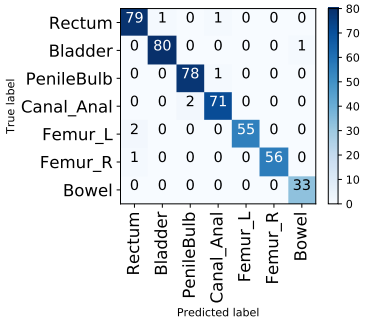



*Figure A.7: 3D engineered features, RF results on the test set.*

| | f1-score | precision | recall | support |
|---|---|---|---|---|
| Rectum | 0.9811 | 1.0000 | 0.9630 | 81 |
| Bladder | 0.9560 | 0.9744 | 0.9383 | 81 |
| PenileBulb | 0.9565 | 0.9390 | 0.9747 | 79 |
| Canal_Anal | 0.9655 | 0.9722 | 0.9589 | 73 |
| Femur_L | 0.9912 | 1.0000 | 0.9825 | 57 |
| Femur_R | 1.0000 | 1.0000 | 1.0000 | 57 |
| Bowel | 0.9296 | 0.8684 | 1.0000 | 33 |
| weighted avg | 0.9704 | 0.9712 | 0.9696 | 461 |

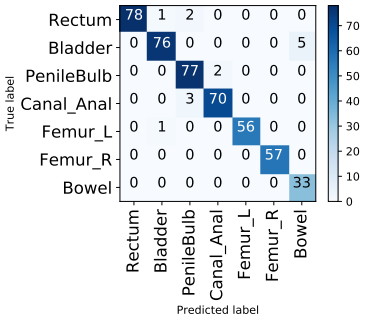



*Figure A.8: 3D engineered features with NCMs, DT results on the test set.*

| | f1-score | precision | recall | support |
|---|---|---|---|---|
| Rectum | 0.9693 | 0.9634 | 0.9753 | 81 |
| Bladder | 0.9756 | 0.9639 | 0.9877 | 81 |
| PenileBulb | 0.9875 | 0.9753 | 1.0000 | 79 |
| Canal_Anal | 0.9790 | 1.0000 | 0.9589 | 73 |
| Femur_L | 0.9912 | 1.0000 | 0.9825 | 57 |
| Femur_R | 0.9912 | 1.0000 | 0.9825 | 57 |
| Bowel | 0.9697 | 0.9697 | 0.9697 | 33 |
| weighted avg | 0.9806 | 0.9808 | 0.9805 | 461 |

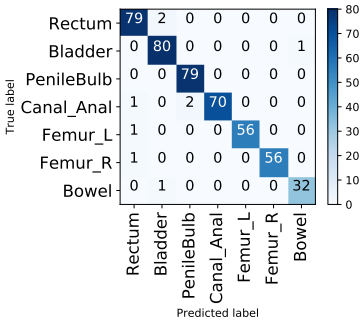



*Figure A.9: 3D engineered features with NCMs, RF results on the test set.*

| | f1-score | precision | recall | support |
|---|---|---|---|---|
| Rectum | 0.9750 | 0.9750 | 0.9750 | 80 |
| Bladder | 0.9390 | 0.9167 | 0.9625 | 80 |
| PenileBulb | 0.9299 | 0.9241 | 0.9359 | 78 |
| Canal_Anal | 0.9371 | 0.9437 | 0.9306 | 72 |
| Femur_L | 0.9912 | 1.0000 | 0.9825 | 57 |
| Femur_R | 0.9913 | 0.9828 | 1.0000 | 57 |
| Bowel | 0.8710 | 0.9310 | 0.8182 | 33 |
| weighted avg | 0.9520 | 0.9521 | 0.9519 | 457 |

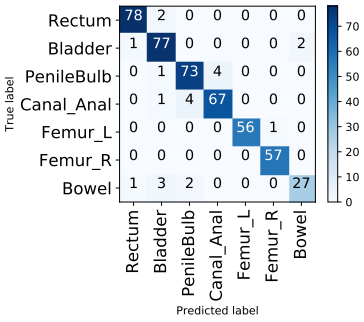



*Figure A.10: 3DCNN, VOXNet results on the test set.*

## A.6 RF Hyper-parameters Tuning

Hyper-parameter tuning was performed using a grid search approach. Moreover, five fold cross validation was used on the train data set. Thus, both the average and standard deviation of the classification metrics were collected.

The final combination of hyperparameters had both the best metrics performance, as well as the lowest standard deviation.

The following tuning parameters were selected according to the literature[116]:

**number of estimators:** total number of estimators. Variance should improve with higher number of estimators, up to a certain tipping point.

**max number of features:** maximum number of features sampled at each node of the tree.

**max depth:** maximum depth of the trees in the forest.

**minimum samples per leaf node:** minimum number of samples in a node to consider it a leaf. Meaning that there could not be a leaf with less samples than this number.

*Table A.3: Parameters used for grid search.*

| parameter | values |
|---|---|
| n_estimators | 4, 8, 16, 32, 64, 128, 256, 512 |
| max_features | 3, 4, 5, 6 |
| max_depth | 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, not limited |
| min_samples_leaf | 1, 2, 4, 8 |

The results are reported in figure A.11

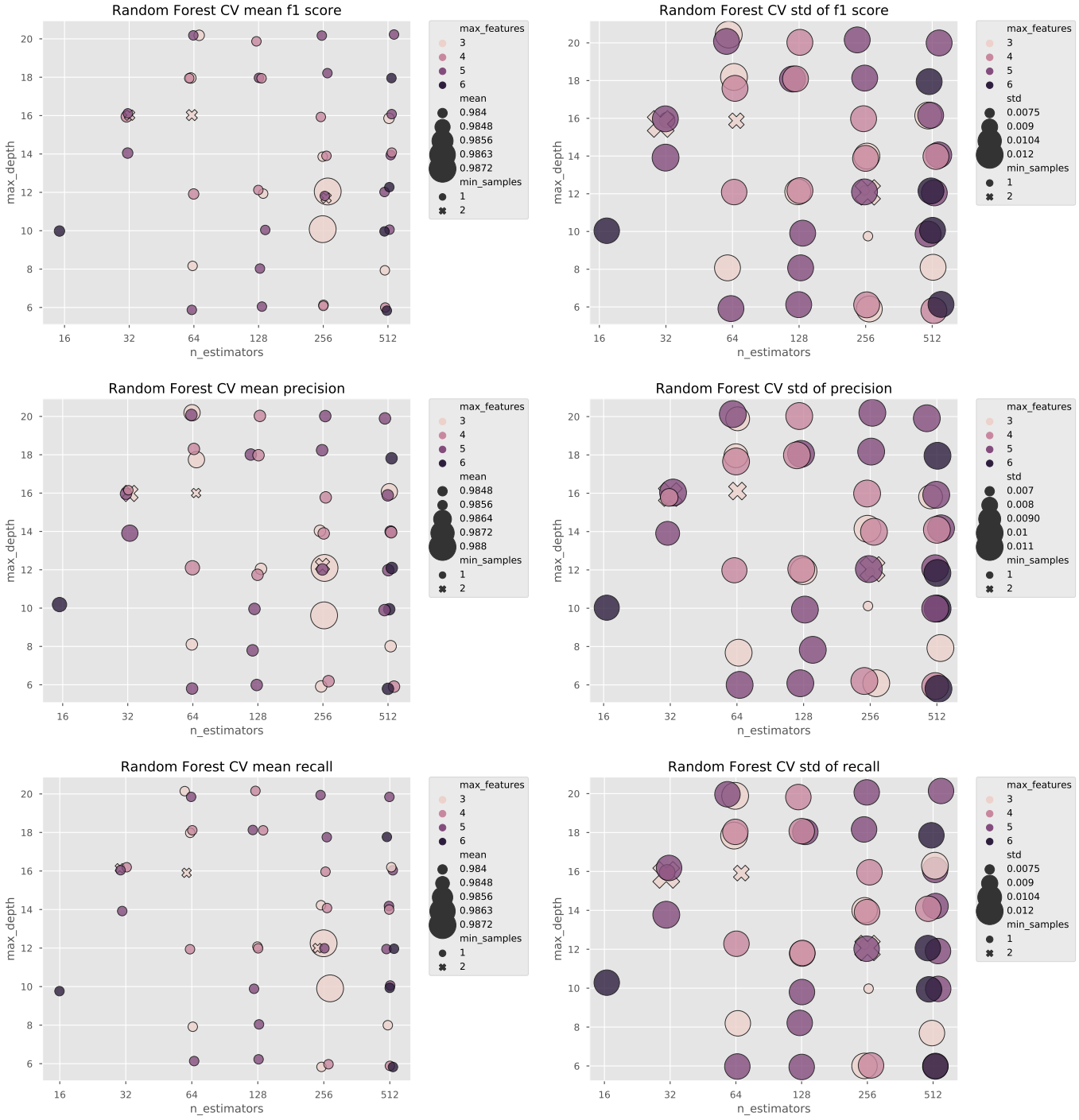



*Figure A.11: 5 fold cross validation results, top 50 results per metric. Left column: average metric result, bigger is better. Right column: metric standard deviation, smaller is better. First row F1 score. Second row precision. Third row recall. Points have been jittered to allow for better interpretation of the results. We can also see that most of top 50 results share the min_samples_leaf value at 1, which means that the three is fully developed until maximum depth.*

which means that the three is fully developed until maximum depth.

## A.7 Experimental Setup and Running Times

Python libraries

```
absl-py==0.7.0
astor==0.7.1
backcall==0.1.0
bleach==3.1.0
cloudpickle==0.7.0
colorama==0.4.1
cycler==0.10.0
Cython==0.29.4
dask==1.1.1
decorator==4.3.2
defusedxml==0.5.0
entrypoints==0.3
gast==0.2.2
graphviz==0.10.1
grpcio==1.18.0
h5py==2.9.0
ipydatawidgets==4.0.0
ipykernel==5.1.0
ipython==7.2.0
ipython-genutils==0.2.0
ipywidgets==7.4.2
jedi==0.13.2
Jinja2==2.10
jsonschema==2.6.0
jupyter-client==5.2.4
jupyter-core==4.4.0
jupyterlab==0.35.4
jupyterlab-server==0.2.0
K3D==2.5.5
Keras==2.2.4
Keras-Applications==1.0.7
Keras-Preprocessing==1.0.9
kiwisolver==1.0.1
mahotas==1.4.5
Markdown==3.0.1
MarkupSafe==1.1.0
matplotlib==3.0.2
mistune==0.8.4
nbconvert==5.4.0
nbformat==4.4.0
networkx==2.2
notebook==5.7.4
numpy==1.16.1
pandas==0.24.1
pandoc==1.0.2
pandocfilters==1.4.2
parso==0.3.3
pickleshare==0.7.5
Pillow==5.4.1
ply==3.11
prometheus-client==0.5.0
prompt-toolkit==2.0.8
protobuf==3.6.1
pydotplus==2.0.2
Pygments==2.3.1
pyparsing==2.3.1
python-dateutil==2.8.0
pythonnet==2.3.0
pytz==2018.9
PyWavelets==1.0.1
pywinpty==0.5.5
PyYAML==5.1
pyzmq==17.1.2
raylearner==1.0
scikit-image==0.14.2
scikit-learn==0.20.2
scipy==1.2.0
seaborn==0.9.0
Send2Trash==1.5.0
```

```
six==1.12.0
sklearn==0.0
tensorboard==1.12.2
tensorflow==1.12.0
tensorflow_gpu==1.12.0
termcolor==1.1.0
terminado==0.8.1
testpath==0.4.2
toolz==0.9.0
tornado==5.1.1
tqdm==4.31.1
traitlets==4.3.2
traittypes==0.2.1
wcwidth==0.1.7
webencodings==0.5.1
Werkzeug==0.14.1
widgetsnbextension==3.4.2
wordcloud==1.5.0
xlrd==1.2.0
```

Workstation hardware (training of DT and RF models):

```
OS Name Microsoft Windows 10 Pro
Version 10.0.17763 Build 17763
System Manufacturer Exertis_CapTech
System Model MS-7A93
System Type x64-based PC
Processor Intel(R) Core(TM) i9-7940X CPU @ 3.10GHz, 3096 Mhz, 14 Core(s), 28 Logical Processor(s)
BaseBoard Manufacturer Micro-Star International Co., Ltd.
BaseBoard Product X299 SLI PLUS (MS-7A93)
BaseBoard Version 1.0
Hardware Abstraction Layer Version = "10.0.17763.503"
Installed Physical Memory (RAM) 64.0 GB
Total Physical Memory 63.7 GB
Available Physical Memory 41.6 GB
Total Virtual Memory 73.2 GB
Available Virtual Memory 23.9 GB
Page File Space 9.50 GB
Hyper-V - VM Monitor Mode Extensions Yes
Hyper-V - Second Level Address Translation Extensions Yes
Hyper-V - Virtualization Enabled in Firmware Yes
Hyper-V - Data Execution Protection Yes
CUDA version    cuda-9.0
```

GPU server hardware (training of VOXNet model):

```
Processor       : Intel(R) Xeon(R) Gold 6140 CPU @ 2.30GHz
cores           : 72
Distributor ID: Ubuntu
Description:    Ubuntu 18.04.2 LTS
Release:        18.04
Codename:       bionic
GPU1: NVIDIA Corporation GV100GL [Tesla V100 SXM2 32GB] (rev a1)
GPU2: NVIDIA Corporation GV100GL [Tesla V100 SXM2 32GB] (rev a1)
GPU3: NVIDIA Corporation GV100GL [Tesla V100 SXM2 32GB] (rev a1)
GPU4: NVIDIA Corporation GV100GL [Tesla V100 SXM2 32GB] (rev a1)
CUDA version: cuda-9.0
```

VOXNet training time: 5 minutes 36 seconds per epoch, 10 total epochs. VOXNet evaluation time for eval data set: 1 minute 42 seconds. VOXNet evaluation time for test data set: 4 minutes 42 seconds.

Random forest cross validation: 7040 fits performed in 2 minutes 40 seconds, parallelism of `n_jobs=12`

Training time for single RF model with best parameters from cross validation: 162 ms ± 1.16 ms per loop (mean ± std. dev. of 7 runs, 10 loops each)

## A.8 VOXNet Training Process

Optimizer used was Adam with an initial $\alpha = 0.0001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$. The batch size was composed of 8 instances, requiring 57 batches to complete a training epoch. The training instances were shuffled randomly at the beginning of each epoch.

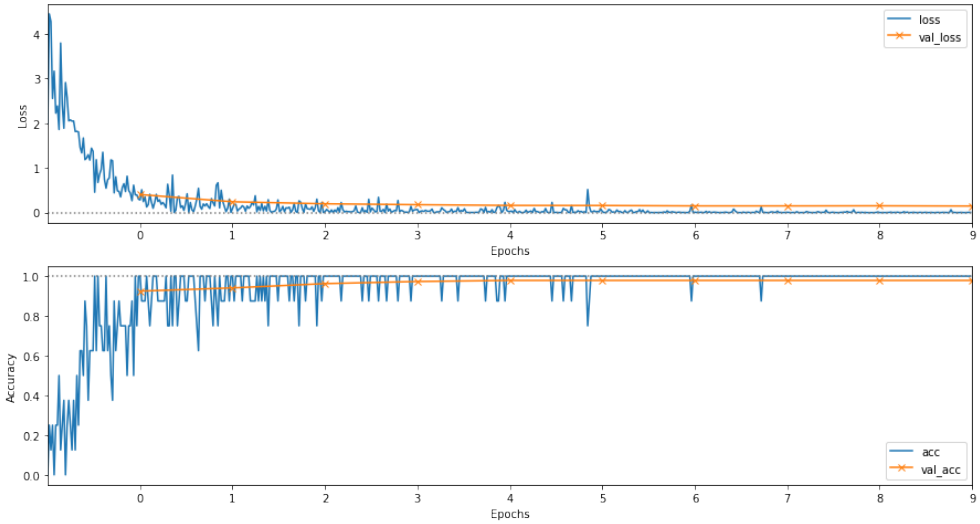



*Figure A.12: Top row loss, bottom row accuracy. Blue training, orange validation. Training metrics computed per each batch. Evaluation metrics computed on the whole evaluation data set.*

The training metrics in figure A.12 (blue lines) are computed per each batch. For this reason, they may be occasionally higher than the evaluation result (for particularly easy batches).

# Bibliography


[1] Fei Jiang et al. "Artificial intelligence in healthcare: past, present and future". In: *Stroke and Vascular Neurology* 2.4 (Dec. 1, 2017), pp. 230–243. ISSN: 2059-8688, 2059-8696. DOI: 10.1136/svn-2017-000101. URL: https://svn.bmj.com/content/2/4/230 (visited on 03/25/2019).

[2] Travis B. Murdoch and Allan S. Detsky. "The Inevitable Application of Big Data to Health Care". In: *JAMA* 309.13 (Apr. 3, 2013), pp. 1351–1352. ISSN: 0098-7484. DOI: 10.1001/jama.2013.393. URL: https://jamanetwork.com/journals/jama/fullarticle/1674245 (visited on 03/25/2019).

[3] Evelyne Kolker, Vural Özdemir, and Eugene Kolker. "How Healthcare Can Refocus on Its Super-Customers and Customers by Leveraging Lessons from Amazon, Uber, and Watson". In: *OMICS: A Journal of Integrative Biology* 20.6 (June 1, 2016), pp. 329–333. DOI: 10.1089/omi.2016.0077. URL: https://www.liebertpub.com/doi/10.1089/omi.2016.0077 (visited on 03/25/2019).

[4] Vimla L. Patel et al. "The coming of age of artificial intelligence in medicine". In: *Artificial Intelligence in Medicine*. Artificial Intelligence in Medicine AIME' 07 46.1 (May 1, 2009), pp. 5–17. ISSN: 0933-3657. DOI: 10.1016/j.artmed.2008.07.017. URL: http://www.sciencedirect.com/science/article/pii/S0933365708000961 (visited on 03/25/2019).

[5] D. B. Neill. "Using Artificial Intelligence to Improve Hospital Inpatient Care". In: *IEEE Intelligent Systems* 28.2 (Mar. 2013), pp. 92–95. ISSN: 1541-1672. DOI: 10.1109/MIS.2013.51.

[6] Insights Team. *AI And Healthcare: A Giant Opportunity*. Forbes. Feb. 11, 2019. URL: https://www.forbes.com/sites/insights-

intelai/2019/02/11/ai-and-healthcare-a-giant-opportunity/ (visited on 03/25/2019).

[7] Mark D. Lloyd. *Artificial intelligence: The next revolution in healthcare?* Medtech Views. June 6, 2018. URL: http://www.medtechviews.eu/article/artificial-intelligence-next-revolution-healthcare (visited on 03/25/2019).

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira et al. Curran Associates, Inc., 2012, pp. 1097–1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[9] C. Farabet et al. "Learning Hierarchical Features for Scene Labeling". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1915–1929. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2012.231.

[10] Jonathan J Tompson et al. "Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation". In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani et al. Curran Associates, Inc., 2014, pp. 1799–1807. URL: http://papers.nips.cc/paper/5573-joint-training-of-a-convolutional-network-and-a-graphical-model-for-human-pose-estimation.pdf.

[11] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *Nature* 521.7553 (May 2015), pp. 436–444. ISSN: 1476-4687. DOI: 10.1038/nature14539. URL: https://www.nature.com/articles/nature14539 (visited on 03/06/2019).

[12] Geert Litjens et al. "A survey on deep learning in medical image analysis". In: *Medical Image Analysis* 42 (Dec. 1, 2017), pp. 60–88. ISSN: 1361-8415. DOI: 10.1016/j.media.2017.07.005. URL: http://www.sciencedirect.com/science/article/pii/S1361841517301135 (visited on 03/04/2019).

[13] Bradley J. Erickson et al. "Machine Learning for Medical Imaging". In: *RadioGraphics* 37.2 (Feb. 17, 2017), pp. 505–515. ISSN: 0271-5333. DOI: 10.1148/rg.2017160130. URL: https://pubs.rsna.org/doi/full/10.1148/rg.2017160130 (visited on 03/04/2019).

[14] Filippo Amato et al. “Artificial neural networks in medical diagnosis”. In: *Journal of Applied Biomedicine* 11.2 (Jan. 1, 2013), pp. 47–58. ISSN: 1214-021X. DOI: 10.2478/v10136-012-0031-x. URL: http://www.sciencedirect.com/science/article/pii/S1214021X14600570 (visited on 03/04/2019).

[15] Erik R. Ranschaert, Sergey Morozov, and PaulR Algra. “Artificial Intelligence in Medical Imaging”. In: *Springer International Publishing*. 2019. DOI: 10.1007/978-3-319-94878-2.

[16] M. N. Wernick et al. “Machine Learning in Medical Imaging”. In: *IEEE Signal Processing Magazine* 27.4 (July 2010), pp. 25–38. ISSN: 1053-5888. DOI: 10.1109/MSP.2010.936730.

[17] H. Greenspan, B. van Ginneken, and R. M. Summers. “Guest Editorial Deep Learning in Medical Imaging: Overview and Future Promise of an Exciting New Technique”. In: *IEEE Transactions on Medical Imaging* 35.5 (May 2016), pp. 1153–1159. ISSN: 0278-0062. DOI: 10.1109/TMI.2016.2553401.

[18] An Tang et al. “Canadian Association of Radiologists White Paper on Artificial Intelligence in Radiology”. In: *Canadian Association of Radiologists Journal* 69.2 (May 1, 2018), pp. 120–135. ISSN: 0846-5371. DOI: 10.1016/j.carj.2018.02.002. URL: http://www.sciencedirect.com/science/article/pii/S0846537118300305 (visited on 02/19/2019).

[19] Dinggang Shen, Guorong Wu, and Heung-Il Suk. “Deep Learning in Medical Image Analysis”. In: *Annual Review of Biomedical Engineering* 19.1 (2017), pp. 221–248. DOI: 10.1146/annurev-bioeng-071516-044442. URL: https://doi.org/10.1146/annurev-bioeng-071516-044442 (visited on 03/11/2019).

[20] Krzysztof J. Cios and G. William Moore. “Uniqueness of medical data mining”. In: *Artificial Intelligence in Medicine*. Medical Data Mining and Knowledge Discovery 26.1 (Sept. 1, 2002), pp. 1–24. ISSN: 0933-3657. DOI: 10.1016/S0933-3657(02)00049-0. URL: http://www.sciencedirect.com/science/article/pii/S0933365702000490 (visited on 03/04/2019).

[21] Marc D. Kohli, Ronald M. Summers, and J. Raymond Geis. “Medical Image Data and Datasets in the Era of Machine Learning—Whitepaper from the 2016 C-MIMI Meeting Dataset Session”. In: *Journal of Digital Imaging* 30.4 (Aug. 1, 2017), pp. 392–399. ISSN: 1618-727X. DOI:

10.1007/s10278-017-9976-3. URL: https://doi.org/10.1007/s10278-017-9976-3 (visited on 01/28/2019).

[22] Beau Norgeot, Benjamin S. Glicksberg, and Atul J. Butte. "A call for deep-learning healthcare". In: *Nature Medicine* 25.1 (Jan. 2019), p. 14. ISSN: 1546-170X. DOI: 10.1038/s41591-018-0320-3. URL: http://www.nature.com/articles/s41591-018-0320-3 (visited on 03/25/2019).

[23] J. Deng et al. "ImageNet: A Large-Scale Hierarchical Image Database". In: *CVPR09*. 2009.

[24] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. "CIFAR-10 (Canadian Institute for Advanced Research)". In: (). URL: http://www.cs.toronto.edu/~kriz/cifar.html.

[25] Tsung-Yi Lin et al. "Microsoft COCO: Common Objects in Context". In: *Computer Vision – ECCV 2014*. Ed. by David Fleet et al. Lecture Notes in Computer Science. Springer International Publishing, 2014, pp. 740–755. ISBN: 978-3-319-10602-1.

[26] Junghwan Cho et al. "How much data is needed to train a medical image deep learning system to achieve necessary high accuracy?" In: *arXiv:1511.06348 [cs]* (Nov. 19, 2015). arXiv: 1511.06348. URL: http://arxiv.org/abs/1511.06348 (visited on 03/11/2019).

[27] Peter Mildenberger, Marco Eichelberg, and Eric Martin. "Introduction to the DICOM standard". In: *European Radiology* 12.4 (Apr. 1, 2002), pp. 920–927. ISSN: 1432-1084. DOI: 10.1007/s003300101100. URL: https://doi.org/10.1007/s003300101100 (visited on 05/15/2019).

[28] *DICOM Standard*. URL: https://www.dicomstandard.org/ (visited on 03/07/2019).

[29] Mark Oliver Gueld et al. "Quality of DICOM header information for image categorization". In: *Medical Imaging 2002: PACS and Integrated Medical Information Systems: Design and Evaluation*. Medical Imaging 2002: PACS and Integrated Medical Information Systems: Design and Evaluation. Vol. 4685. International Society for Optics and Photonics, May 16, 2002, pp. 280–288. DOI: 10.1117/12.467017. URL: https://www-spiedigitallibrary-org.focus.lib.kth.se/conference-proceedings-of-spie/4685/0000/Quality-of-DICOM-header-

information-for-image-categorization/10.1117/12.467017.short (visited on 03/04/2019).

[30] Choong Ho Lee and Hyung-Jin Yoon. "Medical big data: promise and challenges". In: *Kidney Research and Clinical Practice* 36.1 (Mar. 2017), pp. 3–11. ISSN: 2211-9132. DOI: 10.23876/j.krcp.2017.36.1.3. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5331970/ (visited on 03/12/2019).

[31] Rajamanickam Baskar et al. "Cancer and Radiation Therapy: Current Advances and Future Directions". In: *International Journal of Medical Sciences* 9.3 (Feb. 27, 2012), pp. 193–199. ISSN: 1449-1907. DOI: 10.7150/ijms.3635. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3298009/ (visited on 03/12/2019).

[32] Juliette Thariat et al. "Past, present, and future of radiotherapy for the benefit of patients". In: *Nature Reviews Clinical Oncology* 10.1 (Jan. 2013), pp. 52–60. ISSN: 1759-4782. DOI: 10.1038/nrclinonc.2012.203. URL: http://www.nature.com/articles/nrclinonc.2012.203 (visited on 03/11/2019).

[33] Toshiyuki Okada et al. "Construction of Hierarchical Multi-Organ Statistical Atlases and Their Application to Multi-Organ Segmentation from CT Images". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2008*. Ed. by Dimitris Metaxas et al. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008, pp. 502–509. ISBN: 978-3-540-85988-8.

[34] Marius George Linguraru et al. "Statistical 4D graphs for multi-organ abdominal segmentation from multiphase CT". In: *Medical Image Analysis* 16.4 (May 2012), pp. 904–914. ISSN: 1361-8423. DOI: 10.1016/j.media.2012.02.001.

[35] Marius George Linguraru et al. "Automated segmentation and quantification of liver and spleen from CT images using normalized probabilistic atlases and enhancement estimation". In: *Medical Physics* 37.2 (Feb. 2010), pp. 771–783. ISSN: 0094-2405. DOI: 10.1118/1.3284530.

[36] Y. Zhou and J. Bai. "Multiple Abdominal Organ Segmentation: An Atlas-Based Fuzzy Connectedness Approach". In: *IEEE Transactions on Information Technology in Biomedicine* 11.3 (May 2007), pp. 348–352. ISSN: 1089-7771. DOI: 10.1109/TITB.2007.892695.

[37] Masaharu Kobashi and Linda G. Shapiro. “Knowledge-based organ identification from CT images”. In: *Pattern Recognition* 28.4 (Apr. 1, 1995), pp. 475–491. ISSN: 0031-3203. DOI: 10.1016/0031-3203(94)00124-5. URL: http://www.sciencedirect.com/science/article/pii/0031320394001245 (visited on 05/01/2019).

[38] Soumya Ghose et al. “A review of segmentation and deformable registration methods applied to adaptive cervical cancer radiation therapy treatment planning”. In: *Artificial Intelligence in Medicine* 64.2 (June 1, 2015), pp. 75–87. ISSN: 0933-3657. DOI: 10.1016/j.artmed.2015.04.006. URL: http://www.sciencedirect.com/science/article/pii/S0933365715000469 (visited on 03/11/2019).

[39] E. Gibson et al. “Automatic Multi-Organ Segmentation on Abdominal CT With Dense V-Networks”. In: *IEEE Transactions on Medical Imaging* 37.8 (Aug. 2018), pp. 1822–1834. ISSN: 0278-0062. DOI: 10.1109/TMI.2018.2806309.

[40] Walter R. Bosch et al. *Head-Neck Cetuximab (RTOG 0522 and ACRIN 4500) - Imaging - Cancer Imaging Program - National Cancer Institute - Confluence Wiki*. URL: https://wiki.nci.nih.gov/display/CIP/Head-Neck+Cetuximab+(RTOG+0522+and+ACRIN+4500) (visited on 02/06/2019).

[41] C. Mayo et al. “AAPM Task Group 263: Tackling Standardization of Nomenclature for Radiation Therapy”. In: *International Journal of Radiation Oncology*Biology*Physics*. Proceedings of the American Society for Radiation Oncology 57th Annual Meeting 93.3 (Nov. 1, 2015), E383–E384. ISSN: 0360-3016. DOI: 10.1016/j.ijrobp.2015.07.1525. URL: http://www.sciencedirect.com/science/article/pii/S0360301615022567 (visited on 03/19/2019).

[42] Travis R. Denton et al. “Guidelines for treatment naming in radiation oncology”. In: *Journal of Applied Clinical Medical Physics* 17.2 (Nov. 7, 2015), pp. 123–138. ISSN: 1526-9914. DOI: 10.1120/jacmp.v17i2.5953. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5874902/ (visited on 03/19/2019).

[43] Erhard Rahm and Hong Hai Do. “Data Cleaning: Problems and Current Approaches”. In: *IEEE Data Eng. Bull.* 23.4 (2000), pp. 3–13.

URL: http://sites.computer.org/debull/A00DEC-CD.pdf.

[44] *Inductive Approach (Inductive Reasoning)*. Research-Methodology. URL: https://research-methodology.net/research-methodology/research-approach/inductive-approach-2/ (visited on 05/26/2019).

[45] *Occam's razor, Encyclopedia Britannica*. Encyclopedia Britannica. URL: https://www.britannica.com/topic/Occams-razor (visited on 05/29/2019).

[46] Lakshmi Santanam et al. "Standardizing naming conventions in radiation oncology". In: *International Journal of Radiation Oncology, Biology, Physics* 83.4 (July 15, 2012), pp. 1344–1349. ISSN: 1879-355X. DOI: 10.1016/j.ijrobp.2011.09.054.

[47] Elizabeth L. Covington et al. "Improving treatment plan evaluation with automation". In: *Journal of Applied Clinical Medical Physics* 17.6 (2016), pp. 16–31. ISSN: 1526-9914. DOI: 10.1120/jacmp.v17i6.6322.

[48] Charles Mayo. *AAPM Reports - Standarizing Nomenclatures in Radiation Oncology*. 2018. URL: https://www.aapm.org/pubs/reports/detail.asp?docid=171 (visited on 03/19/2019).

[49] Mark L. Graber, Nancy Franklin, and Ruthanna Gordon. "Diagnostic Error in Internal Medicine". In: *Archives of Internal Medicine* 165.13 (July 11, 2005), pp. 1493–1499. ISSN: 0003-9926. DOI: 10.1001/archinte.165.13.1493. URL: https://jamanetwork.com/journals/jamainternalmedicine/fullarticle/486642 (visited on 03/25/2019).

[50] Lakshmi Santanam et al. "Eliminating inconsistencies in simulation and treatment planning orders in radiation therapy". In: *International Journal of Radiation Oncology, Biology, Physics* 85.2 (Feb. 1, 2013), pp. 484–491. ISSN: 1879-355X. DOI: 10.1016/j.ijrobp.2012.03.023.

[51] Kevin L. Moore et al. "Experience-based quality control of clinical intensity-modulated radiotherapy planning". In: *International Journal of Radiation Oncology, Biology, Physics* 81.2 (Oct. 1, 2011), pp. 545–551. ISSN: 1879-355X. DOI: 10.1016/j.ijrobp.2010.11.030.

[52] Binbin Wu et al. “Patient geometry-driven information retrieval for IMRT treatment plan quality control”. In: *Medical Physics* 36.12 (Dec. 2009), pp. 5497–5505. ISSN: 0094-2405. DOI: 10.1118/1.3253464.

[53] James A. Purdy. “Quality assurance issues in conducting multi-institutional advanced technology clinical trials”. In: *International Journal of Radiation Oncology, Biology, Physics* 71.1 (2008), S66–70. ISSN: 0360-3016. DOI: 10.1016/j.ijrobp.2007.07.2393.

[54] H. R. Roth et al. “Anatomy-specific classification of medical images using deep convolutional nets”. In: *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*. 2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI). Apr. 2015, pp. 101–104. DOI: 10.1109/ISBI.2015.7163826.

[55] Suzanne C. Beyea. “Ensuring correct site surgery”. In: *AORN journal* 76.5 (Nov. 2002), pp. 880–882. ISSN: 0001-2092.

[56] Mark Bernstein. “Wrong-side surgery: systems for prevention”. In: *Canadian Journal of Surgery. Journal Canadien De Chirurgie* 46.2 (Apr. 2003), pp. 144–146. ISSN: 0008-428X.

[57] James D. Christensen, Gary C. Hutchins, and Clement J. McDonald. “Computer Automated Detection of Head Orientation for Prevention of Wrong-Side Treatment Errors”. In: *AMIA Annual Symposium Proceedings* 2006 (2006), pp. 136–140. ISSN: 1942-597X. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1839503/ (visited on 05/27/2019).

[58] *X-ray*. In: *Wikipedia*. Page Version ID: 897125334. May 14, 2019. URL: https://en.wikipedia.org/w/index.php?title=X-ray&oldid=897125334 (visited on 05/16/2019).

[59] *Cone_Beam_CT_principle.png (1083×885)*. URL: https://upload.wikimedia.org/wikipedia/commons/b/bb/Cone_Beam_CT_principle.png (visited on 05/16/2019).

[60] Gabor T. Herman. *Fundamentals of Computerized Tomography: Image Reconstruction from Projections*. 2nd ed. Advances in Computer Vision and Pattern Recognition. London: Springer-Verlag, 2009. ISBN: 978-1-85233-617-2. URL: http://www.springer.com/gp/book/9781852336172 (visited on 05/16/2019).

[61] *File:Ct-internals.jpg - Wikimedia Commons*. URL: https://commons.wikimedia.org/wiki/File:Ct-internals.jpg (visited on 05/16/2019).

[62] No machine-readable author provided Braegel assumed. *Durchführung einer CT gesteuerten Periradikulären Therapie.* Mar. 29, 2007. URL: https://commons.wikimedia.org/wiki/File:Prt_gantryview.jpg (visited on 05/16/2019).

[63] *Voxel*. In: *Wikipedia*. Page Version ID: 893830102. Apr. 23, 2019. URL: https://en.wikipedia.org/w/index.php?title=Voxel&oldid=893830102 (visited on 05/15/2019).

[64] Department of Radiology Häggström Uppsala University Hospital Uploaded by Mikael. *English: Computer tomography of human brain, from base of the skull to top. Taken with intravenous contrast medium.* Jan. 17, 2008. URL: https://commons.wikimedia.org/wiki/File:Computed_tomography_of_human_brain_-_large.png (visited on 05/16/2019).

[65] MindwaysCT Software. *English: Quantitative computed tomograph showing abdominal cross section showing vertebrae bone density and calibration phantom.* Dec. 11, 2012. URL: https://commons.wikimedia.org/wiki/File:Image_of_3D_volumetric_QCT_scan.jpg (visited on 05/16/2019).

[66] BruceBlaus When using this image in external sources it can be cited as:Blausen com staff. *English: Sectional Planes of the Brain. See a related animation of this medical topic.* Feb. 11, 2014. URL: https://commons.wikimedia.org/wiki/File:Blausen_0104_Brain_x-secs_SectionalPlanes.png (visited on 05/16/2019).

[67] *3d interactive data visualisation with K3D-jupyter · OpenDreamKit*. URL: https://opendreamkit.org/2018/10/28/3d/ (visited on 06/07/2019).

[68] Radiological Society of North America (RSNA) {and} American College of Radiology (ACR). *Fiducial Marker Placement*. URL: https://www.radiologyinfo.org/en/info.cfm?pg=fiducial-marker (visited on 05/15/2019).

[69] *ROI Name Attribute – DICOM Standard Browser*. URL: https://dicom.innolitics.com/ciods/rt-structure-set/structure-set/30060020/30060026 (visited on 05/16/2019).

[70] Tom Fawcett. "An introduction to ROC analysis". In: *Pattern Recognition Letters* 27.8 (June 2006), pp. 861–874. ISSN: 01678655. DOI: 10.1016/j.patrec.2005.10.010. URL: https://linkinghub.elsevier.com/retrieve/pii/S016786550500303X (visited on 02/22/2019).

[71] *CT Physics*. URL: https://web2.uwindsor.ca/courses/physics/high_schools/2006/Medical_Imaging/ctphysics.html (visited on 05/16/2019).

[72] *RT ROI Observations Module – DICOM Standard Browser*. URL: https://dicom.innolitics.com/ciods/rt-structure-set/rt-roi-observations (visited on 05/16/2019).

[73] *NCBO BioPortal*. URL: http://bioportal.bioontology.org/ontologies (visited on 05/24/2019).

[74] Natalya F. Noy et al. "Pushing the envelope: challenges in a frame-based representation of human anatomy". In: *Data & Knowledge Engineering* 48.3 (Mar. 1, 2004), pp. 335–359. ISSN: 0169-023X. DOI: 10.1016/j.datak.2003.06.002. URL: http://www.sciencedirect.com/science/article/pii/S0169023X03001253 (visited on 05/24/2019).

[75] *SNOMED - 5-Step Briefing*. SNOMED. URL: http://www.snomed.org/snomed-ct/five-step-briefing (visited on 05/24/2019).

[76] J. J. Kim et al. "A Standardized Nomenclature System for Head and Neck (H&N) IMRT Contouring, Planning and Quality Assurance". In: *International Journal of Radiation Oncology*Biology*Physics*. Proceedings of the American Society for Therapeutic Radiology and Oncology 49th Annual Meeting 69.3 (Nov. 1, 2007), S473. ISSN: 0360-3016. DOI: 10.1016/j.ijrobp.2007.07.1667. URL: http://www.sciencedirect.com/science/article/pii/S0360301607029483 (visited on 05/01/2019).

[77] Jialu Yu et al. "Radiation Therapy Digital Data Submission Process for National Clinical Trials Network". In: *International Journal of Radiation Oncology*Biology*Physics* 90.2 (Oct. 1, 2014), pp. 466–467. ISSN: 0360-3016. DOI: 10.1016/j.ijrobp.2014.05.2672. URL: http://www.sciencedirect.com/science/article/pii/S0360301614033252 (visited on 05/01/2019).

[78] Charles S. Mayo et al. “Establishment of practice standards in nomenclature and prescription to enable construction of software and databases for knowledge-based practice review”. In: *Practical Radiation Oncology* 6.4 (July 1, 2016), e117–e126. ISSN: 1879-8500. DOI: 10.1016/j.prro.2015.11.001. URL: http://www.sciencedirect.com/science/article/pii/S1879850015004002 (visited on 05/01/2019).

[79] TG263 AAPM. *TG-263 Structure Spreadsheet*. URL: https://www.aapm.org/pubs/reports/RPT_263_Supplemental/TG263_Nomenclature_Worksheet_20170815.xls (visited on 05/24/2019).

[80] Dengsheng Zhang and Guojun Lu. “Review of shape representation and description techniques”. In: *Pattern Recognition* 37.1 (Jan. 1, 2004), pp. 1–19. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2003.07.008. URL: http://www.sciencedirect.com/science/article/pii/S0031320303002759 (visited on 04/25/2019).

[81] Y. Guo et al. “3D Object Recognition in Cluttered Scenes with Local Surface Features: A Survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.11 (Nov. 2014), pp. 2270–2287. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2014.2316828.

[82] Ming-Kuei Hu. “Visual pattern recognition by moment invariants”. In: *IRE Transactions on Information Theory* 8.2 (Feb. 1962), pp. 179–187. ISSN: 0096-1000. DOI: 10.1109/TIT.1962.1057692.

[83] Michael Reed Teague. “Image analysis via the general theory of moments*”. In: *JOSA* 70.8 (Aug. 1, 1980), pp. 920–930. DOI: 10.1364/JOSA.70.000920. URL: https://www.osapublishing.org/josa/abstract.cfm?uri=josa-70-8-920 (visited on 06/07/2019).

[84] Dengsheng Zhang and Guojun Lu. “Generic Fourier descriptor for shape-based image retrieval”. In: *Proceedings. IEEE International Conference on Multimedia and Expo*. Proceedings. IEEE International Conference on Multimedia and Expo. Vol. 1. Aug. 2002, 425–428 vol.1. DOI: 10.1109/ICME.2002.1035809.

[85] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012. ISBN: 978-0-262-01825-8.

[86] Leo Breiman. “Random Forests”. In: *Machine Learning* 45.1 (Oct. 1, 2001), pp. 5–32. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324. URL: https://doi.org/10.1023/A:1010933404324 (visited on 05/20/2019).

[87] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[88] Maciej A. Mazurowski et al. “Deep learning in radiology: an overview of the concepts and a survey of the state of the art”. In: *arXiv:1802.08717 [cs, stat]* (Feb. 9, 2018). arXiv: 1802.08717. URL: http://arxiv.org/abs/1802.08717 (visited on 02/14/2019).

[89] D. Maturana and S. Scherer. “VoxNet: A 3D Convolutional Neural Network for real-time object recognition”. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Sept. 2015, pp. 922–928. DOI: 10.1109/IROS.2015.7353481.

[90] Zhirong Wu et al. “3D ShapeNets: A deep representation for volumetric shapes”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). June 2015, pp. 1912–1920. DOI: 10.1109/CVPR.2015.7298801.

[91] John Kang et al. “Machine Learning Approaches for Predicting Radiation Therapy Outcomes: A Clinician’s Perspective”. In: *International Journal of Radiation Oncology*Biology*Physics* 93.5 (Dec. 1, 2015), pp. 1127–1135. ISSN: 0360-3016. DOI: 10.1016/j.ijrobp.2015.07.2286. URL: http://www.sciencedirect.com/science/article/pii/S0360301615030783 (visited on 06/08/2019).

[92] Adnan Qayyum et al. “Medical image retrieval using deep convolutional neural network”. In: *Neurocomputing* 266 (Nov. 29, 2017), pp. 8–20. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2017.05.025. URL: http://www.sciencedirect.com/science/article/pii/S0925231217308445 (visited on 06/08/2019).

[93] Zhennan Yan et al. “Bodypart Recognition Using Multi-stage Deep Learning”. In: *Information Processing in Medical Imaging*. Ed. by Sebastien Ourselin et al. Lecture Notes in Computer Science. Springer

International Publishing, 2015, pp. 449–461. ISBN: 978-3-319-19992-4.

[94] Ke Yan et al. "DeepLesion: automated mining of large-scale lesion annotations and universal lesion detection with deep learning." In: *Journal of medical imaging* 5.3 (2018), p. 036501. DOI: 10.1117/1.JMI.5.3.036501.

[95] Timothy Rozario et al. "Towards automated patient data cleaning using deep learning: A feasibility study on the standardization of organ labeling". In: *arXiv:1801.00096 [physics]* (Dec. 30, 2017). arXiv: 1801.00096. URL: http://arxiv.org/abs/1801.00096 (visited on 01/25/2019).

[96] Shachar Kaufman, Saharon Rosset, and Claudia Perlich. "Leakage in Data Mining: Formulation, Detection, and Avoidance". In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '11. event-place: San Diego, California, USA. New York, NY, USA: ACM, 2011, pp. 556–563. ISBN: 978-1-4503-0813-7. DOI: 10.1145/2020408.2020496. URL: http://doi.acm.org/10.1145/2020408.2020496 (visited on 05/14/2019).

[97] Cancer Imaging Archive. *Cancer Imaging Archive Search*. URL: https://nbia.cancerimagingarchive.net/nbia-search/?ImageModalityCriteria=RTSTRUCT&MinNumberOfStudiesCriteria=1 (visited on 05/21/2019).

[98] Fabian Pedregosa et al. "Scikit learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (Oct. 2011), pp. 2825–2830. ISSN: 1533-7928. URL: http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html (visited on 05/23/2019).

[99] Irwin Sobel. "An Isotropic 3x3 Image Gradient Operator". In: *Presentation at Stanford A.I. Project 1968* (Feb. 8, 2014).

[100] Benjamin E. Nelms et al. "Variations in the Contouring of Organs at Risk: Test Case From a Patient With Oropharyngeal Cancer". In: *International Journal of Radiation Oncology*Biology*Physics* 82.1 (Jan. 1, 2012), pp. 368–378. DOI: 10.1016/j.ijrobp.2010.10.019. URL: http://www.sciencedirect.com/science/article/pii/S0360301610034401 (visited on 05/24/2019).

[101] David J. Hand and Robert J. Till. "A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems". In: *Machine Learning* 45.2 (Nov. 1, 2001), pp. 171–186. ISSN: 1573-0565. DOI: 10.1023/A:1010920819831. URL: https://doi.org/10.1023/A:1010920819831 (visited on 05/26/2019).

[102] Foster Provost and Pedro Domingos. "Tree Induction for Probability-Based Ranking". In: *Machine Learning* 52.3 (Sept. 1, 2003), pp. 199–215. ISSN: 1573-0565. DOI: 10.1023/A:1024099825458. URL: https://doi.org/10.1023/A:1024099825458 (visited on 05/26/2019).

[103] D. Mossman. "Three-way ROCs". In: *Medical Decision Making: An International Journal of the Society for Medical Decision Making* 19.1 (Mar. 1999), pp. 78–89. ISSN: 0272-989X. DOI: 10.1177/0272989X9901900110.

[104] *Support for multi-class roc_auc scores · Issue #3298 · scikit-learn/scikit-learn*. GitHub. URL: https://github.com/scikit-learn/scikit-learn/issues/3298 (visited on 05/26/2019).

[105] C. Chow. "On optimum recognition error and reject tradeoff". In: *IEEE Transactions on Information Theory* 16.1 (Jan. 1970), pp. 41–46. ISSN: 0018-9448. DOI: 10.1109/TIT.1970.1054406.

[106] N. Akhtar and A. Mian. "Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey". In: *IEEE Access* 6 (2018), pp. 14410–14430. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2807385.

[107] Li Fei-Fei, R. Fergus, and P. Perona. "One-shot learning of object categories". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.4 (Apr. 2006), pp. 594–611. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2006.79.

[108] Nima Sedaghat et al. "Orientation-boosted Voxel Nets for 3D Object Recognition". In: *arXiv:1604.03351 [cs]* (Apr. 12, 2016). arXiv: 1604.03351. URL: http://arxiv.org/abs/1604.03351 (visited on 05/13/2019).

[109] D. M. J. Tax and R. P. W. Duin. "Growing a multi-class classifier with a reject option". In: *Pattern Recognition Letters* 29.10 (July 15, 2008), pp. 1565–1570. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2008.03.010. URL: http://www.sciencedirect.com/science/article/pii/S016786550800113X (visited on 06/01/2019).

[110] Lydia Fischer, Barbara Hammer, and Heiko Wersing. "Optimal local rejection for classifiers". In: *Neurocomputing* 214 (Nov. 19, 2016), pp. 445–457. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2016.06.038. URL: http://www.sciencedirect.com/science/article/pii/S0925231216306762 (visited on 06/01/2019).

[111] Bianca Zadrozny and Charles Elkan. *Transforming Classifier Scores into Accurate Multiclass Probability Estimates*. 2002.

[112] Bianca Zadrozny and Charles Elkan. "Obtaining Calibrated Probability Estimates from Decision Trees and Naive Bayesian Classifiers". In: *Proceedings of the Eighteenth International Conference on Machine Learning*. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 609–616. ISBN: 978-1-55860-778-1. URL: http://dl.acm.org/citation.cfm?id=645530.655658 (visited on 06/06/2019).

[113] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. "Mining Association Rules between Sets of Items in Large Databases". In: *In: Proceedings of the 1993 Acm Sigmod International Conference on Management of Data, Washington Dc (usa*. 1993, pp. 207–216.

[114] *MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack*. The Journal of Open Source Software. Apr. 22, 2018. URL: http://joss.theoj.org (visited on 05/21/2019).

[115] Joseph P. Hornak. *The Basics of MRI*. URL: https://www.cis.rit.edu/htbooks/mri/inside.htm (visited on 05/20/2019).

[116] Thais Mayumi Oshiro, Pedro Santoro Perez, and José Baranauskas. "How Many Trees in a Random Forest?" In: Lecture notes in computer science. Vol. 7376. July 1, 2012. DOI: 10.1007/978-3-642-31537-4_13.