# MRI prostate cancer radiomics

## Assessment of effectiveness and perspectives

Pavlos Chatzoudis

Technische Universiteit Delft

TU Delft Delft University of Technology

Challenge the future

# MRI PROSTATE CANCER RADIOMICS

## ASSESSMENT OF EFFECTIVENESS AND PERSPECTIVES

by

**Pavlos Chatzoudis**

in partial fulfillment of the requirements for the degree of

**Master of Biomedical Engineering**

at the Delft University of Technology,
to be defended publicly on 2 February 2018 at 09.30.

| | | |
|---|---|---|
| Supervisor: | dr. J. Veenland | |
| Thesis committee: | Prof. Dr. W. Niessen, | TU Delft |
| | Dr. J. Veenland, | Erasmus MC |
| | Prof. Dr. F. Vos, | TU Delft |



TU Delft Delft University of Technology

# PREFACE

The cover photo is cited in [1]. The presented structure and patterns could well be a slice of an axial prostate MRI scan, distorted through the eyes of the artist. The colorful circles could resemble a representation of the various regions of interest within the prostate. Art has been and should keep being an inspiration for scientific progress.

I would like to express my deepest gratitude to Dr. Jifke Veenland, my direct supervisor, for her constant help and support throughout the completion of this project.

Finally, I would like to dedicate this work to the people that supported me through this effort selflessly and disregarding any personal costs.

to George,
Ioannis,
Cornelia,
Maria,
Dimitrios,
Angeliki

*Pavlos Chatzoudis*
*Delft, November 2017*

# CONTENTS

# 1

# INTRODUCTION

Radiomics is a newly emerging scientific field. Its purpose is to infer phenotypic differentiations between cancer lesions or healthy tissue. This can be achieved by taking into account features such as patient age and sex, or computed features such as statistical features or filters. Computed features can be extracted from medical images acquired by any modality. Commonly reported modalities include Magnetic Resonance Imaging (MRI), Computed Tomography (CT), Positron Emission Tomography (PET), Single Photon Emission Tomography (SPECT) and Ultrasounds (US). Fusion of multiple modalities is not uncommon. Applications concern many types of cancer, such as brain or lung cancer. One of the applications of radiomics is the attempt to classify prostate cancer (CaP) from MRI images, which is the focus of this thesis. This is very important, as it is aimed towards early diagnosis of aggressive prostate cancer which can lead to effective personalized treatment plans. Many efforts to this direction are reported in literature, where various feature types and classification methods are suggested. Although results reported in literature are often great, clinical application is vastly limitted. A possible reason for that is that the reported results, especially for MRI-based studies, might strongly rely on the available datasets, as well as the validation methods.

A common pipeline for prostate cancer MRI radiomics (CaP-MRI) is to extract features from MRI sequences. These features are subsequently used to train a model, which is able to diagnose cancer on a specific level of accuracy. Several methods are proposed in literature, however, in all of the reported cases, both training and testing data is obtained from a single clinic. This implies a single acquisition protocol and scanner manufacturer. The point of concern is that while computational radiomic features are highly quantitative, MRI is not. Therefore, it is questionable whether similarly good results could be achieved on data from another clinic. In this case, the scalability of the model's clinical application would be severely deterred.

This study tries to assess and reproduce results reported in literature. A custom built software routine extracts the most commonly reported features in literature. This comprises a set of 198 features for each pixel of the axial T2 sequences of two independent datasets. The following will be examined:

- Performance and prognostic value of the reported features
- Performance of different classifiers
- If distinction in peripheral zone improves results compared to examination of the entire prostate
- Finally, the potential to apply a model trained on a data from a single clinic on data from another one

This report follows the outline explained below:

1. A brief scientific review will clarify the nature of the most commonly reported features, as well as imaging modalities.
2. A brief review of the currently available literature on the subject will reveal the quality of the reported results and the fact that a totally independent dataset is nowhere used.
3. A discussion over the available datasets, as well as methods that were used to extract the features and evaluate classifier performance is then presented.
4. Results are discussed and conclusions are drawn.

## MOTIVATION

Prostate cancer (CaP) is the most frequent form of cancer for men in the developped countries. At the same time, it is the third leading cause of cancer related death for men in the same group of countries [2]. The extremely high rates of incidence and mortality globally dictate that action is needed. CaP-MRI radiomics are aimed towards an in time and effective diagnosis.

Concerning diagnosing practices, in past the urologist took random samples to investigate the prostate. More recent, imaging techniques have been used to improve the diagnostic process. The radiologist uses different sequences in combination in the diagnostic process since different characteristics can be inferred from the sequences. For the anatomical location the T2 sequence is used, where tumors are often represented by a lower signal intensity than the environment. With the Diffusion weighted sequence (DWI), tumors that consist of closely packed cells will be visible due to the restricted water motion. With the DCE (Dynamic Contrast Enhanced) MRI leaky vessels of tumor tissue can be detected.

Images can also be used to guide the sampling process during biopsy taking TransRectal Ultrasound guided biopsies (TRUS) or MRI guided biopsies. However, the taking of biopsies is an invasive procedure with the risk of causing infections. Moreover, lesions can be missed [3].

The radiomics hypothesis, presented in detail in 1.2 suggests that a tumor's phenotype can be inferred by imaging data, in a non-invasive way. The attempt to use quantitative image features in deciding about a tumor's stage is not only limited in diagnosing a tumor. It is also possible to classify tumors according to their aggressiveness, compute the possibility of metastasis, assess and even maybe foresee treatment response or recurrence. The aforementioned capabilities, open a new way to personalized treatment plans [4–6]. This could potentially lead to great treatment efficiency. Moreover, this sets a non-invasive and zero-exposure way to diagnose. All these in turn, set the motivation for those efforts.

## THE RADIOMICS HYPOTHESIS

Each solid tumor presents different characteristics of genes and genes' expression. According to these genomic differentiations, response to different kinds of treatment approaches might greatly differ. Examining the phenotype itself would sometimes be a highly invasive procedure [7]. The advancement of machine learning techniques, combined with computational power that emerged in recent years, gave birth to a whole new possibility, that of radiomics.

Radiomics hypothesis states that a solid tumor's phenotype can be inferred by medical imaging data [8, 9]. This can be achieved from information acquired by any modern imaging modality (PET, CT, MRI, US, SPECT). Efforts to evaluate the prostate by applying image assessment techniques according to this hypothesis have been a topic of study since the 80's [10]. Some steps have been taken towards the confirmation of this statement. Regarding prostate cancer, according to [11], it has been shown in [12, 13] that an actual correlation between MRI-image data and the tumor's Gleason Score is actually present.

## THE RADIOMICS PIPELINE

The radiomics pipeline involves obtaining a set of medical images, segmenting the Regions of Interest (ROIs), extracting the features and finally analyzing [8, 14]. This pipeline is common for any application besides CaP-MR, therefore, a generic workflow outline is presented below:

- Images can be obtained by any imaging modality, such as CT and FDG-PET [6, 15, 16], that are both used pretty often as a basis for radiomics as they both contain easily quantifiable measurements. Furthermore, several MRI modalities are commonly used (such as Diffusion Weighted Imaging (DWI), T2 weighted imaging (T2W), Dynamic Contrast Enhanced MRI (DCE-MRI) and Magnetic Resonance Spectroscopy (MRS) [12, 17, 18]). As these images are going to be used as an evaluation set, they come from patients' cohorts with confirmed histopathology, which is going to be used as ground truth in the subsequent steps [5, 19].
- Segmentation of the ROIs that are candidate tumor voxels, can be carried out either manually or automatically. When done manually, it is usually carried out by experts, where however interobserver variability is an issue [16]. On the other hand, there are several classical image processing approaches suggested that could be applied to get a semi-automatically or automatically segmented ROI [20–22].

Applications of automatic segmentation, could be a step towards a Computer Aided Software (CAD) for cancer diagnosis and maybe aggressiveness classification [23–26].

- Radiomic features are mostly applied to extract information explicitly about the image itself, such as texture [27]. They can also be combined with functional features such as ADC maps (see 2.3) and metabolic features, such as MRS images (see 2.2). This could provide a highly informative feature set [28]. Some key challenges in this step include fusing information from various modalities and keeping feature space low dimensional [29].
- Analysis consists in training a classifier with the extracted features, labeled according to the ground truth, which consists of prostatectomies. As with features, there is a large number of classification approaches that has been tested. Common classifiers are Support Vector Machines (SVM) [23, 24, 27, 30, 31], Adaptive Boosting (AdaBoost) [18, 30] and Random Forests [25], although there are also many more options, as Neural Networks. Given the relatively small size of the currently available datasets, Neural Networks do not seem to be the optimal solution.

## DECISION OBJECTIVES-PURPOSE OF STUDIES

Study objectives in CeP-radiomics literature vary. Several subcategories can be distinguished, the most common being plain diagnosis [25, 26, 30–37]. Authors also used classification techniques to classify tumors according to their aggressiveness and estimate the Gleason Score (GS) for prostate cancer [17, 38, 39], where the most reliable results are reported for GS > 7. Attempts to classify according to the newly established PI-RADS reporting system, presented in [40], are also present [18]. While most attempts adopt a single classifier approach, in [41] a cascaded classifier approach to distinguish between lower and higher GS is reported. Lowest Gleason Score candidates are first classified against the entire sample, followed by higher GS samples. In this way, it is deemed easier to distinguish between different GS, than address the problem at a single step aproach. A cascaded classifier is also trained in [42], in order to distinguish between benign and malignant confounders.

By further evolving this perspective, there have been several attempts to create a complete Computer Aided Diagnosis system (CAD). In [43] a review of all the CAD development approaches and results is provided. According to the authors, CADs can be distinguished in Computer Aided Detection (CADe) and Computer Aided Diagnosis Systems (CADx). Although it is common to attempt the creation of a fully automated system, a combined, CAD-Radiologist diagnosis plan is an option that might yield very good results [44].

From the above, it is obvious that, as radiomics are mostly used to diagnose the presence of a malignant lesion or even to classify the aggressiveness, they can also assist in choosing between treatment options, as lesions of different aggressiveness require a corresponding treatment plan.

## QUANTITATIVE IMAGING-THE NEED FOR STANDARDIZATION

Extraction of radiomic features refers to quantitative imaging features. This implies that conclusions drawn on a specific dataset should be reproducible. Moreover, the training of any classifier, that normally takes place on a specific dataset, should be globally applicable. That is, if a classifier is trained on a dataset, available at a specific clinic, this same classifier with the current training, should be able to be applied on any data, regardless the clinic. This also holds for images of the same patient, taken on different times or dates. This implies independency from the imaging device, as well as imaging protocols. To this point, CT and PET provide measurements that are easier to quantify than MRI. CT yields Hounsfield Units (HU) that are, within a statistical interval, independent of the scanner. PET measurements are reported in Standardized Uptake Values (SUVs) that follow the same principles as HUs.

MRI, however, is a modality that is hardly quantifiable [45]. In [30] it is noted that a robust feature set should perform the same with or without preprocessing, as robust features might be features that try to exploit texture information, thus not relying directly on gray level values that might significantly vary according to device/protocol. An application of classification on features that directly rely on quantitative maps intensity values is though not uncommon. Common preprocessing steps try to reassure this replicability, as well as to ensure the relevancy of the calculated feature values by removing noise etc. Preprocessing steps include:

- Intensity Normalization
- Denoising

- Registration/Segmentation
- Background Inhomogeneity/Bias Correction

## INTENSITY NORMALIZATION

An initial attempt to address the issue of intensity normalization is presented in [46].Extraction of a range for region based volume scans is carried out from a training set consisting of several volume images of the region. Subsequently, a linear interpolation scheme is suggested that would exploit the entire dynamic range and would normalize intensity values within this interval. According to [45], the most efficient way to normalize for T2W gray values and achieve reproducibility, is to restrict all values within an interval of $\pm 3\sigma$ of the mean value and quantize to a fixed bin size. In the above, $\sigma$ is the variance of gray values distribution. This technique is applied by several authors during MRI-based radiomic features extraction [31, 47]. A novel approach to normalization is presented in [48]. What the authors propose, is an alternative to the z-score method that is currently widely applied. Z-score method normalizes gray values as follows:

$$Y_{ij} = \frac{x_{ij} - \mu_i}{\sigma_i} \tag{1.1}$$

Where in the above, indices i and j refer to the image type and patient respectively. An average $\mu$ has to be calculated for each image and a standard deviation $\sigma$, so that transformed grey values have a zero mean and unit standard deviation.

Instead, it is suggested that Z-score normalization should be carried out class-wise, where there can be multiple classes. Depending on the application, these classes could be comprised of cancer vs non cancer tissue (which is the presented case), or even slighter discriminations. As, however, the classes cannot be a priori in hand, an iterative implementation is suggested, where at each iteration the estimation of a voxel membership in a class is updated. In that way, the range of values within a class is most effective exploited.

### DENOISING

The noise by which MRI images are corrupted is a further issue. Raw calculated values' real and imaginary part that correspond to the k-space coordinates, both suffer from Gaussian noise, but the amplitude of the complex data results in Rician noise distribution [49]. Rician distribution has the following mathematical expression:

$$p(M_i|A) = \frac{M_i}{\sigma^2} e^{-(\frac{M_i^2+A^2}{2\sigma^2})} I_0(\frac{AM_i}{\sigma^2}) u(M_i) \tag{1.2}$$

Where in the above, $\sigma$ corresponds to Gaussian noise standard deviation affecting imaginary and real part (with amplitudes $A_i$ and $A_r$ respectively) of values calculated in k-space, $A$ is the amplitude of $A_i$ and $A_r$, $I_0$ is the 0-th order first kind Bessel function and $M_i$ is the i-th data point of magnitude image.

Therefore, Gaussian noise filters should be avoided [50, 51]. In [36], the Minimum Absolute Deviation (MAD) technique presented in [52] is applied to remove the Rician noise, by which MRI images suffer. There are several more possibilities for Rician noise denoising, like those mentioned in [53–55].

### REGISTRATION

Image registration is needed both for motion correction and to register data obtained from different MR modalities and also possibly even MR fusion with completely other imaging modalities, such as US and CT [36, 56]. Moreover, image registration is necessary to correspond MR images with ground truth (most often manually registered histopathological slices [42, 57]). Although the rest of the preprocessing steps are not always present, registration is necessary in every study.

In [23], a specialized MRI prostate registration approach is presented, based on previous works presented in [58, 59]. The proposed method combines a global and a local transformation model. The global model is utilized via a rigid transformation (a transformation model that does not allow for any elastic deformations, is however simpler to implement as it contains less degrees of freedom). The local motion model, is being described by B-splines deformation model. B-splines rely on a deformable grid of control points that adapt to

the new shapes. Maximization of mutual information (an entropy based metric according to the voxel values' alignment) leads in the end to registration.

Atlas based registration is another popular approach. In [60], a multi-atlas model is used. An atlas is a reference labeled image of known anatomy, estimated as a reference. The image is first registered to each atlas separately via a non-rigid B-spline registration procedure and then a fusion of all deformation vectors is implemented by majority voting. There is also the option of using just one of the available atlas images, according to the maximiztion of a similarity measure between the reference image and the corresponding atlas.

In [61], the atlas is first estimated by a set of images as an average, alongside with a probability map $I_p$ that epxresses the probability that a point actually belongs to the structure of interest. The demons matching algorithm [62] is applied on the levels of a pyramidal decomposition (see sec. 3.3) to register the pyramid levels to the atlas. Finally, a deformable model is initialized according to the grid dictated by the demons' forces calculated in the first step. This consists of an image feature model, a spatial constraint model and a statistical shape model (SSM).Finally, an iterated procedure leads to automatic prostate gland segmentation.

Registration and segmentation is a huge chapter in literature and there are many proposed techniques. Most of the excluded query results were excluded due to their aim to registration and segmentation of the prostate gland, which by itself is a very interesting topic but it is out of interest for now.

### INHOMOGENEITY/BIAS CORRECTION

Inhomogeneity can be described as an artifact of variation of the image intensity across the image. It can be ascribed to "static field inhomogeneity, bandwidth filtering of the data, eddy currents driven by field gradients, and especially radio frequency (RF) transmission and reception inhomogeneity."[63]. Other sources are the stochastic nature of the imaged object's magnetic properties and shape. While the first kind of inhomogeneity is possible to accurately bypass with frequent proper calibration and by applying proper protocols, the second has to be dealt with great care. A popular way to handle this is presented in [64]. According to it, bias field has an effect of the form:

$$v(x) = u(x)f(x) + n(x) \tag{1.3}$$

Where in the above equation, $u$ is the actual signal, $f$ is a slowly varying bias field and $n$ is the Rician noise. $f$ is considered to follow a Gaussian distribution. It is subsequently considered as the convolution product of narrower Gaussians. Actual distribution $u$ can then be estimated by applying the following deconvolution filter in the Fourrier domain:

$$\hat{U} = \frac{\hat{F}^\star}{|\hat{F}|^2 + Z^2}\hat{V} \tag{1.4}$$

Where in the above, $\hat{F}$ is the bias field estimation in the Fourrier domain, $F^\star$ is its complex conjucate and $\hat{V}$ is the received signal.

Several further techniques are reported regarding this kind of correction in literature [63]. According to [65], inhomogeneity correction should be performed after gray value standardization.

### OTHER PREPROCESSING TECHNIQUES

An interesting preprocessing step, that does not actually have something to do with the properties of MRI itself but merely the proper training of the classifier, is presented in [17]. As the classes (cancerous vs non-cancerous) present imbalance, oversampling of the minority class or sample weighting is applied in order to achieve balance. This is a very important step for the specific implementation, as the applied classifiers' performance (AdaBoost and some SVM variants) strongly depend on the reported accuracies. Techniques used to oversample are Synthetic Minority Oversampling Technique (SMOTE) [66] and sample generation with Gibbs sampling [67].

# 2

# MULTI-PARAMETRIC MRI

Multi-paramteric MRI (mpMRI) includes Diffusion Weighted Imaging (DWI), Dynamic Contrast Enhanced (DCE) and T2 Weighted Imaging (T2W), while in some cases also Magnetic Resonance Spectroscopy (MRS) is included. Although fusion of information from these modalities might be a challenging step [25], it is recommended that the combination of all these modalities can provide useful diagnostic information [68]. On the other hand, in [69], authors suggest that fusion of several MR modalities does not necessarily improve detection accuracy. This dispute however falls outside our specific area of interest, as we aim only for the assessment of texture and texture related radiomic features. In general, lots of combinations of features and feature selections have been tested. There are several feature subsets that comprise the entire feature set usually used for CaP CAD detection. There are several components or statistics in each of these families, that comprise in some cases a family of as many as 635 features [6]. Basic principles that concern feature extraction from the modalities that consist mpMRI are presented below.

## DCE

Througout the examined literature, DCE-MRI has mostly been applied to extract functional pharmacokinetic features. DCE-MRI results after injecting gadolinium-based contrast agents [70]. Pharmacokinetic features are curves that estimate perfusion of an area according to some contrast enhancement medium parameters related to its concentration through time, such as wash-in, peak and wash-out times. Pharmacokinetic features can be extracted either for a specific ROI or per-voxel basis [71]. There have been proposed models for the estimation of these model based parameters ($K_{trans}$), such as in [72]. $K_{trans}$ maps is one of the most commonly applied quantitative maps in literature, from which information other than functional is extracted. $K_{trans}$ (bulk transfer coefficient) is a constant that, according to [71] can be generally estimated from:

$$C_t(t) = K_{trans} \int_0^t C_p(t') e^{-\frac{K_{trans}}{v_e}} (t - t') dt' \tag{2.1}$$

Where in the above equation, $C_t(t)$ is the concentration of the agent in voxel t at time t, $C_p(t)$ is the concentration of the agent in the plasma volume and finally $v_e$ is the proportional volume of the extravascular extracellular distribution space.

It can be also estimated from the modified Tofts model, that, at the same time, allows for the calculation of $v_p$ (fractional plasma volume):

$$C_t(t) = v_p C_p(t) + K_{trans} \int_0^t C_p(t') e^{-\frac{K_{trans}}{v_e}} (t - t') dt' \tag{2.2}$$

It is however also possible to extract quantitative information from a model free Area Under the Curve (AUC) estimation for the concentration versus time curve ($IAUC_{60}$) [73, 74]. In this review, we are only interested in caclulation of map/image features based on the DCE maps and not their functional interpretation. The most common method would be to exploit directly pharmacokinetic map intensity values. However, it is mentioned

in [75] that a blob detection technique described in [37] is used not only on $K_{trans}$ but also on other quantitative DCE-MRI maps.

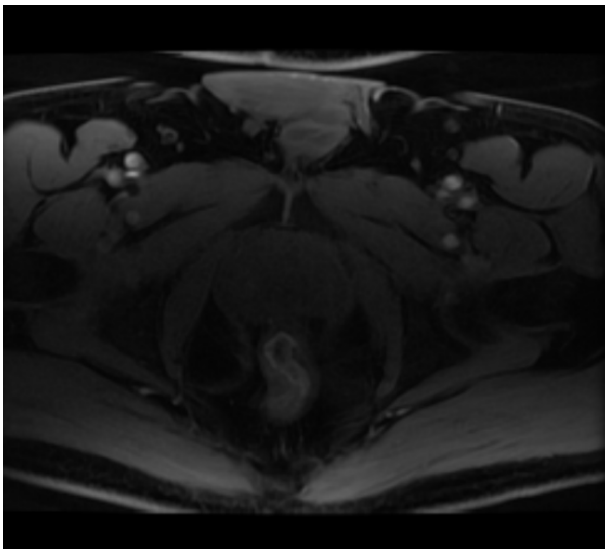


Figure 2.1: Axial DCE image of the prostate of patient 01.0058 of the PCMM dataset

## MRS

Another modality that often comes alongside MRI in CaP detection efforts, is Magnetic Resonance Spectroscopy (MRS). MRS is recommended as a useful marker in [68]. Received MR signal contains several resonances that are due to the difference of specific metabolites' concentration within a region of interest. In turn, differences in resonance frequencies of these metabolites are due to differences in 1H concentration in the metabolites. These metabolites are Choline (Cho), Phosphocreatine and Creatine (Cr) and Citrate (Ci), yielding three respective spectral peaks. It is feasible to localize where this signal originates from with a very good accuracy in the range of 6.2 mm. From the height of the peaks, it is feasible to possible to determine metabolite concentration in the specific region and a high $(Cho+Cr)/Ci$ ratio indicates presence of cancer[76, 77]. Through the literature, MRS maps only exploit this ratio without extracting any further radiomic features. Therefore it will not be further examined.

## DWI

DWI works by applying inversed pulses. Accumulated phase for a single spin under the influence of a gradient rectangular pulse is:

$$\phi(t) = \gamma B_0 t + \gamma \int_0^t \mathbf{G}(t') \cdot \mathbf{x}(t') dt' \tag{2.3}$$

Where $B_0$ is the bias DC magnetic field, $\gamma$ is the gyromagnetic ratio and $\mathbf{G}$ is the refocusing pulse. Under the influence of a 180 delayed refocusing pulse, applied at a point in time $t_1 + \Delta$ after the first pulse applied at $t_1$ and both having a duration $\delta$, the same quantity becomes

$$\phi(t) = \gamma B_0 t + \gamma \int_{t_1}^{t_1+\delta} \mathbf{G}(t') \cdot \mathbf{x}(t') dt' - \gamma \int_{t_1+\Delta}^{t_1+\Delta+\delta} \mathbf{G}(t') \cdot \mathbf{x}(t') dt' \tag{2.4}$$

The influence of this sequence at a spin that is not moving is a zero shift in phase, as the phase accumulation cancels out. For the moving spins, it can be shown that the signal intensity reduces exponentially according to

$$SI = SI_0 e^{-bD} \tag{2.5}$$

Where $D$ is the apparent diffusion coefficient and $b$ is the diffusion-sensitizing factor, depending on the pulse sequence properties:

$$b = \gamma^2 G^2 \delta^2 (\Delta - \frac{\delta}{3}) \tag{2.6}$$

From all the above, we can estimate apparent diffusion coefficient for the regions of interest [78]. As diffusion coefficient is closely related to cellular density in an area, denser areas are expected to present lower diffusion. This is the case of cancerous regions [79]. A common approach is to extract an ADC for several different b-values. An ADC map constitutes in the interpolation of all intermediate b values on the regions of interest. A tumorous region is expected to have a smaller slope than a healthy one [79]. Although a high b-value is expected to have higer diagnostic accuracy [34, 75], due to hardware limitations an upper limit of b-value can be achieved. b-value is reported in $s/mm^2$ units and the highest values that can be currently achieved are around $1500 s/mm^2$. A way to bypass this issue is suggested in [80] and implemented as a part of [34]. A high-b value can be estimated from eq. 2.5 for the acquisition of $S_i$ measurements for several different b-values. A Bayesian Inference approach is suggested, according to which $D$ could be estimated and subsequently a higher b could be extracted. Throughout the examined literature, ADC maps is basically the feature extracted from DWI.

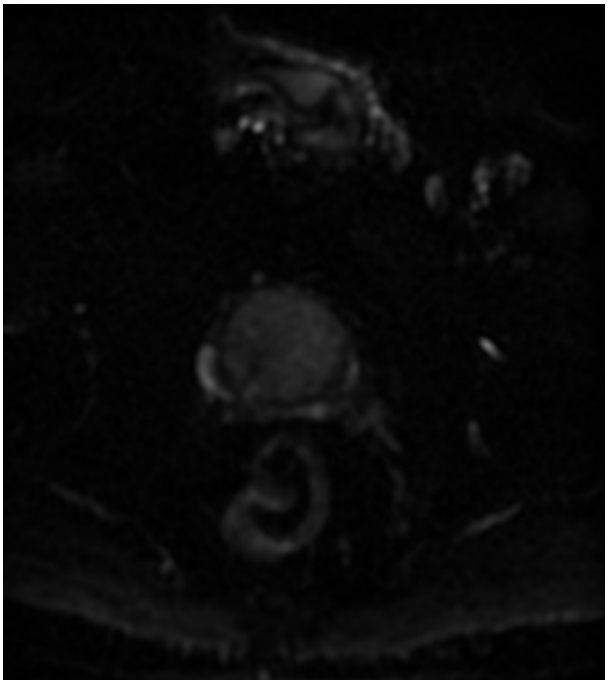


Figure 2.2: Axial DWI image for b=100 of the prostate of patient 01.0058 of the PCMM dataset

An alternative to ADC maps, namely Correlated Diffusion Imaging (CDI) is suggested in [81] and implemented in [34]. CDI does not rely on signal attenuation. Instead of interpolating between known values for b, CDI is calculated by computing the correlation between several signals acquired by the application of pulse sequences with different characteristics (which would result to different b-values according to eq. 2.6) and the calculate the correlation between them. If $x$ indicates a spatial location and $S_{q_a}, S_{q_b}, \ldots, S_{q_n}$ are the acquired signals for the site indicated by $x$ and f is the conditional joint probability density function, CDI signal is calculated for $x$ by:

$$\text{CDI} = \int \cdots \int S_{q_a} S_{q_b} \ldots S_{q_n} f(S_{q_a} S_{q_b} \ldots S_{q_n} | V(x)) x dS_{q_a} dS_{q_b} \ldots dS_{q_n} \tag{2.7}$$

Authors that extract any kind of radiomic features for T2W images, usually extract the same features for DWI extracted information. DWI radiomic features might be drawn ADC maps or, most rarely CDI. Also features based on intensity values of DWI images are sometimes examined. In [82, 83], results of possible combinations between features' extraction from ADC, high-b value DWI and T2W images show that the most potent combination is fusion of features calculated on ADC maps and T2W.

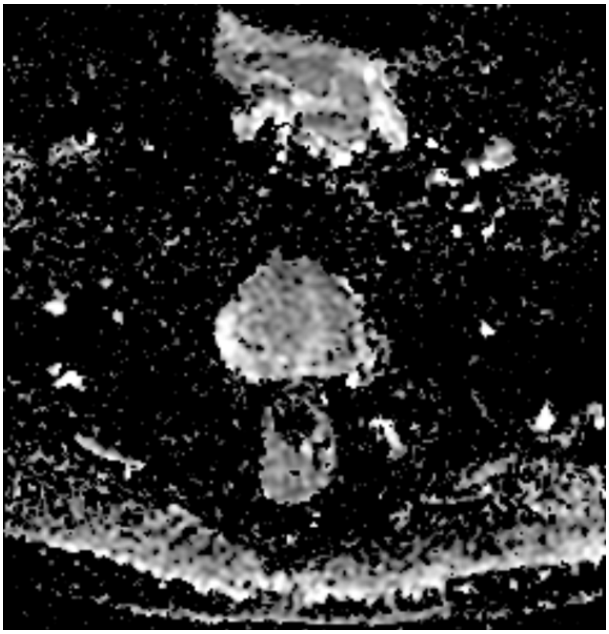


Figure 2.3: Axial ADC map of the prostate of patient 01.0058 of the PCMM dataset

## T2W AND T1W

T2W is the most commonly used modality in the assessment of the prostate, being present in almost any of the examined publications. Depending on scanner field strangth and protocols, T2W images can usually achieve sub milimeter accuracies, while other modalities can only achieve accuracies in the order of several milimeters (see tab. 2.1 [84]). Therefore, it is usually used also as the basic anatomic image on which maps from other modalities are registered. T2W Imaging relies on estimating the transverse relaxation times of the specimen under examination (T2 times). Alternatively, T1W Imaging works by estimating the longitudinal relaxation times. Usually, T1 times are way longer than T2 times. In mpMRI CaP detection, T1 times are most often measured at DCE-MRI [71, 85], while it is used directly as a quantitative map in [37].

Table 2.1: typically reported mpMRI modalities resolutions

| Modality | Voxel Size (mm) | Slice Thickness (mm) |
|---|---|---|
| T2W | 0.72-0.96 | 2.5 |
| DCE | 1.16 | 3 |
| DWI | 2.26-2.89 | 5 |
| MRS | 8-9 | 15 |

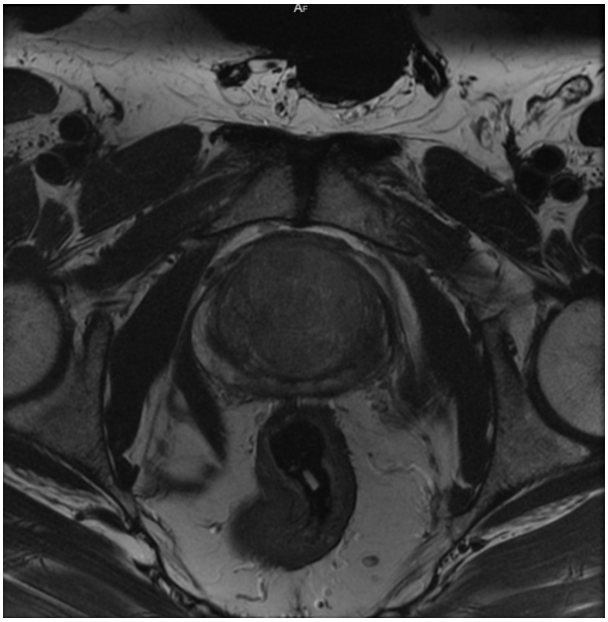


Figure 2.4: Axial T2 image of the prostate of patient 01.0058 of the PCMM dataset

# 3

# Common Radiomic Features

Computed radiomic features are mostly features trying to describe texture. According to [86], they can be distinguished in the following categories:

- Histogram Based Features (statistical)
- Gradient Features
- Run-Length Matrix
- Co-occurence Matrix
- Auto-regressive model
- Wavelets

In this study, next to these features, also the following ones are examined:

- Fractal Dimension
- Morphological Features

Moreover, in [87], some spatial filters presented in [88] are implemented as features. These features are not going to be specially examined in the following section. Nevertheless, they have been implemented in our routine.

## Histogram Based Features

### First Order Statistics

If a region consists of N pixels and I of them have an intensity level i, then we can define the probability distribution $P(I)$ as follows:

$$P(I) = \frac{I}{N} \tag{3.1}$$

According to this distribution we can define first- and second-order statistics. Under the term first order statistics, the $k^{th}$ order moments are meant. A $k^{th}$ order moment is defined as:

$$\mu_k = E\left[(I - E(I))^k\right] = \sum_{1}^{N_g - 1} (I - m^1)^k P(I) \tag{3.2}$$

Where $\mu_1 = \sum_{I=0}^{N_g-1} I^1 P(I)$

The following first order statistics are thus defined [89]:

Table 3.1: First order statistical features

| k | moment name |
|---|---|
| 1 | mean |
| 2 | variance |
| 3 | skewness |
| 4 | kurtosis |

Furthermore, in several cases [28, 90], only a thresholded low-or high- percentage of the values of an image is kept, as it is associated with functional features. Usually this is the case for ADC maps, where, a spot with lower diffusivity could potentially comprise a hotspot [75]. It is common for first order statistics to be calculated over a moving window in the range of 3 to 9 pixels or voxels, as for example in [36], a 3x3 pixel neighborhood is considered, while in [41], first order statistics from three different window sizes are used, namely 3x3, 5x5 and 7x7. At this point we should also mention some special features introduced in [13]. For a 3D ROI of size 3x3x3 or 5x5x5:

$$\mathbf{f^1}(\mathbf{u}) = MED_{\mathbf{u_t}\in N_k(\mathbf{u})}[f(\mathbf{u_t})] \tag{3.3}$$

$$\mathbf{f^2}(\mathbf{u}) = \sqrt{\frac{1}{N_k(\mathbf{u})}\sum_{\mathbf{u_t}\in N_k(\mathbf{u})}[f(\mathbf{u_t})-f((u))]^2} \tag{3.4}$$

$$\mathbf{f^3}(\mathbf{u}) = \left|f(\mathbf{u})-\frac{1}{N_k(\mathbf{u})}\sum_{\mathbf{u_t}\in N_k(\mathbf{u})}[f(\mathbf{u_t})]\right| \tag{3.5}$$

Where in the above, MED is the median operator and the followng 2 measures are standard and average deviation. $\mathbf{u}\in C$ is a voxel within the ROi, where $C$ is a scene. $u$ is the central pixel of the ROI, around which first order statistcs are calculated and $N_k$ the set of pixels that surrounds it. Moreover, in [34] , the following statistical measures are introduced

$$M = \frac{1}{N}\sum_{x,y=0}^{N-1} pix(i,j) \tag{3.6}$$

$$En = -\sum_{l=1}^{k} p(l)log_2[p(l)] \tag{3.7}$$

$$U = \sum_{l=1}^{k}[p(l)^2] \tag{3.8}$$

Where in the above equations, $p(l)$ is the histogram value at gray level $l$ and $pix(i,j)$ is the pixel intensity value at a position $(i,j)$ inside the ROI.

Finally, an interesting approach for the variance estimation is presented in [82], where the local variance around a central pixel $g_c$ is estimated by the values of $P$ neighboring pixels lying on a radius $R$ (according to a local binary patterns' arrangement, see 3.2)

$$VAR_{P,R} = \frac{1}{P}\sum_{p=0}^{p-1}(g_p-\mu)^2$$

$$\text{where}\mu = \frac{1}{P}\sum_{p=0}^{p-1} g_p$$

## SECOND ORDER STATISTICS

Haralick features are statistics that are directly associated to the texure of an image. They are computed form the Grey Level Co-occurence Matrix (GLCM). GLCM occurs as follows: If all the grey levels contained in an image are M, it is an MxM matrix. The element in the position (i,j) contains the total number of pixels with a grey intensity i that are direct neighbors with a pixel of grey intensity j [39].

Thereafter, 28 statistical features presented in [91] can be calculated. Four statisctial features can also be calculated for Run-Length Matrices (GLRLM) described in [92]. A GLRLM is a texture measure somewhat simillar to

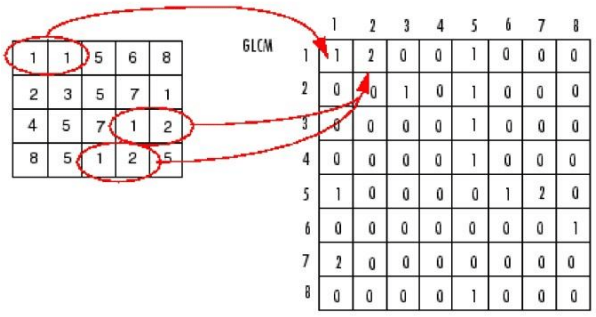



Figure 3.1: How the Grey Level Co-occurence Matrix at direction 0 is extracted

a GLCM. For an image of size NxN containing M grey levels, a GLRLM is an MxN matrix, where each row represents a grey level and each column represents the population of uninterrupted chains of pixels of that specific intensity contained within the image. There are 4 possible GLRLMs, meant for chains directed towards 0°, 45°, 90°and 135°, as in case of GLCMs.

## GRADIENTS

Image gradients are the grey value gradients to a selected direction, therefore they can be applied also as direction selective features. Definition of a gradient for an image $f$ is as follows:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \tag{3.9}$$

Where the most basic filters applied to estimate x and y are respectively:

$$\begin{aligned} \mathbf{G}_x &= \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \text{or} \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \\ \mathbf{G}_y &= \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \text{or} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \end{aligned}$$

For diagonally oriented gradients, one can use the Roberts cross-gradient operators:

$$\begin{aligned} \mathbf{G}_{diag1} &= \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \text{or} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \\ \mathbf{G}_{diag2} &= \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \text{or} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \end{aligned}$$

While for x and y oriented gradients, there are also the Sobel filters presented below. Sobel filters somewhat smooth the gradient by assigning a weight of 2 to the center point:

$$\begin{aligned} \mathbf{G}_x &= \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{or} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \\ \mathbf{G}_y &= \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \text{or} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \end{aligned}$$

As we can see in the above equations, alternatives within the same gradient type refer to different direction or orientation selection. For example, in eq. 3.10, the first matrix is highly selective to vertical edges, while the second one to horizontal.

Another popular option is the Kirsch filters. Kirsch filters are used for blob detection, where as blob is considered any contiguous region surrounded by a border of similar brightness. It is a gradient based filter, where the kernel described in eq. 3.10 is rotated and applied in 8 directions. Subsequently, the maximum value of all these convolutions is kept as the track of the contour [93].

$$\mathbf{g^i} = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \tag{3.10}$$

Application of Gaussian filters is also mentioned as another option. Gaussian filters consist in the convolution with Gaussian kernels of variable standard deviation $\sigma$, if $x$ and $y$ refer to the central pixel around which the filter is applied:

$$\mathbf{G}(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3.11}$$

In [75], Gaussian filter banks of $\sigma$ ranging exponentially between 2 and 8 mm are used. The choice of the specific kernel sizes is justified as being most sensitive to malignant areas, as the lesion size is said to have been observed empirically to vary within a specified range of typically 2 to 20 mm. Another filter that is used in the same reference is the filter reported in [88]. This filter is sensitive in blob detection. Originally developed as a method to extract information for lung cancer from CT images, it also could be useful in case of MRI prostate cancer images. This filter, has been developped as to achieve high sensitivity to lines and dots that would represent pulmonary vessels and nodules. If $\lambda_1$ and $\lambda_2$ are the two eigenvalues of the diagonalized Hessian matrix applied on the entire image, the filter can be expressed as

$$z_{dot} = \begin{cases} \frac{|\lambda_2|^2}{|\lambda_1|} \text{if} \lambda_1 < 0, \lambda_2 < 0 \\ 0, \text{otherwise} \end{cases} \tag{3.12}$$

$$Z_{line} = \begin{cases} |\lambda_1| - |\lambda_2| \text{if} \lambda_1 < 0 \\ 0, \text{otherwise} \end{cases} \tag{3.13}$$

A filter that is really popular in texture classification is Local Binary Patterns, presented in [94]. Local binary patterns around any point within an image is roughly obtained as described below. Values on a selected radius of $R$ pixels around a point of interest are obtained by interpolation. Then, a set of points $P$ is selected on the specified radius. This filter is denoted by $LBP_{P,R}$ and its value is calculated as follows:

$$LBP_{P,R} = \sum_{p=0}^{p-1} s(g_p - g_c) 2^p \tag{3.14}$$

In the above equation, $g_c$ stands for the value of the central pixel, $g_p$ is the gray value of the selected pixel $p$ on a raius $R$ and $s$ is the step function defined as:

$$s(t) = \begin{cases} 1, t \geq 0 \\ 0, t < 0 \end{cases}$$

According to the above equation, each unique combination would yield a unique code that would identify the local binary pattern. A rotation invariant version of this procedure, yields a unique set of possible combinations, or local binary patterns. A great property of LBP filters is that they are gray scale invariant, as they solely rely on the differences of gray values between neighboring pixels.

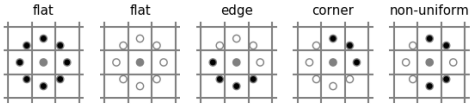



Figure 3.2: How Local Binary Patterns' histogramms are extracted

In [82], local binary patterns are calculated as a feature. Moreover, a second order modified version is presented, namely Local Directed Derivative Patterns (LDDP).

$$LDDP^2_{P,R} = \sum_{p=0}^{p-1} s(d^2_{p,R})2^p \tag{3.15}$$

$$\text{where} d^2_{p,R} = g_p^{R_2} + g_c - g_p^{R_1} \tag{3.16}$$

$R_1$ and $R_2$ are two successive radii.

While Gabor filters can be regarded as a wavelet family (see 3.3), they will be treated in this section as plain filters. Gabor filters are 2D sinusoidals contained in a Gaussian envelope. They are usually applied in a 2-D Fourrier space. Gabor filters are also related to Gabor wavelets, since they can be applied in several directions and scales (see section 3.3). Therefore, they imply the generation of many possible features, to many scales and directions [95].

While more sophisticated filter designs, as the one presented in [88], Gaussian filters or Gabor filters deal with the issue of noise, the major drawback of plain gradients is that they can turn out to be very prone to noise.

## MULTI-SCALE FEATURES

Wavelets are used to conduct multi scale analysis. Multi scale analysis is used in image compression, or to easily detect structures suspected to be within a specific scale. A popular multi-scale analysis tool is image pyramids, where on the basis of the pyramid there is the full resolution image and at each level up, resolution reduces by two. Each level of the pyramid is extracted by low pass filtering and downsampling the previous level by a factor of 2. Thus, an image of NxN pixels is reduced at each step to an image of $\frac{N}{2} x \frac{N}{2}$, where the previous level preserves the level of details that are lost in the previous one, so that a perfect reconstruction is in the end possible. Pyramid decomposition is used in [41], where features are extracted for each resolution scale. Low pass filtering is necessary to avoid aliasing in the next detail level. The most popular kind of pyramids is Laplacian pyramids, where the filter is implemented by Gaussian bluring at each scale [96], [97].

Another approach is that of subband coding, where the input signal is analyzed into discrete frequency subbands that are subsequently downsampled without loss of information.

Wavelet decomposition is applied when it comes to multiresolution or multi-scale analysis. While the Fourrier transform of an image signal provides information about multiple scales, its downside is that it cannot be correlated to where the details at each scale are within an image. With wavelet decomposition, it is possible to localize the presence of scales and orientations within an image. While image pyramids are useful to extract information about different scales, the wavelet transform, a mathematical tool which can be viewd as an expansion of the Fourrier Transform, can also provide informations about scales and orientations, localized within an image. Wavelet decomposition has the following form:

$$W(\mathbf{f})(a,b) = |a|^{-\frac{1}{2}} \int_{-\inf}^{+\inf} f(t)\psi\left(\frac{t-b}{a}\right) dt \tag{3.17}$$

Where $a, b$ are scaling and translation parameters and $\psi$ is called the mother wavelet and it has to fulfil some criteria [98, 99]. Translations and scalings of the mother functions result into an orthogonal basis. Reconstruction can be achieved by summing up the basis functions multiplied by the corresponding coefficients, that indicate the content of power at each scale spanned by a basis function:

$$f = \sum \alpha_{j,k}\psi_{j,k}(t) \tag{3.18}$$

$$\alpha_{j,k} = < f(t), \psi_{j,k}(t) > \tag{3.19}$$

Where indices $j, k$ are integer parameters that act on parameters $a, b$ of eq. 3.17 and result in different basis functions. The application of these parameters to each scale, yields four images, namely the respective $j$ level image itself, an image where the downscale has been applied to rows, one image where the downscale has been applied to columns and one image for both. The coefficients of each image at each scale are used as features for texture description.

There are several popular options as mother wavelet functions. One of the most common options is the Haar wavelets, where the Haar functions are an orthonormal basis 3.20. The Haar transform of an image **F** can be calculated as $\mathbf{F}' = \mathbf{HFH}$ and **H** is the matrix containing the Haar basis functions cases, that are calculated as follows: For an image NxN,

$$\begin{aligned} z &\in [0,1] \\ k &= 0,1,2,...,N-1 \\ p &\in [0,n-1) \\ q &= \{0,1\} \end{aligned}$$

$$h_{pq}(z) = \frac{1}{\sqrt{N}} \begin{cases} 2^{p/2} & (q-1)/2^p \le z < (q-0.5)/2^p \\ -2^{p/2} & (q-0.5)/2^p \le z < q/2^p \\ 0 & \text{otherwise} \end{cases} \tag{3.20}$$

Besides Haar wavelet basis functions, there are also other wavelet basis functions, like Gabor wavelets applied in [13, 35, 41] or Daubechies waveletes [100]. Coefficients of Daubechies wavelets averaged over a 7x7x7 voxel window are used as texture descriptors in [30].

## FRACTAL FEATURES

In [101], [32] and [30] authors use fractal features to discriminate between textures. Fractal dimension is a measure describing the heterogeneity of an object. It was first described as follows. Consider a coastline surrounded by a strip of width $2\epsilon$. The suggested length of the coastline is $l(\epsilon) = \frac{\text{striparea}}{2\epsilon}$. As epsilon decreases, $l(\epsilon)$ increases. It was shown in [102], that for $l(\epsilon)$ holds

$$l(\epsilon) = F\epsilon^{1-D} \tag{3.21}$$

Where in the above, F and D are constants, D being the fractal dimension (FD). It is straightforward, that the fractal dimension is directly associated with the irregularities within an image. The problem for image texture analysis comes to estimate D accurately.

In [32], histogram fractal dimension (and not of the image itself) is extracted by the popular method of differential box counting, originally described in [103]. According to this method, the entire support field (an array MxM) is segmented into progressively thinner sections (of size exe), for which the FD is estimated according to the following equation:

$$N_e = \sum ceil\left[\frac{Md_e}{Ge}\right] \tag{3.22}$$

Where $d_e$ is the difference between the greatest and smallest value within each subregion of size exe and G is the global maximum value for the entire MxM array.

In [101] the blanket method originally reported in [104] is applied to extract the FD from the image texture.

As a remark, it is noteworthy that in [32], application of fractal features from only T2W images manage to achieve very high classification accuracy at a one-dimensional feature space. This might show a great potential in fractal features or raise further questions about the reproducability of the report results.

In [30], a novel method for fractal dimension approximation is introduced. According to the authors, 3D images can be modeled as a multifractional Brownian motion process, in which $FD = 4 - H$, where $H$ is the Hurst parameter.For the Hurst parameter, for the correlation of two points lying at points $t$ and $s$ respectively, if the fBm values are $X(t)$ and $X(s)$ respectively, it holds for their correlation:

$$\mathrm{E}[X(t)X(s)] \approx \frac{1}{2}||t^{2H(t)} + s^{2H(t)} - ||t-s||^{2H(t)}|| \tag{3.23}$$

It is concluded that the fractal dimension of such an image can be approximated by approximating the Hurst parameter by the following equation

$$log_2(\frac{V_j(B_H(t))}{V_{2j}(B_H(t))} \approx (2H(t)+3)j \tag{3.24}$$

Where indices $j$ and $2j$ stand for two successive scales and if $w_{j,k}$ is the Discrete Wavelet Transform coefficient at a scale $j$ translated by a vector $k$

$$V_j(B(H(t)) = \frac{1}{n_j}\sum_k w_{j,k}^2(B(H(t)) \tag{3.25}$$

Finally, we should mention the Discrete Cosine Transform (DCT) applied in [31] in local windows of 7x7x7 voxels. Although not a multi-scale feature itself, its mathematical representation resembles the decomposition of a signal to an orthogonal basis and therefore it is presented in this section. DCT implements the decomposition of a digital signal sequence $X(m)$ in translated cosine components, pretty similarly to DFT. As it is defined in [**?** ], DCT coefficients are calculated for a 1-D signal as

$$G_x(0) = \frac{\sqrt{2}}{M}\sum_{m=0}^{M-1} X(m) \tag{3.26}$$

$$G_x(k) = \frac{2}{M}\sum_{m=0}^{M-1} X(m)\cos\frac{(2m+1)k\pi}{2M}, k = 1,2,\ldots,(M-1) \tag{3.27}$$

## MORPHOLOGICAL FEATURES

In [18] and [75], it is suggested that also tumor shape characterisitcs should be quantified and used as features. In [18], three such features are used:

- Difference between morhpologicaly closed and open are of the lesion:

$$f_M^1 = \frac{A_{closed} - A_{open}}{A_{initial}}$$

- Difference between perimeter length before and after eliminating high frequency components:

$$f_M^2 = \frac{|P_{initial} - P_{reconstruction}|}{P_{initial}}$$

- Difference in area after low pass Fourrier reconstruction of the lesion shape:

$$f_M^3 = \frac{|A_{initial} \oplus A_{reconstructed}|}{|A_{initial} \cup A_{reconstructed}|}$$

- Asymmetry estimation by splitting the lesion with an axis along the center of mass and calculating the difference between the residual areas:

$$f^A = \frac{A_{large} - A_{small}}{A_{initial}}$$

Morhphological features used in [75] include tumor sphericity, volume and compactness. Asymmetry is also taken into account. While it is reported in [75] that application of such can prevented false positives that occurred during segmentation errors (as automatic segmentation is applied in their approach), it can be clearly seen that these features are rather prone to segmentation, even if it is manual (interobserver variability) [16]. In our approach, morphological features will not be taken into account.

# 4

# DATA PROCESSING

## DATA SPACES

It should be noted that the features mentioned in the previous chapter often reside within different spaces. A case that commonly occurs is that the features refer to a pixel or voxel. This happens for example in case of the first order statistics. First order statistics extracted from a window around a central pixel are calculated for each pixel, therefore, if e.g. 4 first order moments are calculated according to this procedure, each pixel or voxel has four features that refer to it. On the other hand, features can also refer to a selected or suspicious ROI. In case of the fractal dimension, for example, FD is calculated for the image gray values within the ROI or for the histogram of the image within the ROI. Therefore, it might not be possible to arbitrarily apply any feature family combination to train a plain classifier. Some techniques that could make this however possible are suggested in **??**. Below are mentioned shortly all spaces within each of the feature sets presented above reside.

1. **First Order Statistics**
   First order statistics are calculated per image element (voxel/pixel). Thus, each element is ascribed with k values and comparisons take place between pixels.

2. **Second order statistics**
   Second order statistics are calculated per ROI. Depending on the implementation, usually 28 GLCM (or GLRLM) features refer to a ROI.

3. **gradient features**
   All features mentioned in the gradients section (3.2) yield a different per voxel or pixel values. Thus, for example, if both directions Roberts' filters are applied on the image, the result will be two more images, hence each pixel will be ascribed with two more values.

4. **Multi-scale features**
   Wavelet features' data spaces are very interesting, as they can be divided in the subcategories. First, pyramidal decomposition is used and any kind of feature that is extracted from the original scale image can be extracted on all levels of the pyramid. This is not really a space itself, but in this case merely new data spaces are generated by this procedure. For example, an image of original size 512x512 pixels broken down in two levels, will yield a new image of 256x256. Features extracted for a ROI of original size, e.g. 24x24 will be also extracted for the low-resolution ROI of size 12x12. Thus, a new data space has occurred. DWT coefficients, on the other hand, yield a variable size 3-D vector at each deeper scale decomposition. This variable size can't be considered really as a per pixel feature space, therefore, it will be viewed as a per ROI feature set.

5. **Fractal features**
   Fractal features yield a per ROI result, as they depend on the texture of a region.

6. **Morphological features**
   Morphological features do not really comprise a set that is in either of these categories (per picture element or per ROI), as information is inferred just by the ROI shape. Functionally, however, in an implementation of a CAD system it could be considered as a per ROI feature.

A possible suggestion to integrate between per pixel and per ROI decision objectives, would be to estimate a ROI metric from decisions made for voxels or pixels and then subsequently apply this as a per ROI feature (e.g. first order statistics of a filter over a ROI). Another possible solution would be to assign ROI values to each pixel within a ROI. Thus, ROI features can be faced as per-pixel (or voxel) features. Finally, ROIs could be sliding windows, giving a unique value to each pixel. In our implementation, the last two suggestions are present.

## DIMENSIONALITY REDUCTION

Extraction of features usually leads to face the problem in a space that is way too high for the size of the datasets that are usually available. Dimensionality can be sometimes even in the order of 635 features [6], while the largest reported dataset in the examined literature is reported in [82], where a training set of 244 patients with an independent test set of 136 patients have been used. The use of datasets whose size fluctuates between 100 and 200 patients is not uncommon [37, 87], but in most of the cases, size of the datasets is usually in the order of 20 patients [29, 35]. In some cases, it is around or it could even consist of 5 patients [13].

Even if the dimensionality of 635 features is an extreme case, usually features that refer to texture form a space whose dimensionality is usually well above 100. Even for a dataset of 200 patients, this is way too big, due to the notorious "curse of dimensionality" [105]. According to this principle, as the dimension of a space increases, this space becomes more and more sparsely populated by the available samples. Therefore, in order to properly train a classifier in high dimensional space, more data samples are necessary. The relationship of the density of data points within a space depends exponentially to the number of dimensions. Therefore, the corresponding necessary increase in data points is exponentially dependent on the number of dimensions.

When talking about texture features, it is highly likely that features are not completely uncorrelated one with another. For example, features that result from the application of directional filters, might have a high degree of correlation between them. This results in information redundancy and it should be eliminated, in order to obtain the optimal number of dimensions. Several methods have been proposed in order to achieve this. They can mostly be separated in feature extraction and feature selection techniques. Below are briefly described some of the most important methods used in literature.

### FEATURE EXTRACTION

Feature extraction is about transforming the available features into new features that should be more descriptive than the original ones. In [36], authors use the Discrete Cosine Transform (DCT) to their original 30 D feature space and only consider the lowest 3 components. DCT is a continuous wavelet transform, so its principles are those described in sec. 3.3. This means that the 3 lowest components are used to describe the coarser characteristics, this coarseness though referring to the features and not the original image itself.

In [74] authors use a very popular technique, namely Principal Components Analysis (PCA). PCA works by trying to project the data from the original space to a lower dimensional manifold within the space and keep the transformed features. Its successful application requires data to be indeed sufficiently projectable into such a manifold without great loss of information [106].

Somewhat similar to PCA, in [107] authors introduce an embedding space related technique. An embedding space is a lower dimensional projection space of the original data points, where however a certain dissimilarity measure criterion has to be fulfilled, so that "distances" between data points retain their original mutual relationships as in the original high dimensional space. There are several popular embedding techniques that can be also used to visualize data relationships [108]. In [107] however it is recommended that classification is carried out in the embedding space. Dimensionality reduction is being achieved by projecting the original data to directions of the eigenvectors of the matrix formed by their mutual distances, where the metric used as distance has an exponential form and is for points $c$ and $d$, a feature vector $\mathbf{f}$ and $\omega_t$ the tumor class:

$$W(c,d) = e^{-||\hat{P}(c\rightarrow\omega_t|\mathbf{f}-\hat{P}(d\rightarrow\omega_t|\mathbf{f})||} \tag{4.1}$$

### FEATURE SELECTION

Feature selection, unlike feature extraction, is a technique where original features, considered to be highly informative are preserved. In [109], authors evaluate the features according to their ability to discriminate between cancerous-non cancerous regions. The metric to judge is the AUC. Subsequently, the calculate a

correlation matrix for all features. If two of them are highly (more than 0.8) correlated, they only keep the feature with better AUC performance.

In [42, 82], authors implement the mininal-redundancy-maximal-relevance (mRMR) criterion introduced in [110]. According to that crieterion, if the mutual information for two probabilistic variables $x, y$ is defined by

$$I(x;y) = \int\int p(x,y) log \frac{p(x,y)}{p(x)p(y)} dxdy \tag{4.2}$$

mRMR criterion works by trying to maximize $\Phi = D - R$ metric, where $D$ is relevance and $R$ redundancy and they are defined as follows:

$$maxD(S,c), D = \frac{1}{S}\sum_{x_i \in S} I(x_i;c) \tag{4.3}$$

$$minR(S,c), R = \frac{1}{|S|^2}\sum_{x_i,x_j \in S} I(x_i;x_j) \tag{4.4}$$

Where in the above equations, $S$ is the feature set comprising of features $\{x_1, x_2, \ldots, x_n\}$ and $c$ is the target class. In [17], after trying several feature selection techniques, the authors conclude that the most effective one is a feature selection scheme embedded on the classifier (recursive feature selection support vector machine, RFE-SVM). SVM classifier is trained for many possible feature sets. If the margin between two classifiers is small in the absence of a feature compared to the margins in the absence of other features, this one feature is excluded.

## CLASSIFIERS

As seen in table 5.1, the most commonly used classifiers are Support Vector Machines (SVMs) and Random Forests (RFs). As both of these classifiers are used in our experiments, a brief explanation of their principles is presented below. Other options include Neural Networks and AdaBoost. Neural networks, although a great option under other circumstances, strongly rely on a large size of a dataset with many independent samples. Our datasets' sizes do not allow for the use of such a method. Finally, AdaBoost is also a promising option. Basically being an ensemble classifier, as this method is covered adequately by Random Forests, this method will also not be applied.

### SVM

SVMs are really popular classifiers that have been developped by V. Vapnik and A. Chervonenkys [111]. Given a linearly separable labeled dataset, there can be several linear classifiers described by

$$y(\mathbf{x}) = \mathbf{w}^{\mathbf{T}}\phi(\mathbf{x}) + b \tag{4.5}$$

where $y$ is the assigned label, $(w)$ is a weight vector, $b$ is a bias vector and $\phi(\mathbf{x})$ is a transformation function of the original data space. SVM guarantees to find the solution that maximizes the *margin* of the classifier. The margin is defined as the minimum distance of a point of the available dataset from the decision hyperplane. Deriving from eq. 4.5, if true labels are defined as $t_n$ for a data point $n$ this can be found by maximizing ([105], eq. 7.3):

$$argmax_{\mathbf{w},b}\left\{\frac{1}{||\mathbf{w}||} min_n[t_n(\mathbf{w}^{\mathbf{T}}\phi(\mathbf{x}) + b)])\right\} \tag{4.6}$$

By scaling optimal $\mathbf{w}, b$ by a factor $\kappa$ so as to set $t_n(\mathbf{w}^{\mathbf{T}}\phi(\mathbf{x}) + b)]) = 1$, then for all data points holds:

$$t_n(\mathbf{w}^{\mathbf{T}}\phi(\mathbf{x}) + b)]) \geq 1 \tag{4.7}$$

Where the inequality holds, points are said to be inactive, whereas points for which the equality holds are active and the are the marginal points, or *support vectors*. An intuitive notion of this is presented in figure **??**. The

optimization continues just by considering the support vectors that are the most prone points in the dataset and can still define the most robust decision hyperplane. As a result, in the end, very few points are actually needed to draw a robust decision limit.

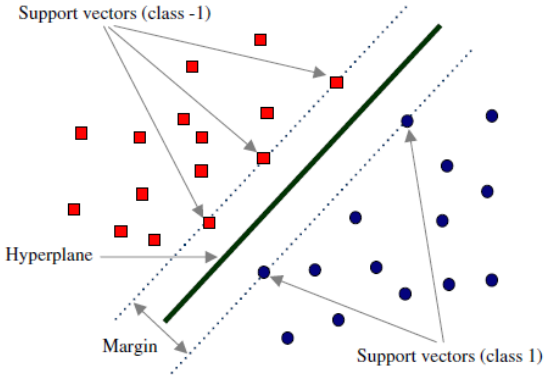



Figure 4.1: A demo of support vectors and margin in a 2D-2 class linearly separable problem

The two major assumption of linear separability can be bypassed by the transformation function $\phi(\mathbf{x})$, the full form of which is not necessary, as during optimization, it is not needed as it is. Instead, the *kernel* function is needed, which does not describe the full transform, instead it sets a rule of transforming the distance metric:

$$k(\mathbf{x},\mathbf{x}') = \phi(\mathbf{x})^{\mathbf{T}}\phi(\mathbf{x}') \tag{4.8}$$

Application of a kernel can transfer an originally non-linear problem to a linearly separable equivalent. Overlapping classes issue can also be adressed by allowing for an error during optimization.

Unfortunatelly, an analytic rule of transfering the problem to a linear domain is rarely known, therefore estimations or approximations have to be done. There are several popular kernel functions. Radial Basis Function (RBF) kernel is one of those. RBF kernel is defined as:

$$\mathbf{K}(\mathbf{x},\mathbf{x}') = e^{-\frac{||\mathbf{x}-\mathbf{x}'||^2}{2*\sigma^2}} \tag{4.9}$$

From the above description and the commonly available dataset sizes, it becomes clear that SVM is a good choice for this problem, as high-dimensional spaces with small datasets are a common case.

Nonetheless, we have to consider that SVM is very prone to support vectors whose dimmensions do not in the same scale. Moreover, online learning for SVM is not an option. Beacause of these last drawbacks, although SVM is tested in our experiments, the best option seems to be random forests, described below.

### RANDOM FORESTS

Random forests were introduced by L. Breiman in [112]. Random forests is an ensemble learning technique based on tree classifiers.

A decision tree is a classifier that splits an N-dimensional classification problem in the combination of N-one dimensional problems. Training consists on assigning a threshold on each feature and making a decision based on if that feature is above or below that threshold. Bootstrap aggregating consists in training a set of classifiers on different subsets of the dataset and choosing according to a voting scheme each time. There are several voting schemes, such as majority voting ("democracy", or the class with the most voters is assigned) or just the most confident classifier could be taken into account.

Random forests train a set of decision trees as classifiers (the forest). The novelty lies within the fact that bootstrapping this time does not consist in subsampling the dataset, but the features' space. By training on a random subset of features with replacement each time, again there is an ensemble selection of classifiers, whose result is combined.

Things that have motivated for the application of random forests include the fact that they do not overfit ([112]). Moreover, a decision tree is a classifier that does not need any data preprocessing and is completely

independent of scale between features (as each feature is treated independently). Finally, this method allows for an online learning model.

# 5

# Literature Review

All articles that have been retrieved can be distinguished in two major categories of publications, namely publications concerning:

1. Evaluation of specific biomarkers relevance and effectiveness
2. Attempts to build a Computer Aided Diagnosis System (CAD)

Some basic pilars of each of these approaches are presented in the summary tables 5.2 and 5.1, concerning categories 1 and 2 respectively. There were also some publications trying to assess a combined radiologist-CAD decision support system [44, 113], that do not fall in any of those categories, however, they could be very useful in clinical practice. Moreover, some publications whose results are presented vaguely are omitted. In order to avoid clutter, the following groupings of properties are introduced:

- **Preprocessing**
  - NF (Noise Filtering)
  - IBC (Inhomogeneity/Bias Correction)
  - INT (Intensity Normalization)
- **Features**
  Although we are only interested in radiomic features, it is necessary to take into account the entire feature set used at each study. Papers that only include functional pharmacokinetic or MRS features are not reported. The following groups are introduced:
  - C0 (Quantitative Map Intensity Features)
  - C1 (First Order Statistics)
  - C2 (Second Order Statistics including (GLCM, GLRLM, Haralick features))
  - C3 (Gradient Features, Including Sobel and Kirsch filters and RI-LPQ ([114]))
  - C4 (Wavelet-MultiScale Features)
  - C5 (Fractal Features)
  - C6 (Morphological Features)
- **Evaluation Report**
  When reporting results, usually atuhors report sensitivity, specificity and accuracy. Area Under the Curve (AUC) is another option that has the same meaning as accuracy. When AUC or accuracy measures are available, results are reported in terms of accuracy. In any other case, the metric is specified explicitly.

In favour of keeping these tables readable, some possibly useful informations are not included. Concerning the datasets, in some publications there is an independent evaluation dataset -although from within the same clinic-([82, 87]) while in most papers cross validation is performed [34, 74]. An important aspect to that is the number of folds for the cross validation. Also, some publications include in their datasets only scenes from

patients with confirmed histopathology [32, 115], while others also take into account images from healthy prostates [39]. Another potentially important aspect is what the ground truth is for each of the approaches. Most often, this is manually registered images from histopathological slices [25, 116], while also in other cases MRI-TRUS guided biopsies [117] or even manual MRI delineations are considered as ground truth [118]. Regarding the reported results, only the best one is always presented. However, there can be cases that results within a study strongly diverge, as the best results usually only apply for high GS values ([37]) or only apply to Central Gland (CG) or Peripheral Zone (PZ) Cap [119], while in some cases there are also reported different accuracies for various decision objectives that the CAD tries to resolve [75]. Finally, frequently several combinations of features' families or mpMRI modalities are examined, however, only the ones yielding the best results are presented. Notation "-" means that there is no implementation of this technique in this publication and "NA" means that it is not applicable. Column GS refers to the considered Gleason Score differentiation between tumorous and healthy, wherever this measure is reported. The remarkable issue presented below, is that all of these studies are only evaluated on data from a single clinic. This means that, even if a completely independent dataset was used, the images were acquired under the same scanners and protocols.

Table 5.1: suggested CADs reports

| Citation | Scanner | Data | GS | Preprocessing | T2W | DCE | DWI | Results | Evaluation method(pixels or prostates) | Independent Validation Dataset | Classifier | single protocol | Retrospective study | Clinical features | ground truth |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| [36] | 1.5 T | 10 | - | NF | C0, C1 | C0 | - | 0.8977 | per patient | No (cross validation) | Probabilistic Neural Networks | yes | Not mentioned | no | radiologist marked ROIS according to biopsy |
| [109] | 1.5 T | 45 out of 91 | 7 | IBC, INT | C0 | | C0 [9] | 0.91 | per ROI | No | Correlation Coefficients | yes | online | no | radiologist marked ROIS according to histology |
| [13] | 4 T [1] | 5 | - | IBC, INT | C1, C2, C3, C4 | - | - | 0.2 [2] | per pixel | No | AdaBoost | yes | yes | no | radiologist marked ROIS according to histology |
| [82] | 3 T | 280 | - | INT | C1, C2, C3 | - | C1, C2, C3 [10] | 0.85 | per ROI | yes | SVM and Willcoxon | yes | online | no | fused biopsy-MRI |
| [56] | 1.5 T [1] | 20 | - | NF, INT | C0 | C0 | C0 [9] | 0.84 [3]0.8 [4] | per pixel | No (cross validation) | SVM | yes | Not mentioned | no | radiologist marked ROIS according to histology |
| [17] | 3 T | 217 | - | - | C1, C2 | - | C1, C2 [9] | 0.92 | per ROI | No (cross validation) | T-test SVM, RFE-SVM, AdaBoost | yes | yes | no | radiologist marked ROIS according to histology |
| [28] [5] | 3 T | 7 | - | - | C0, C1, C2, C3 | C0 | C0, C1, C2, C3 [9] | 0.71 | per ROI | No (cross validation) | SVM, Random Forest | yes | yes | no | radiologist only |
| [37] [11] | 3 T | 177 | 6 | - | C1 | C1, C3 | C1 [9] | 0.92 [3] | per set of pixels | Not mentioned | LDA | yes | online | no | radiologist marked ROIS according to histology |
| [29] | 1. 5 T | 15 | - | INT | C0 | C0, C5 | C0, C5 [9] | 0.7 [3]0.8 [4] | per ROI | Not mentioned | C-Means | yes | yes | no | radiologist marked ROIS according to histology |
| [30] | NA | 27 | - | - | C2, C4, C5 | - | - | 0.85 [3]0.93 [4] | per ROI | No (cross validation) | SVM, AdaBoost | yes | Not mentioned | no | radiologist marked ROIS according to histology |
| [75] [6] | 3 T | 347 | 3 [7] | - | C0, C1, C3 | C0, C3 | C0, C1, C3 [9,0] | 0.83 | per voxel | No (cross validation) | GentleBoost, RandomForest | yes | yes | no | radiologist marked ROIS according to biopsy |
| [24] | 1.5 T | 30 | - | - | C0, C1, C2, C3 | C0, C1, C2, C3 | C0, C1, C2, C3 [9] | 0.89 | per ROI | No (cross validation) | SVM | yes | Not mentioned | no | Many radiologists marked ROIs |
| [32] | 1.5 T | 55 | 7 | INT | C5 | - | - | 0.9905 | per ROI | Not mentioned | T-test | yes | Not mentioned | no | fused biopsy-MRI |
| [31] | 1.5 T | 15 | 6 | INT | C1, C6 | - | - | 0.839 | per patient | No (cross validation) | SVM, FLD | yes | Not mentioned | no | radiologist marked ROIS according to biopsy |
| [42] [6] | 3 T | 31 | NA | IBC, INT | C0, C1, C2, C3 | C0, C1, C2, C3 | C0, C1, C2, C3 [9,0] | 0.76 | Not mentioned | Not mentioned | cascaded | yes | yes | no | fused digitized histology |
| [116] | 3 T | 12 | - | IBC, INT | C1, C2, C3 | C0 | - | 0.848 | per pixel | No (cross validation) | SVM | yes | Not mentioned | no | radiologist marked ROIS according to biopsy |
| [25] | 3 T | 6 | - | IBC, INT | C1, C2, C3 | C0 | - | 0.815 | per pixel | No (cross validation) | RandomForest | yes | Not mentioned | no | radiologist marked ROIS according to histology |
| [83] | 3 T | 26 | - | - | C0, C5 | - | C0 [9,0] | 0.883 | per patient | No (cross validation) | Random forest | yes | Not mentioned | no | radiologist marked ROIS according to histology |
| [120] | 3 T | 18 | - | - | C0, C1, C2 | - | - | 1 | per patient | No (cross validation) | SVM | yes | no | no | radiologist marked ROIS according to histology |
| [121] | 3 T | 42 | 7 | - | - | C0 [12] | - | 0.83 | per ROI | No (cross validation) | SVM | yes | yes | no | radiologist marked ROIS according to histology |
| [33] | 1.5 T | 11 | - | - | C0 | C0 | C0 [9] | 0.8938 | per set of pixels | Not mentioned | MRF (unsupervised) | yes | Not mentioned | no | radiologist marked ROIS according to histology |
| [122] | 1.5 T | 20 | - | NF, INT | C0 | C0 | C0 [9] | 0.8 | per ROI | No (cross validation) | SVM, RVM, MRF | yes | yes | no | radiologist marked ROIS according to digitized histology |

[1] Ex Vivo
[2] Positive Predictive Value
[3] Sensitivity
[4] Specificity
[5] includes MRS
[6] includes Proton Density Weighted Imaging (PDW)
[7] per zone (Central Gland (CG) or Peripheral Zone (PZ)
[8] features extracted from CDI maps
[9] features extracted from ADC maps
[10] features extracted from single b-value DWI
[11] includes C1 features on T1W
[12] includes several pharmacokinetic functional maps

When referring to feature sets' and biomarkers' evaluation, results are usually not reported in terms of accuracy as they can be totally different and usually there is a different study objective. Therefore, columns referring to reported accuracies cannot be present in the same manner as in table 5.1. In short, included references' objectives are described below.

In [35], authors conclude that there are significant quantitative differences in biomarkers between CG and PZ CaP. In [123] data consists of a dataset acquired by two different scanners, which is an interesting and seldom case. It is concluded that Haralick features on T2W and median on ADC maps have a significant correlation with biochemical recurrence after PZ radiotherapy, while Haralick features calculated on ADC maps yield no important correlation. In [39], the purpose is to find out what features are efficient in GS differentiation. Haralick entropy and inertia are positively and energy, correlation and homogeneity negative correlated to GS. In [117], the features that are deemed more efficient for d'Amicco clinical risk score evaluation are first order statistics on ADC maps. Authors in [124] come up to the conclusion that, within a broader set of examined features, gradient features are the most efficient in evaluation of Laser Inerstitial Thermal Therapy (LITT) outcome. In [125] it is estimated that ADC values contain useful information about GS score classification. In [57] it is concluded that the use of relative signal intensity (rSI) of very high-b DWI can be more informative than ADC maps and can lead to great results in conjunction with T2W images. Finally, the purpose of [18, 90, 119, 126] is to introduce a novel feature ensemble for automatic delineation between cancerous and healthy regions.

Table 5.2: Feature Sets Evaluation Studies

| Citation | Scanner | Data | Preprocessing | T2W | DCE | DWI |
|---|---|---|---|---|---|---|
| [35] | 3 T | 22 | IBC, INT | C1, C2, C4, C5 | - | - |
| [123] | 3 T | 74 | INT | C1, C2, C3 | - | C1, C2, C3 [3] |
| [39] | 3 T | 174 | - | C2 | - | C2 [3] |
| [124] | 1.5 T | 5 | INT | C2, C3 | - | C2, C3 [3] |
| [117] | 3 T | 48 | - | C0, C1 | - | - |
| [126] | 3 T | 5 | NF | C1, C2, C7 | - | C1, C2, C7 [3] |
| [18] | 3 T | 13 | - | C0, C1, C2, C6 | C0, C1, C2, C6 | C0, C1, C2, C6 [2,4] |
| [57] | 1.5 T | 17 | - | C0 | - | C0 |
| [119] | 3 T | 23 | INT, IBC | C0, C1, C2, C3, C5 | C0 | C0, C1, C2, C3, C5 [3] |
| [90] | 3 T | 48 | - | C0, C1 | C0 | C0, C1 [3] |
| [125] | 1.5 T | 57 | - | C0 | | |
| [34] | 3 T | NA [1] | - | C1, C2 | - | C1, C2 [2,3] |

[1] dataset size is given in ROIs and not number of patients. Size of the dataset consists of 5260 pixel windows for T2W and 1315 for DWI.
[2] features extracted from CDI maps
[3] features extracted from ADC maps
[4] features extracted from single b-value DWI
[2] computed on both CDI and ADC maps extracted from CHB-DWI

In table 5.1, one can realize that all currently available studies restrain to data obtained from a single clinic, namely under one protocol and scanner. Moreover, a completely independent dataset is seldom used.

# 6

# METHODS

## DATA

Our datasets are the TCIA ([127, 128]) and the PCMM datasets.

### TCIA DATASET

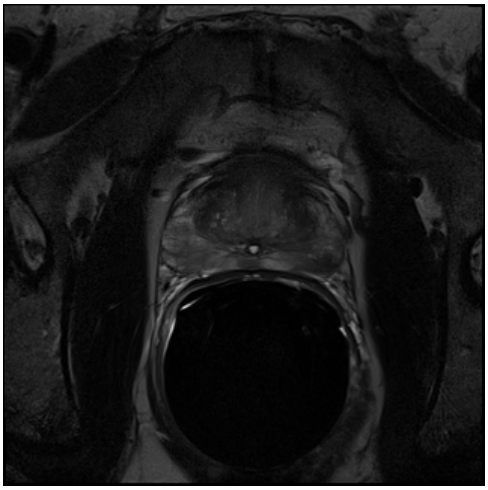
"TCIA Prostate fused-MRI-Pathology" dataset is a dataset from the Cancer Imaging Archive (TCIA), an open collection of cancer images datasets. The specific dataset ([128]) includes multi-parametric MRI (mpMRI) for CaP patients. 28 patients are included, however labels are provided for only 15 of them. Labels refer to Gleason Score 6 and above (positive) or below (negative). Prostates about which labels are available have been extracted and sliced alongside an axis alongside the urethra every 3-4 mm. Each of these slices was further sliced in quadrants and after fixation in formalin has been stained with hematoxylin and eosin. Annotations based on staining have been made by an expert pathologist. Subsequently, after resected prostate slices' reconstruction by a software package, correspondence between the extracted prostates and the T2 MRI images have been made by an expert pathologist and an expert radiologist based on major anatomic landmarks (such as the urethra). These landmarks have been fed to a thin-plate spline registration algorithm to complete the registration between T2-MRI images (fixed image) and pathology images (moving image). The computed transformation is then applied to register the labels. Detailed information about this process as also any further information about this dataset can be found in [129]. The provided labels for this dataset concern Gleason Scores of 6 and above.

The images of TCIA are usually of consistently high quality (0.4-0.5 mm resolution). Endorectal coils have been used and the contrast is relatively high. An example of two randomly picked TCIA axial T2 image slices is shown in fig. 6.1a, 6.1b.

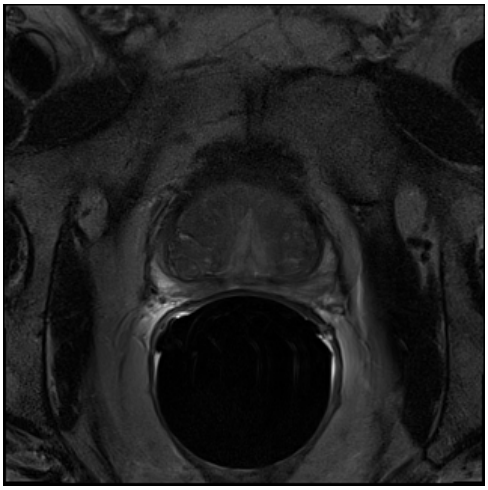
### PCMM DATASET

The PCMM dataset contains mpMRI scans from 3 hospitals. These are the Erasmus MC, NKI and RadboudUMC. Moreover, macroslides of the extirpated prostates and the histology of each macroslide are available for some of the Erasmus MC cases. The PCMM dataset is developed by a multidisciplinary consortium composed of academic hospitals to address clinical needs in the field of prostate cancer. PCMM dataset primarily consists of 183 folders. Nonetheless, this is not the actual number of patients, as several duplicate patients with different patient IDs are observed. Moreover, due to incomplete information for many of these cases, most of the patients had to be excluded. The most common reason for this is the absence of a histology report. Further reasons include absence of metadata such as sequence and spacing description of the images.

In detail, by the time of data processing, 23 pathology reports were available. As the constructed software relies on patient identification based on the patient ID dicom tag, a missmatch in dicom tags could result in a patient being excluded. Patient ID was chosen as a safe key to avoid duplicate entries. However, it is recommended that in future versions this is switched to DOI(Digital Object Identifier). Pathology reports are identified using the patient name. In most cases, the patient name dicom tag is the same or there is a clear connection between



(a) slice of patient TCIA.aaa0044 axial T2



(b) slice of patient TCIA.aaa0064 axial T2

those two. However, for 8 out of the 23 patients a match between dicom tags and patient names was not achieved and the images could not be retrieved. Therefore, another 8 patients had to be excluded. This resulted in a remaining dataset of 15 patients.

In order to correspond these images to labels, the pipeline described bellow was applied. Prostates were extracted and sliced every 3-4 mm along the urethra axis, resulting in macroslides that were photographed. Pathological annotations were made on thiner slices, sliced from the top of the macroslides. These slices were manually reconstructed to resemble the whole single slice. Tumor contours that were originally acquired by histology reports on the thinner slices were then manually transferred on the macroslide reconstruction. The contours were drawn on the extracted prostate macro photos. By manually stacking all these slices together along a vertical axis, it is possible to get a reconstruction of the prostatic volume. Handdrawn ROIs should now represent the volumes of the lesions. For further details on this method, refer to Appendix B.

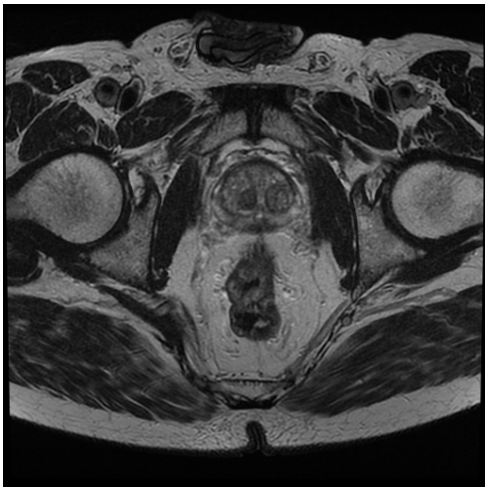
The problem that this method sets is, that, after extraction the prostate loses the shape that it had within the body, making a direct correspondence of the annotated ROIs to the MRI images very difficult. However, if a 3 dimmensional B-Spline registration is applied on a mask referring to the volume reconstructed from the macros to the mask of the axial T2 image, the resulting transformation could be applied to the ROIs of the reconstructed volume. Then they should correspond to the correct regions within the image. For details about all the preprocessing of the PCMM dataset, the reader can refer to Appendix B.

Most of the images of the PCMM dataset are of good quality. However, note that, unlike TCIA dataset, images of the PCMM dataset were obtained without an endorectal coil. Moreover, phantoms (see fig. 6.2b) or poorer resolution in the order of 0.7 mm (see fig. 6.2a) are not uncommon. It is furthermore interesting to notice in these pictures, how the range of gray value varies, even for images within the same clinic. Beyond anatomic information, differences in resolution are expected to affect the performance of the model, as the training is based on pixels.

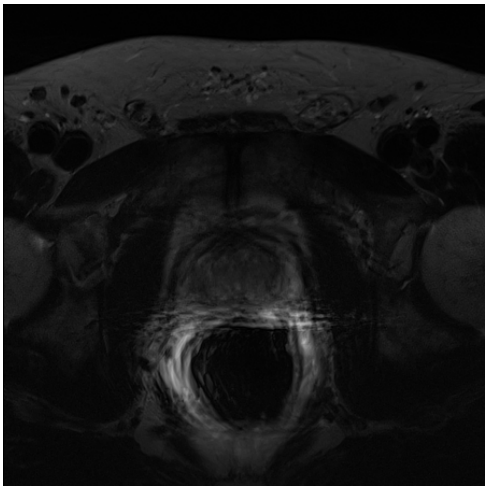
The accuracy of the process according to which the labels were assigned is questionable, as there is no previous report of a similar attempt in literature, where usually labelling is being carried out by radiologists directly on the T2 images. This means that a region which should be marked as a lesion on the T2 image could actually be a healthy region and vice versa. As a result, poorer results should be expected for this dataset. The provided labels for this dataset concern Gleason Scores of 7 and above. Note that, in case of the TCIA dataset the Gleason Score was 6 and above. In case where a model is trained on one dataset and applied on the other, this is expected to further deter the results.

## UTILIZED FEATURES

All features have been calculated for the axial T2 sequences. The features are extracted on a per-pixel level (and not per voxel). The reason for this option was the fear that the vastly different resolution along the 3rd axis might deter the model training. Features are only calculated over a region defined by a rectangle circumscribed over each slice of the prostate for computational efficiency. They can be visualized over the original image,



(a) slice of patient PCMM.s9 axial T2 (with low resolution)



(b) slice of patient PCMM.s30 axial T2 (presenting a phantom)

whereas the rest of space is filled by black for visualization purposes. Wherever a library is not mentioned, a custom implementation is implied. A more scientifically detailed explanation of those features can be found in chapter 3.

### FIRST ORDER STATISTICS (C1)

First order statistical features (C1-category -1) that have been implemented include:

- mean (*numpy* implementation)
- standard deviation (*numpy* implementation)
- skewness (*numpy* implementation)
- kurtosis (*numpy* implementation)
- median (*numpy* implementation)
- average deviation,described in [13]
- entropy, described in [34]
- uniformity, described in [34]
- variance, described in [82]

All category 1 features are calculated within a sliding window of size 9x9 pixels. This size has been chosen as it is a typical window size in literature (7 to 9 pixels are the most common sliding window sizes). Depending on the resolution of the scan, it can correspond to a square of physical dimensions 3 to 5 mm, which is typically adequate to capture a whole lesion or a big part of it.

### SECOND ORDER STATISTICS (C2)

Second order statistics include:

- 10 features for Grey Level Run Length Matrix (GLRLM) features, described in [92]. In order to avoid a sparse histogram and highlight the importance of contrast, for these features a 5 level gray value image was considered.
- 26 features for Grey Level Cooccurence Matrix (GLCM) features, described in [91]. For those features, *mahotas* ([130]) python library has been used. As implementation of the 14th feature proved to be buggy and unstable, 14th feature was finally disregarded. In order to avoid a sparse histogram and highlight the importance of contrast, for these features a 5 level gray value image was considered.

- A histogram of Local Binary Patterns (LBP), described in [94]. Again, *mahotas* libray has been used for the estimation of these features. A radius of 1 and 8 points have been considered. Implementation of radius of 2 for 16 points is also available and can be switched on.
- A histogram of Local Directional Derivative Patterns (LDDP), described in [82]. Radii 1 and 2 for 8 points have been considered. Implementation of radii of 2 and 4 pixels for 16 points is also available and can be switched on.

As category 2 features are based on the gray values over an entire region of interest, two approaches have been examined:

- In this case, ROIs over which to calculate the features were defined by the ground truth. As *mahotas* library (or any library available) does not cover the case of support regions of arbitrary shape, the method to apply this was to consider a circumscribed rectangle around each ROI marked as positive. As a slice might contain multiple ROIs, the remaining healthy space was then partitioned in a set of rectangles. The boundaries of those were defined by the boundaries of the rectangles corresponding to positive ROIs extended until the end of the image and the boundaries of the image itself. This can be formulated as follows:

  If $N$ contours described by a parameter $t_i(x_i, y_i)$, $i \in [1, N]$ respectively lie within a rectangle bounded by $x \in [x^0, x^1]$ and $y \in [y^0, y^1]$, then for each $x_i, y_i$ holds $x_i \in [x_i^0, x_i^1]$ and $y_i \in [y_i^0, y_i^1]$, where $x_i^0, x_i^1 \in [x^0, x^1]$ and $y_i^0, y_i^1 \in [y^0, y^1]$. Then the ROIs over which the features are calculated are defined by the intersections of the lines defined by

  $$x = x^0, x_1^0, \ldots, x^1$$
  $$y = y^0, y_1^0, \ldots, y^1$$

  An example of this can be seen in fig. 6.3c where different values of a specific feature correspond to different regions of interest. In this way, the partitioning of the specific prostate slice in regions of interest can be distinguished.

  Thereafter, the values corresponding to each feature were assigned to all the pixels within the ROI, in order to proceed in a pixel level classification.

  Downsides of this method are that, first, it would not support a fully automated CaP detection approach as suspicious ROIs' marking by a radiologist would be a prerequisite. Moreover, feature values were found to have a dependence on the region size (the smaller the region the smaller the value). This would be expected, as features described above correspond to histogram bin counts. Normalization by the ROI size (area in pixels) was tried, but as many feature values are small (in the order growth of 10e-2), arithmetic underflow is very often. Although in average the size of regions corresponding to positive ROIs and ROIs corresponding to the healthy space partitions are comparable, it would not be safe to consider this approach. However, classification results by this approach are presented in the last section.

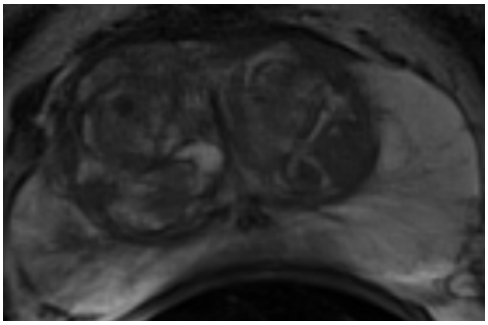
  As a suggestion for future research, implementation of a non-compact arbitrary shape support region and area independent feature extraction routine would be an interesting case of study. Although fully automated detection would not be supported, a combined radiologist-CAD approach with potentially great results would be possible.

- A 9x9 sliding window based approach was also examined. This is expected to be more accurate, as its results' magnitude does not differ due to differences in ROI area (it is constantly 81 pixels). Furthermore, it makes a fully automated detection effort possible. An example of a figure calculated in this manner can be seen in fig. 6.3e, in contrast to the same feature calculated according to the previously mentioned approach shown in 6.3c. A 5 pixel wide stripe around the image is being disregarded during classification in order to avoid the effect of edge artifacts that might be present in this stripe.

### GRADIENT FILTERS (C3)

Gradient filters include:

- Kirsch kernels
- Roberts kernels
- Sobel filters (implemented by library *scipy* [131, 132])



(a) axial T2 image

(b) marked lesions (ROIs)

(c) lbp(1,8) histogram 9th bin for ROI based partitioned space

(d) circumscribed rectangles around original ROIs

(e) lbp(1,8) histogram 9th bin calculated by 9x9 pixels sliding window

Figure 6.3: axial T2 (14th slice) derived images for patient TCIA.aaa0054

- Gradients (implemented by library *numpy* [133])
- Real part of Gabor filters (implemented by *scikit-image* [134]) for scales ranging between 2 and 8 mm. The reason for choosing these scales is described in [75].
- Gaussian filters (implemented by *scikit-image*) for scales ranging linearly between 2 and 8 mm.
- Li dot and line filters (the use of which is reported by [75] and explained in [88]) with scales ranging linearly between 2 and 8 mm.

### WAVELET COEFFICIENTS (C4)

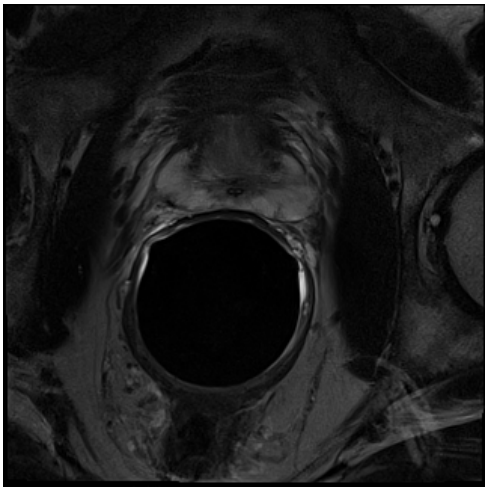
Wavelet coefficients include:

- Haar wavelet coefficients (implemented by *pywavelet*)
- Daubechies wavelet coefficients (implemented by *pywavelet*)

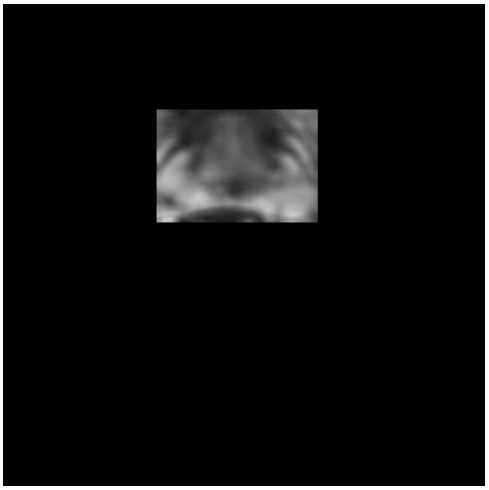
In order to incorporate lower decomposition levels in the pixels, *scikit-image* zoom function was used for the lower detail decomposition level in order to create a replica of an image equal to the original size. In this way, wavelet coefficients can be assigned to each pixel within the image. This means that if an original image is 512x512 pixels, the first decomposition level will consist of 3 256x256 images. By zooming in with a factor of 2,

3 512x512 images that can fit on the original image are obtained. Therefore, only one decomposition level was used.

Note that the details of the zoomed image are coarser. This might increase sensitivity to bigger lesions. As T2 image absolute gray values or gray value based features are most likely not an informing feature, this approach is not expected to greatly contribute to the results. However, it might be very useful for obtaining informations from functional MRI images, such as ADC maps. An example of Daubechies LH coefficients visualized at the first decomposition level over the prostate of TCIA patient aaa0072 is presented in fig. 6.4b. Note that the black surrounding is not actually feature computed value, but values assingned for visualization. The region over which the feature values are calculated is solely confined by the prostate-circumscribed recangle.



(a) slice 13 of TCIA.aaa072 axial T2 image



(b) Daubechies LH wavelet coefficients for $1^{st}$ decomposition level of slice 13 of patient TCIA.aaa0072

### FRACTAL FEATURES (C5)

Fractal features include:

- Image fractal dimension (proposed in [32])
- Histogram fractal dimension (proposed in [32])

For this category, the same methods as described in section 6.2.2 regarding a ROI or sliding window based approach have been applied. However, fractal dimension is only defined for a rectangular support region, therefore this method could not be extended to an arbitrary shape ROI generalization either way. Unfortunately, as fractal dimension of our images fluctuates between 1.4 and 1.6 it is not possible to visualize as in case of section 6.2.2.

## CLASSIFICATION

Decision objective for classification is to assign a pixel a hard label, wether it is considered above or below Gleason score 7. Data space consists of 198-dimensional vectors referring to pixels of all prostates combined together in a single set. Classifiers that have been examined include a Random Forest(RF) and a Radial Basis Function (RBF) kernel Support Vector Machine (SVM). While for the Random Forest no preprocessing is required, for the SVM classifier the following preprocessing steps had to be taken:

1. Scale features to zero mean and unit variance as SVM is strongly dependend on features' scales. A robust scaler has been used, that disregards outliers by using data only between the 25th and the 75th percentile (Robust Scaler of the library *scikit-learn* [135]).
2. Conduct PCA analysis on the new scaled feature set (implementation by *scikit-learn*). Eigenvalues growth is stored in output text files and in all of the cases keeping the 15 first features is deemed as representative enough.

3. Apply PCA by keeping the first 15 features. In case of an independent test set save the transform and apply the same transform on the test set (implementation by *scikit-learn*).

Although results of SVM and RF are comparable, it is deemed as preferable to use a Random Forest as feature scaling might result in loss of potentially useful information. Generally, minimal preprocessing would be preferred to keep original information as intact as possible.

Finally, class balance is achieved by random undersampling of the majority class. Random undersampling has been chosen as it is stated in [136] that more advanced undersampling techniques yield no significantly better results. Moreover, a minority class oversampling techinique is thought as potentially dangerous as the nature of our data requires minimal assumptions and moving in a data space as closely to the original as possible. Therefore, as results as examined in terms of accuracy, 0.5 is the worst case.

## EMPROST 0.1 SOFTWARE DESCRIPTION

In order to calculate the features that we want and evaluate classification approaches, emPROST 0.1 has been developed in python 2.7. It has been tested on Windows 7 and Ubuntu 16.04. It works in two stages:

1. Call feature calculation routine before taking any further steps.

   - **Inputs** This call accepts as input the root directory under which the dataset is stored and the root directory under which prostate segmentation information and label information is stored. Additionally, a working directory has to be provided. Optionally, one can also choose which features to calculate (between families 1,2,3,4 and 5), whether to use the entire prostate segmentation or simply the peripheral zone and the method according to which the features will be calculated (sliding window or Region of Interest based, described in 6.2.2).

   - **Outputs** Inside the working directory a folder for each patient is created. This folder is named under the patient-ID to whom it refers. It contains segmentation and labeling information as well as the T2 image of this patient. Moreover, it contains all the features that refer to this patient to a single file (.npy file). Several text files necessary for retrieving the features by name are also stored. Inside the working directory itself, some text files referring to a directory mapping of the dataset are created. A progress file is also created and updated during runtime. The last one makes possible to pause and resume feature calculation which might be very computationally intensive. Execution can be reset or resumed to any required stage just by modifying or deleting the corresponding files.

2. Classify and evaluate.

   - **Inputs** Inputs should be classification method and working directory (which reflects the way that features have been calculated). Optionally, a switch that would indicate which features to consider can be provided. An input of multiple working directories is possible, in which case independent test set evaluation or fused dataset cross-validation is possible. Details such as cross validation folds can be editted in the source code. Instead of classification, also principal components analysis (PCA) could be applied with the same arguments. Finally, a brief experiment description should be provided, after which the corresponding output files will be named.

   - **Outputs** In case of classification text files with validation accuracies are saved in the working directory. In case of PCA analysis the size of the eigenvalues is saved in a text file. The name of the files depends on the name given as input in the previous step.

3. Additional functionalities
   On top of these main tasks, this software can also retrieve and save in itk image format any feature given by name for any patient for reviewing purposes. Moreover, massive preprocessing tasks such as image size reduction can be applied to the entire dataset. In case that a SimpleElastix [137] Simple-ITK built is present on the machine, also fine tuned BSpline registration specifically required for preprocessing during label registration is possible.

The full documentation of *emPROST* can be found in Appendix A.

## EXPERIMENTS

The experiments that have been ran, concern the following cases:

- distinction between peripheral zone and whole prostate
- distinction between fully automatic and combined approach (see sec. 6.2.2)
- distinction between Random Forest and Support Vector Machine classifiers
- distinction between C2 and C5 features only versus the entire feature set

The first step would be to see what can be achieved by extracting features from T2 and doing a cross validation. Therefore, the first two experiments concern cross validation on the same datasets. Thereafter, training models on PCMM (TCIA) and applying on TCIA (PCMM) shall show if it is possible to apply a model trained on a dataset on a completely new one. A further distinction is between whole prostate and peripheral zone. As the peripheral zone is more homogeneous than the central gland, better performance is expected. Finally, the differences in performance between a sliding window and a ROI based estimation of C2 and C5 features is examined.

In cases where a cross validation on the same dataset was performed, the cross validation is applied on a per-patient basis. This means that for a dataset cosisting of images of N patients, at each validation iteration, the model is trained on N-1 images and tested on the last one. This is to avoid an overestimation of the achieved accuracies, as training and testing on pixels of the same image would yield results better than they really are. Concering data of the PCMM dataset, it is noteworthy that 4 out of totally 15 patients appear to not have any positive labels. Of course, as we know that they do, this paradox can be attributed to the label assignment process described in Appendix B. Therefore, the labels assigned to these patients are definitelly wrong and they will not be considered. This leaves us with only 11 patients from the PCMM dataset. A conclusion from these mistakes is, initially, the need for development of a rigid protocol for assigning the labels. Moreover, we can observe that there is a big proportion of the available part of the PCMM dataset's labeling, which is definitelly wrong. This could raise questions about the labeling of the rest of the PCMM dataset. The questionable quality of the labeling of our data could detrimentally affect the quality of the models evaluated on these data.

The four tables in which results are reported distinct between:

- training on TCIA - testing on TCIA (cross validation) (see 7.1)
- training on PCMM - testing on PCMM (cross validation) (see 7.2)
- training on TCIA - testing on PCMM (independent dataset) (see 7.3)
- training on PCMM - testing on TCIA (independent dataset) (see 7.4)

# 7

# RESULTS

## CLASSIFICATION PERFORMANCE

The values reported throughout are classifier mean accuracy (abbreviated as MA) and, wherever applicable, variance between cross validation folds (abbreviated as var).

Table 7.1: Per patient cross validation results on TCIA. RF MA stands for Random Forest cross validation mean accuracy (when applicable), RF A var is Random Forest Accuracy variance and the same hold for SVM which stands for Support Vector Machine.

| examined case | RF MA | RF A var | SVM MA | SVM A var |
|---|---|---|---|---|
| all features, whole prostate, semi automatic | 0.694 | 0.017 | 0.643 | 0.002 |
| all features, peripheral zone, semi automatic | 0.623 | 0.029 | 0.671 | 0.006 |
| C2, C5 features, whole prostate, semi automatic | 0.668 | 0.02 | 0.647 | 0.004 |
| C2, C5 features, peripheral zone, semi automatic | 0.674 | 0.023 | 0.66 | 0.009 |
| all features, whole prostate, fully automatic | 0.558 | 0.002 | 0.555 | 0.003 |
| all features, peripheral zone, fully automatic | 0.544 | 0.003 | 0.543 | 0.002 |
| C2, C5 features, whole prostate, fully automatic | 0.559 | 0.002 | 0.559 | 0.003 |
| C2, C5 features, peripheral zone, fully automatic | 0.542 | 0.003 | 0.538 | 0.002 |

Table 7.2: Per patient cross validation results on PCMM. RF MA stands for Random Forest cross validation mean accuracy (when applicable), RF A var is Random Forest Accuracy variance and the same hold for SVM which stands for Support Vector Machine.

| examined case | RF MA | RF A var | SVM MA | SVM A var |
|---|---|---|---|---|
| all features, whole prostate, semi automatic | 0.5 | 0.004 | 0.485 | 0.006 |
| all features, peripheral zone, semi automatic | 0.52 | 0.002 | 0.508 | 0.002 |
| C2, C5 features, whole prostate, semi automatic | 0.5 | 0.005 | 0.493 | 0.006 |
| C2, C5 features, peripheral zone, semi automatic | 0.51 | 0.002 | 0.518 | 0.002 |
| all features, whole prostate, fully automatic | 0.53 | 0.004 | 0.51 | 0.002 |
| all features, peripheral zone, fully automatic | 0.508 | 0.0 | 0.516 | 0.0 |
| C2, C5 features, whole prostate, fully automatic | 0.52 | 0.004 | 0.514 | 0.002 |
| C2, C5 features, peripheral zone, fully automatic | 0.518 | 0.0 | 0.509 | 0.0 |

## FEATURE SIGNIFICANCE

As the Random Forest classifier evaluates relative feature importance during the intrinsic feature bagging procedure, a report of the relative significance of the computed features is provided below. Note that Random Forest feature importance evaluation has the following drawbak: when some of the features are correlated, information is only drawn by random feature within this feature selection. This will yield an importance index which will be higher for this specific feature, whereas the rest of the features in this selection will be underestimated. In our case, where many features are correlated within subsets (for example the GLCM features) only

Table 7.3: Independent dataset validation-training on TCIA, testing on PCMM. RF MA stands for Random Forest cross validation mean accuracy (when applicable), RF A var is Random Forest Accuracy variance and the same hold for SVM which stands for Support Vector Machine.

| examined case | RF MA | RF A var | SVM MA | SVM A var |
|---|---|---|---|---|
| all features, whole prostate, semi automatic | 0.469 | - | 0.548 | - |
| all features, peripheral zone, semi automatic | 0.454 | - | 0.514 | - |
| C2, C5 features, whole prostate, semi automatic | 0.469 | - | 0.553 | - |
| C2, C5 features, peripheral zone, semi automatic | 0.39 | - | 0.506 | - |
| all features, whole prostate, fully automatic | 0.503 | - | 0.509 | - |
| all features, peripheral zone, fully automatic | 0.507 | - | 0.522 | - |
| C2, C5 features, whole prostate, fully automatic | 0.502 | - | 0.509 | - |
| C2, C5 features, peripheral zone, fully automatic | 0.509 | - | 0.522 | - |

Table 7.4: Independent dataset validation-training on PCMM, testing on TCIA. RF MA stands for Random Forest cross validation mean accuracy (when applicable), RF A var is Random Forest Accuracy variance and the same hold for SVM which stands for Support Vector Machine.

| examined case | RF MA | RF A var | SVM MA | SVM A var |
|---|---|---|---|---|
| all features, whole prostate, semi automatic | 0.512 | - | 0.52 | - |
| all features, peripheral zone, semi automatic | 0.5 | - | 0.453 | - |
| C2, C5 features, whole prostate, semi automatic | 0.513 | - | 0.523 | - |
| C2, C5 features, peripheral zone, semi automatic | 0.515 | - | 0.459 | - |
| all features, whole prostate, fully automatic | 0.508 | - | 0.504 | - |
| all features, peripheral zone, fully automatic | 0.469 | - | 0.495 | - |
| C2, C5 features, whole prostate, fully automatic | 0.508 | - | 0.503 | - |
| C2, C5 features, peripheral zone, fully automatic | 0.48 | - | 0.49 | - |

one of them will be assigned a high performance value. Nonetheless, this evaluation shall provide an adequate insight concerining wich feature sets are the most important.

The mean value of the significance for all relevant experiments is presented in the tables below, whereas box plots of the same values accross all experiments can be found in Appendix C. As feature performance is expected to vary between cases where a ROI based or a sliding window approach is used, performance assessment is split in the following two cases:

- a semi-automated ROI based approach and
- a sliding window approach.

Feature names are as extracted from the *emPROST* software. They are briefly explained below:

- *mean, standard deviation, skewness, kurtosis, median1, median2, median3, lbp like variance, entropy* are the first order statistics presented in section 3.1.1. *median1, median2, median3* are described in equation 3.3. Entropy is defined in 3.6. *lbp like variance* is defined in 3.9.
- *lbp R**x** P**y** **z*** is the $z^{th}$ bin of the local binary patterns histogram for radius $x$ pixels and $y$ points. LBPs are defined in 3.14.
- *lddpp R1**x1** R2**x2** P1**y1** P2**y2** **z*** is the $z^{th}$ bin of the local directional derivative patterns histogram for inner radius $x1$ pixels, outer radius $x2$ pixels, $y1$ points along the inner radius and $y2$ points along the outer radius. LDDPs are defined in 3.15.
- *Haralick mean **x**, Haralick range **x**, GLRLM mean **x**, GRLRM **x*** refer to the $x^t h$ Haralick or Gray Level Run Length Matrix feature mean or range along 4 directions (0, 45, 90, 135) respectively (see section 3.1.2).
- *Kirsch kernel rotation **x**, Roberts kernel rotation **x*** refer to Krisch and Roberts kernels respectively, defined in 3.10 and 3.10. *Sobel filter* and *unoriented gradient* are self explaining.
- For *Gabor, Li* and Gaussian filters, scales are 8 exponentially increasing scales that consist of different amounts of pixels, depending on the resolution, so that a range of 2-8mm can be captured. This approach has turned out to be efficient, as many prostate lesions are on this scale (see [75]). Li filters are

defined in [88]. *li dot* refers to a blob sensitive filter and *li line* to a line sensitive one.

- *Daubechies, Haar* refer to Daubechies or Haar wavelet coefficients for the first level of decomposition respectively.
- *Image FD, Histogramm FD* refer to image fractal dimension or histogram fractal dimension respectively. Fractal dimensions are treated in 3.4.

Table 7.5: Feature significance-ROI based approach

| Feature Name | Feature Significance |
|---|---|
| Daubechies LH | 0.0111625344974 |
| Gabor scale2 direction 135 | 0.0106361723195 |
| Gabor scale5 direction 90 | 0.0104846094792 |
| Gabor scale0 direction 90 | 0.0101294485862 |
| lbp R1 P8 28 | 0.0101245405794 |
| lddp R1 1 R2 2 P8 13 | 0.0101076422316 |
| lddp R1 1 R2 2 P8 5 | 0.00998145627495 |
| Gabor scale1 direction 0 | 0.00991327061626 |
| lddp R1 1 R2 2 P8 34 | 0.00971352114923 |
| Roberts kernel rotation 1 | 0.00965891378378 |
| median3 | 0.00960929039081 |
| Unoriented gradients axis 1 | 0.00952792739766 |
| Gabor scale3 direction 0 | 0.00944506887119 |
| lddp R1 1 R2 2 P8 18 | 0.0090883611321 |
| Gabor scale4 direction 135 | 0.00899838484282 |
| lddp R1 1 R2 2 P8 9 | 0.00894898546261 |
| Li filter lines scale 5 | 0.00882438865607 |
| lbp R1 P8 26 | 0.00880830241525 |
| lddp R1 1 R2 2 P8 15 | 0.00878758474599 |
| Haralick range 7 | 0.0087762355016 |
| lddp R1 1 R2 2 P8 11 | 0.00871104263822 |
| lddp R1 1 R2 2 P8 7 | 0.00866870953446 |
| Li filter lines scale 2 | 0.00852636696168 |
| Haralick range 1 | 0.00848985606136 |
| lbp R1 P8 25 | 0.00841718867053 |
| lbp R1 P8 10 | 0.00840835678381 |
| lbp R1 P8 19 | 0.008403659206 |
| Haralick range 2 | 0.00828366264256 |
| lddp R1 1 R2 2 P8 32 | 0.00809009634422 |
| lbp R1 P8 12 | 0.00805314584668 |
| lbp R1 P8 16 | 0.00797042462509 |
| Gauss scale 8 | 0.00786571163636 |
| lbp R1 P8 5 | 0.00786496582883 |
| Gabor scale7 direction 135 | 0.00783977969809 |
| kurtosis | 0.00781646753139 |
| lbp R1 P8 35 | 0.00767174439179 |
| Haar LL | 0.00762253459147 |
| Li filter dots scale 7 | 0.0076013568964 |
| Haralick mean 8 | 0.00751275141405 |
| Haralick mean 4 | 0.00737502681607 |
| lbp R1 P8 33 | 0.00737029056761 |
| image FD | 0.00736130755681 |
| lbp R1 P8 7 | 0.00735032279972 |
| Haralick mean 10 | 0.00729795558321 |
| Kirsch kernel rotation 8 | 0.00726000134909 |
| GLRLM range 2 | 0.00721265053985 |
| Kirsch kernel rotation 5 | 0.00715662663707 |
| lbp R1 P8 30 | 0.00711622606455 |
| Gabor scale7 direction 90 | 0.00711309536202 |
| lddp R1 1 R2 2 P8 25 | 0.00710592308353 |
| Gabor scale5 direction 135 | 0.00702595787371 |
| skewness | 0.00701822445277 |
| Li filter dots scale 5 | 0.00698538767027 |
| lbp R1 P8 9 | 0.00697055854152 |
| lddp R1 1 R2 2 P8 23 | 0.00694056325096 |
| Gabor scale4 direction 45 | 0.00688815047626 |
| GLRLM mean 2 | 0.00685450802234 |
| Haralick mean 1 | 0.00685264592299 |
| lddp R1 1 R2 2 P8 21 | 0.0068331530652 |
| GLRLM mean 1 | 0.00676007990377 |
| Kirsch kernel rotation 2 | 0.00673697579201 |
| Gabor scale6 direction 90 | 0.00672906840517 |
| lbp R1 P8 36 | 0.00665860280142 |
| lbp R1 P8 2 | 0.00663389245351 |
| Haar HL | 0.00661753436483 |

| Feature Name | Feature Significance |
|---|---|
| lbp R1 P8 20 | 0.00653785339392 |
| GLRLM range 5 | 0.00649003714783 |
| Li filter lines scale 6 | 0.00643601462925 |
| Haralick mean 2 | 0.00639260260701 |
| Gabor scale4 direction 90 | 0.00639071437446 |
| Gabor scale1 direction 45 | 0.00634377827786 |
| U | 0.00633862098842 |
| Haralick range 5 | 0.00615410859193 |
| lddp R1 1 R2 2 P8 17 | 0.0061187973664 |
| lbp R1 P8 24 | 0.00604573494365 |
| lddp R1 1 R2 2 P8 1 | 0.00602122353705 |
| Gauss scale 5 | 0.0059661994441 |
| Gauss scale 1 | 0.00591906561594 |
| GLRLM mean 5 | 0.00563491866985 |
| Haralick range 12 | 0.00555015815733 |
| Gabor scale5 direction 45 | 0.00550795448569 |
| Haralick range 8 | 0.00546622646791 |
| lddp R1 1 R2 2 P8 3 | 0.00545738623225 |
| Daubechies HH | 0.00544728661865 |
| Haralick mean 7 | 0.00539613280865 |
| Gabor scale6 direction 0 | 0.00537608713338 |
| Gabor scale4 direction 0 | 0.00530970381446 |
| Kirsch kernel rotation 7 | 0.00528809281863 |
| Li filter lines scale 3 | 0.0052396528765 |
| Li filter dots scale 1 | 0.00521229925674 |
| lddp R1 1 R2 2 P8 27 | 0.00520656110058 |
| lddp R1 1 R2 2 P8 16 | 0.00519343989601 |
| Gauss scale 6 | 0.0051899298955 |
| Daubechies HL | 0.00515036880209 |
| lddp R1 1 R2 2 P8 29 | 0.00510642216387 |
| Gabor scale0 direction 135 | 0.00510231353539 |
| Haralick range 11 | 0.00505190626336 |
| Haralick mean 12 | 0.00502960343399 |
| Daubechies LL | 0.00485792610533 |
| lddp R1 1 R2 2 P8 36 | 0.00470522366795 |
| Gabor scale5 direction 0 | 0.00464088713463 |
| lddp R1 1 R2 2 P8 30 | 0.00457940673565 |
| Gauss scale 2 | 0.00456757325213 |
| Li filter lines scale 4 | 0.00452394338671 |
| median1 | 0.00448419008847 |
| Gauss scale 3 | 0.00428946007681 |
| lbp R1 P8 23 | 0.00422056442136 |
| Li filter dots scale 2 | 0.00419404855368 |
| lbp R1 P8 3 | 0.00417286474165 |
| GLRLM range 4 | 0.00407685537905 |
| Gauss scale 4 | 0.00407667810438 |
| Haralick mean 3 | 0.00404667745008 |
| lbp like variance | 0.00381688149332 |
| Gabor scale7 direction 45 | 0.00380976272062 |
| mean | 0.00370617865063 |
| Kirsch kernel rotation 6 | 0.00359086924995 |
| Kirsch kernel rotation 3 | 0.0035831486833 |
| Gabor scale6 direction 135 | 0.00354934408885 |
| lddp R1 1 R2 2 P8 26 | 0.00354174300477 |
| GLRLM mean 3 | 0.00352957813716 |
| lddp R1 1 R2 2 P8 22 | 0.00352903514392 |
| Kirsch kernel rotation 4 | 0.00351303560906 |
| Gabor scale3 direction 135 | 0.00351064507597 |
| Li filter lines scale 7 | 0.00350753647026 |
| Li filter dots scale 6 | 0.00349944653712 |
| lddp R1 1 R2 2 P8 24 | 0.00349191879531 |
| Li filter dots scale 8 | 0.00348044386091 |
| Haralick mean 5 | 0.0034601514911 |
| Haralick mean 13 | 0.00344603493549 |
| Haralick range 10 | 0.00343370251987 |

| Feature Name | Feature Significance |
|---|---|
| Gabor scale0 direction 45 | 0.00342679095012 |
| lddp R1 1 R2 2 P8 20 | 0.00340951520161 |
| Sobel filter | 0.00338126513956 |
| lddp R1 1 R2 2 P8 28 | 0.00336510020633 |
| Haralick mean 11 | 0.00335545150039 |
| Haralick mean 9 | 0.00335469222705 |
| Li filter dots scale 4 | 0.00335015142284 |
| lbp R1 P8 6 | 0.00333835871345 |
| lbp R1 P8 34 | 0.00333035483815 |
| GLRLM mean 4 | 0.00332240829709 |
| Haralick range 13 | 0.00330576063977 |
| Kirsch kernel rotation 1 | 0.00325367404273 |
| Li filter dots scale 3 | 0.00322725098804 |
| lbp R1 P8 32 | 0.00321563560718 |
| lbp R1 P8 15 | 0.00320942614518 |
| Unoriented gradients axis 2 | 0.00320908597071 |
| lbp R1 P8 22 | 0.0031786741044 |
| lbp R1 P8 4 | 0.00316727342694 |
| GLRLM range 3 | 0.00316716457223 |
| Gabor scale2 direction 0 | 0.00316359058058 |
| GLRLM range 1 | 0.00314148801888 |
| lbp R1 P8 8 | 0.00312613285884 |
| lddp R1 1 R2 2 P8 31 | 0.00309351616083 |
| lbp R1 P8 13 | 0.00304901322091 |
| lbp R1 P8 18 | 0.00299215133888 |
| lbp R1 P8 21 | 0.00292621514968 |
| lbp R1 P8 1 | 0.00285010469346 |
| lbp R1 P8 31 | 0.00284499838049 |
| median2 | 0.00280055590256 |
| histogram FD | 0.0027592163277 |
| DCT | 0.00267273567213 |
| lddp R1 1 R2 2 P8 6 | 0.00266969767907 |
| lbp R1 P8 27 | 0.00265387198784 |
| Gabor scale6 direction 45 | 0.00263196556022 |
| Gabor scale2 direction 45 | 0.00258351483041 |
| lddp R1 1 R2 2 P8 33 | 0.00246078512107 |
| entropy | 0.00242724657238 |
| lddp R1 1 R2 2 P8 14 | 0.00238807977371 |
| Haralick range 4 | 0.00226548830133 |
| Haralick range 3 | 0.00224479933299 |
| lbp R1 P8 11 | 0.00222261581085 |
| Gabor scale0 direction 0 | 0.00219423202378 |
| lbp R1 P8 17 | 0.00215213944642 |
| Gabor scale1 direction 135 | 0.00209305323851 |
| Haar HH | 0.00204822991143 |
| lbp R1 P8 14 | 0.00181210945575 |
| Gabor scale7 direction 0 | 0.00180944032187 |
| Haralick range 9 | 0.00179568772907 |
| Gabor scale1 direction 90 | 0.00174821247035 |
| lddp R1 1 R2 2 P8 4 | 0.00173965526106 |
| lddp R1 1 R2 2 P8 35 | 0.00170983720799 |
| standard deviation | 0.00163956758388 |
| lddp R1 1 R2 2 P8 19 | 0.00162122408481 |
| lddp R1 1 R2 2 P8 2 | 0.00155658309718 |
| Haralick range 6 | 0.00155110083015 |
| Gabor scale3 direction 90 | 0.00143469534661 |
| lbp R1 P8 29 | 0.00120814619045 |
| lddp R1 1 R2 2 P8 12 | 0.00114323375837 |
| lddp R1 1 R2 2 P8 10 | 0.0010061824656 |
| Haar LH | 0.000997831872377 |
| Li filter lines scale 8 | 0.000884510803127 |
| lddp R1 1 R2 2 P8 8 | 0.000797522141631 |
| Li filter lines scale 1 | 0.000711542516032 |
| Gauss scale 7 | 0.000605527853797 |
| Gabor scale2 direction 90 | 0.000517858597515 |
| Gabor scale3 direction 45 | 0.000500100053772 |
| Roberts kernel rotation 2 | 0.000498969587154 |
| Haralick mean 6 | 0.000410848375898 |

Table 7.6: Feature significance-sliding window approach

| Feature Name | Feature Significance |
|---|---|
| Gabor scale6 direction 90 | 0.00842215836757 |
| Haralick range 7 | 0.00841903955573 |
| lddp R1 1 R2 2 P8 14 | 0.00831397565918 |
| lbp R1 P8 1 | 0.00821154377519 |
| Li filter lines scale 4 | 0.00812841241833 |
| Gabor scale6 direction 0 | 0.00812285654039 |
| lddp R1 1 R2 2 P8 28 | 0.00810167537295 |
| Li filter lines scale 5 | 0.00809138952582 |
| lbp R1 P8 6 | 0.00807090095503 |
| lbp R1 P8 17 | 0.00806819869891 |
| lddp R1 1 R2 2 P8 32 | 0.00805723825033 |
| lddp R1 1 R2 2 P8 11 | 0.00805292249742 |
| lbp R1 P8 20 | 0.00801440212432 |
| lddp R1 1 R2 2 P8 34 | 0.00797582671723 |
| Gabor scale0 direction 45 | 0.00796231936367 |
| lddp R1 1 R2 2 P8 19 | 0.00795193990574 |
| Haar HL | 0.00793012822934 |
| Gauss scale 5 | 0.00792612381977 |
| lddp R1 1 R2 2 P8 12 | 0.00784678869401 |
| Daubechies HH | 0.0078390962998 |
| lbp R1 P8 18 | 0.00781319575152 |
| Li filter lines scale 7 | 0.00779725246519 |
| Haralick range 1 | 0.00779460775088 |
| Haralick mean 1 | 0.00779426763859 |
| Haralick mean 8 | 0.00778434954246 |
| Li filter dots scale 7 | 0.00778143968311 |
| Gabor scale2 direction 0 | 0.00777252505935 |
| Haralick range 8 | 0.00774275123882 |
| Gauss scale 2 | 0.00773690162349 |
| Gabor scale4 direction 0 | 0.00771327960749 |
| Gabor scale2 direction 90 | 0.0077093143076 |
| Li filter dots scale 5 | 0.00770581018428 |
| Gabor scale0 direction 90 | 0.00768723915484 |
| lbp R1 P8 27 | 0.00760086144304 |
| Gabor scale4 direction 45 | 0.00756336945732 |
| GLRLM mean 4 | 0.00755240135355 |
| lddp R1 1 R2 2 P8 25 | 0.00754454612367 |
| Gabor scale0 direction 0 | 0.0075209025639 |
| lbp R1 P8 33 | 0.00751839075973 |
| Gabor scale7 direction 0 | 0.00749847558613 |
| lbp R1 P8 22 | 0.00747167297496 |
| standard deviation | 0.0074656373612 |
| Unoriented gradients axis 1 | 0.00742858223526 |
| Li filter lines scale 3 | 0.00741922261199 |
| lddp R1 1 R2 2 P8 20 | 0.0074023460773 |
| lbp like variance | 0.00739326963642 |
| Gabor scale1 direction 0 | 0.00736884074955 |
| Haralick mean 11 | 0.00732651779449 |
| kurtosis | 0.00732164887134 |
| Gabor scale5 direction 0 | 0.0073135520583 |
| lddp R1 1 R2 2 P8 9 | 0.00730348980135 |
| Sobel filter | 0.00729408425265 |
| Haralick mean 5 | 0.00729244004016 |
| lbp R1 P8 14 | 0.00727028035299 |
| lddp R1 1 R2 2 P8 7 | 0.00720628093168 |
| Daubechies LH | 0.00719369000253 |
| Haralick range 12 | 0.00711789358571 |
| GLRLM mean 2 | 0.00708731831836 |
| lbp R1 P8 3 | 0.00707719482148 |
| lbp R1 P8 36 | 0.00699294934395 |
| lbp R1 P8 11 | 0.00693333842131 |
| Daubechies HL | 0.00692646485636 |
| Gabor scale3 direction 45 | 0.00679071859087 |
| lbp R1 P8 8 | 0.00669687616107 |
| lbp R1 P8 29 | 0.00663656963506 |

| Feature Name | Feature Significance |
|---|---|
| U | 0.00654608043497 |
| Haralick mean 12 | 0.00642875808083 |
| lddp R1 1 R2 2 P8 22 | 0.00622614799643 |
| lbp R1 P8 35 | 0.00620391950958 |
| Gauss scale 7 | 0.0061170981542 |
| Kirsch kernel rotation 5 | 0.00607606579697 |
| lddp R1 1 R2 2 P8 36 | 0.00605842053442 |
| lddp R1 1 R2 2 P8 3 | 0.00597966408097 |
| histogram FD | 0.00584727877285 |
| Kirsch kernel rotation 3 | 0.00582829272 |
| lddp R1 1 R2 2 P8 27 | 0.00572647034801 |
| lbp R1 P8 32 | 0.00571879824204 |
| Gabor scale3 direction 0 | 0.00568014615149 |
| Gabor scale1 direction 135 | 0.00553854811701 |
| Kirsch kernel rotation 2 | 0.00549591499379 |
| lbp R1 P8 30 | 0.00538364779137 |
| lddp R1 1 R2 2 P8 30 | 0.00536817131154 |
| lddp R1 1 R2 2 P8 1 | 0.00536549900534 |
| Haralick mean 6 | 0.00532076005194 |
| lddp R1 1 R2 2 P8 5 | 0.00527268263338 |
| Gauss scale 8 | 0.00501645428419 |
| Kirsch kernel rotation 6 | 0.00498024072515 |
| lbp R1 P8 34 | 0.00495469651971 |
| lddp R1 1 R2 2 P8 23 | 0.00495060482847 |
| lbp R1 P8 12 | 0.00452552705167 |
| lbp R1 P8 31 | 0.00416269216955 |
| lbp R1 P8 28 | 0.00409115215269 |
| Haralick mean 10 | 0.00397146907827 |
| Haralick mean 13 | 0.00396172493602 |
| GLRLM mean 1 | 0.00395990883157 |
| Gabor scale3 direction 135 | 0.00395727737489 |
| Gabor scale2 direction 135 | 0.0039350223707 |
| entropy | 0.00389793387773 |
| DCT | 0.00388999959359 |
| Gabor scale1 direction 90 | 0.00383234504108 |
| lbp R1 P8 21 | 0.00381301087041 |
| median3 | 0.00380064028242 |
| skewness | 0.00378855311449 |
| Gabor scale7 direction 135 | 0.00378815869669 |
| Kirsch kernel rotation 8 | 0.0037739558485 |
| Li filter dots scale 6 | 0.00375339392878 |
| lddp R1 1 R2 2 P8 8 | 0.00373784869946 |
| Li filter lines scale 8 | 0.0037215450195 |
| lddp R1 1 R2 2 P8 6 | 0.00370607332483 |
| lddp R1 1 R2 2 P8 26 | 0.0036902580116 |
| lbp R1 P8 9 | 0.00368228082809 |
| Haralick range 2 | 0.0036639751183 |
| Haralick range 13 | 0.00364015104864 |
| lbp R1 P8 24 | 0.00363822495949 |
| median1 | 0.00363629943057 |
| Haralick range 10 | 0.00361545150163 |
| lbp R1 P8 26 | 0.00361212574608 |
| GLRLM mean 5 | 0.00361069449384 |
| Li filter lines scale 2 | 0.00360998654802 |
| lddp R1 1 R2 2 P8 21 | 0.00360089914822 |
| Kirsch kernel rotation 1 | 0.00359511648647 |
| Roberts kernel rotation 1 | 0.0035538802644 |
| GLRLM range 1 | 0.00355301514278 |
| Li filter lines scale 1 | 0.00355273609262 |
| Roberts kernel rotation 2 | 0.00354143055522 |
| Gauss scale 6 | 0.00353180085236 |
| Li filter dots scale 2 | 0.0035275226823 |
| lddp R1 1 R2 2 P8 10 | 0.00352523292055 |
| mean | 0.00350435799256 |
| Gauss scale 1 | 0.00349387105586 |

| Feature Name | Feature Significance |
|---|---|
| Kirsch kernel rotation 4 | 0.00349159870617 |
| Daubechies LL | 0.00348212858751 |
| Gabor scale5 direction 90 | 0.00347752245837 |
| Haar LL | 0.00347687015941 |
| Gabor scale1 direction 45 | 0.00345234715648 |
| Haralick mean 9 | 0.00344737551353 |
| lddp R1 1 R2 2 P8 29 | 0.00344027192357 |
| Haralick mean 3 | 0.00342892187153 |
| lbp R1 P8 2 | 0.0034130053769 |
| Haralick range 11 | 0.00340876217216 |
| Gabor scale4 direction 90 | 0.00340505709255 |
| Li filter dots scale 1 | 0.00338382186827 |
| lbp R1 P8 5 | 0.00337020888223 |
| Haar HH | 0.00336940782033 |
| Haralick mean 4 | 0.00336188277802 |
| Gauss scale 3 | 0.0033543482997 |
| lbp R1 P8 4 | 0.00335074714613 |
| lbp R1 P8 16 | 0.00334099760601 |
| Haar LH | 0.00334029277988 |
| Gabor scale2 direction 45 | 0.00330327609051 |
| GLRLM range 2 | 0.00329954299575 |
| Haralick mean 2 | 0.00328905784387 |
| lddp R1 1 R2 2 P8 17 | 0.00327838962443 |
| lbp R1 P8 15 | 0.00327732513599 |
| lbp R1 P8 7 | 0.00327731153492 |
| Gabor scale4 direction 135 | 0.00325727369699 |
| lbp R1 P8 19 | 0.00325278318507 |
| Haralick range 4 | 0.0032524320311 |
| Gabor scale6 direction 45 | 0.00324584562612 |
| lddp R1 1 R2 2 P8 31 | 0.00324155241718 |
| lbp R1 P8 25 | 0.00323998678174 |
| lbp R1 P8 13 | 0.00323962935852 |
| GLRLM mean 3 | 0.00323646536666 |
| Haralick range 3 | 0.00322907165843 |
| Gabor scale5 direction 45 | 0.00322415292949 |
| lddp R1 1 R2 2 P8 18 | 0.0031862199697 |
| Kirsch kernel rotation 7 | 0.00318596822751 |
| Gabor scale0 direction 135 | 0.00318246725671 |
| lddp R1 1 R2 2 P8 15 | 0.00317866100759 |
| median2 | 0.00315894750617 |
| Gabor scale6 direction 135 | 0.0031527876122 |
| Gabor scale3 direction 90 | 0.00314550538153 |
| GLRLM range 4 | 0.00313964978178 |
| Haralick mean 7 | 0.0031249489669 |
| GLRLM range 3 | 0.00312272946221 |
| lbp R1 P8 23 | 0.00310116071595 |
| Gauss scale 4 | 0.00306740648338 |
| Haralick range 6 | 0.00306729878731 |
| lddp R1 1 R2 2 P8 4 | 0.00306496939691 |
| lddp R1 1 R2 2 P8 13 | 0.00303367794026 |
| Haralick range 9 | 0.00300306073085 |
| Li filter dots scale 3 | 0.00300061032869 |
| Gabor scale7 direction 45 | 0.00299948659471 |
| Li filter dots scale 4 | 0.002994651456 |
| Gabor scale7 direction 90 | 0.00298644329331 |
| lddp R1 1 R2 2 P8 24 | 0.00295922293955 |
| lddp R1 1 R2 2 P8 2 | 0.0029451937237 |
| Gabor scale5 direction 135 | 0.00293498223452 |
| Haralick range 5 | 0.00290928030404 |
| Li filter lines scale 6 | 0.00288839605144 |
| Unoriented gradients axis 2 | 0.00288470826008 |
| GLRLM range 5 | 0.002864648981 |
| Li filter dots scale 8 | 0.00284630081527 |
| image FD | 0.00278815479633 |
| lddp R1 1 R2 2 P8 33 | 0.00278317029101 |
| lbp R1 P8 10 | 0.00266287613629 |
| lddp R1 1 R2 2 P8 16 | 0.00266208064017 |
| lddp R1 1 R2 2 P8 35 | 0.00263305175518 |

# 8

# CONCLUSIONS

## MAIN CONCLUSIONS

The main conclusions can be drawn from tables 7.1, 7.2, 7.3, 7.4. The following become clear:

- In literature, commonly reported accuracy results range between 0.8 and 0.95. These results have been achieved by using multi-parametric MRI, which included T2 and ADC or DWI and in some cases DCE. Only one case where only T2 is used is reported (see table 5.1). This indicates that **T2 sequences alone are inadequate for a radiomics prostate cancer diagnosis.** Although there is information about the presence of cancer or not (as shown in table 7.1, classification results are above random chance), it is not enough to proceed in clinical application. This information is usually seen as dark regions on the T2 images, as shown in fig. 8.1.
- The distinction between peripheral zone versus examination of the entire prostate did not improve the results. Thus, the distinction in peripheral zone in our approach is deemed as redundant when talking solely about T2 images.
- The distinction between Random Forest or the proposed SVM classification scheme did not make a significant difference in classification results.
- Semi-automatic ROI based estimation of C2 and C5 features performs significantly better than the fully automatic sliding window approach. However, this performance is still not good enough.
- Transfering a model trained on a dataset to a completely new dataset was not possible.
- The method proposed for label assignment in Appendix B was not trustworthy.
- The most significant features seem to be Gabor filters, local directional derivative patterns, local binary patterns and wavelet coefficients. Intuitivelly, someone would expect that C2 and C5 features would be the most informative, as they rely merely on texture. Within the same dataset, where the gray values and contrasts are within the same range, this is a reasonable outcome. It is possible that this feature evaluation merely relies on the cross validation experiments, as independent dataset validation experiments are practically a random guess.

Furthermore, the following should be noted:

- The best performance is achieved by ROI based calculations of the C2 and C5 features (see 6.2.2). However, big variances are observed in this case. This reassures our fears that the fact that the ROI size has an impact on the features' values. This might raise questions about the reliabillity of this method. However, it seems to be a promising approach if the issue of shape and area dependence is resolved.
- No differences in performance are observed when using the entire feature set or just the feature set $C2, C5$. In our case, where features are extracted from T2 images, this makes sense as:
    1. Features in both C2 and C5 are designed to capture differences in texture and do not rely in absolute grey values. This could be beneficial in case of T2 imaging, which provides anatomic information and lesions are expected to appear more rough than healthy tissue (see figure 8.1). Moreover, this

(absolute gray value independence) could potentially make applicabillity of a model trained on a dataset and used on a new one possible.

2. Those features comprise 110 out of 198 extracted features, bearing more than half of the available information either way.

3. Note that although working on a space of almost half the dimensionality would be computationally efficient for the classification process, the most computationally expensive procedure in the whole pipeline is by far the extraction of C2 and C5 features with a sliding window. Regarding the "curse of dimensionality", redundant features are disregarded either ways either by PCA or by Random Forest feature selection.

4. Note that this should not apply in case of mpMRI, where, for example, first order statistics that strongly rely on gray values have been found to hold great informational value in case of Diffusion Weighted Images (DWI).

5. While cross validation on the TCIA dataset showed that there is indeed information about the presence of a lesion in the T2 sequences alone, results of the PCMM cross validation experiment were slightly above random chance. The label assignmet process should be considered responsible for this. Moreover, regarding the independent dataset validation experiments, we observe that classification with the SVM classifier on the whole prostate for a ROI based estimation of C2 and C5 features is above random chance. In fact, it is comparable to the best results achieved for cross validation on the PCMM dataset alone. Although accuracies are too low to draw a safe conclusion about applicability on a new dataset, this could imply that this could be a good candidate case that should be further examined for the application of a model on a new dataset.

- If considering a dataset consisting of all pixels for all patients during cross validation, training and testing sets are likely to contain different pixels from images refering to the same patient. In this case, accuracy will be overestimated. Therefore, it is necessary to validate on a dataset consisting of pixels that refer to patients' images that have not been used during training.

- If a single patient is kept out for validation, reported results will probably fluctuate depending on that patient's lesions size and shape. This could be seen as a warning for real life applications, but in our case where more stable results are preferred, more patients should be used to validate

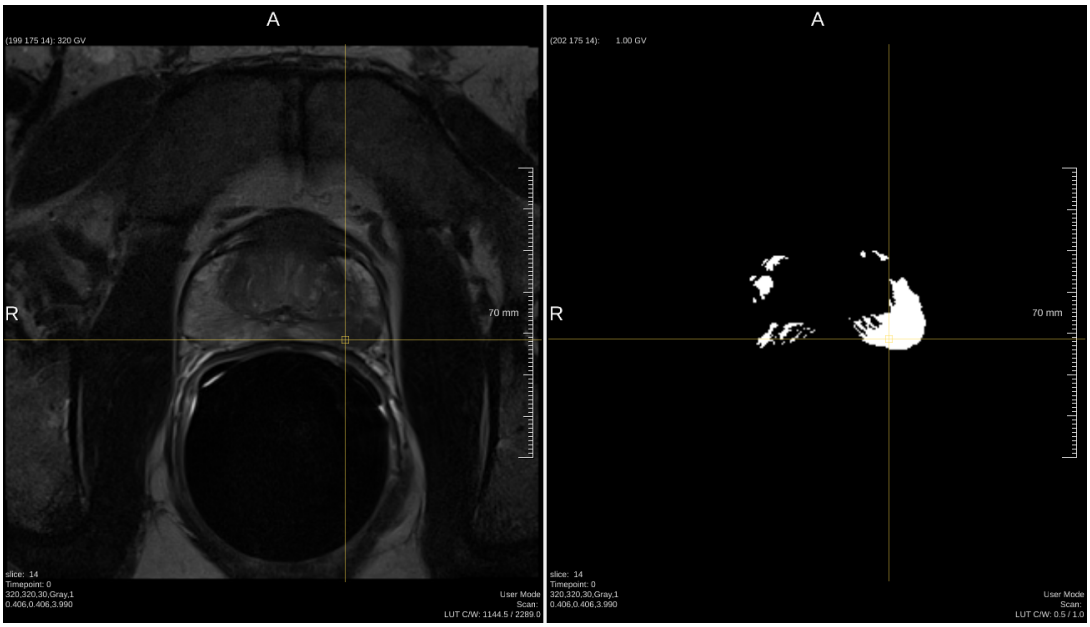



Figure 8.1: A lesion label image (right) and its corresponding point on the axial T2 image (left). Lesions on axial T2 images appear as rough dark regions.

## FUTURE RESEARCH

As future research suggestions, a repeat of the experiment with bigger and better labelled datasets is recommended. This way, safer conclusions could be drawn about the limits of the proposed method. In these experiments, application of multiparametric MRI is also recommended. Dynamic contrast enhanced images that provide information about the vascularization could potentially add great informational value. Diffusion weighted derived images, such as ADC maps are expected to severely improve the results and therefore are highly recommended. This is clear throughout the literature, where great results are reported wherever DWI derived images are used.

Regarding application on T2 images, C2 and C5 ROI based features (see section 6.2.2) might be a great candidate for achieving better overal accuracies in a combined CAD-radiologist detection approach. An implementation of a ROI-size independent and non-compact arbitrary shape support region feature extraction method might yield interesting results.

A fully automated scheme in case of the ROI-based estimation of C2 and C5 features could also be possible. This could be done by an iterrative window size optimization. The process can be as described below:

1. Apply a big window over the entire image slice and find candidate regions.
2. Apply the same procedure with a decreasing window size on each of these regions and further refine search areas.
3. Repeat until a lower limit of window size or until an optimum.

Although the sliding window approach for C2 and C5 features did not seem to bring any good results, thus meaning that it probably is not a good method for feature extraction, this might be an alternative that could work. However, the best case is expected to be a semi-automatic approach. Wherever only T2 sequences are available, this method could provide a good solution. A further advantage of this method, is that feature extraction in this way is not computationally expensive and can be even done on a simple PC.

Perhaps the most promising perspective about fully automated CaP-MRI radiomics overall, seems to be application of deep learning methods. By applying a cleverly designed CNN, manual extraction of features would be redundant and only the optimal features would be extracted and used. No matter how clever manual feature definition might be, mathematically optimal features will be most likely missed. Note that even in this case, exploitation of information provided by DWI would be necessary.

Convolution neural networks have turned out to be the most efficient tool for addressing generic computer vision challenges, such as ImageNet [138]. The problem in case of CaP-MRI readiomics, is the limitted amount of the available data. In all of the reported studies, number of patients does never exceed 300. As a comparison, dataset used for training ImageNet was comprised of 15 million high resolution labelled images for 22000 classes, meaning hardly 650 images for each class. Therefore, an estimation of an adequate dataset size would be in the order of thousands of patients. Such a dataset does not yet exist, however, there are attempts by the community to buid a scalable system that would accommodate a large repository containing images from multiple clinics [139].

In any case, in order for the proposed systems to be clinically applied, they should prove better in some aspect than the current diagnostic techniques. Better might imply higher diagnostic accuracy or a lower degree of invasiveness. This seems as a hard goal for the current state of the art. For exapmple, the first clinical application of radiomics was about breast cancer, in July 2017 [140]. Therefore, in order for clinical application to be possible, great results on a research level is a strong prerequisite. A good candidate for achieving that good results could be deep learning, discussed before. After that, long evaluation studies are also necessary. As a conclusion, in any case, extended clinical application of radiomics is expected to take place in several years.

Finally, as long as efforts are done in the direction of supervised learning, what is of utmost importance is the definition of a robust labeling method. Correct labelling is a tough and essential process. Whereas in table 7.1 classification yields results that imply that there is information in sequences, results of table 7.2 indicate the significance of correct labelling. Therefore, for proceeding in supervised learning approaches, labelling is a crucial step that needs to be adressed. Manual labeling by one or more radiologists is not the best option, as it is subject to human subjection. However, as seen by our results, it has to be the safest option given the current circumstances.

In case where big enough datasets are available, it could be enough to label an entire prostate as the maximum Gleason Score present in a prostate. In that case, our classification space would consist of healthy or not

healthy prostates, making it a lot easier to increase even more the size of the datasets by reducing the required manual labor.

Summing up, the following are recommended, in descending order of significance:

1. The development of a safe and repeatable labelling protocol
2. The use of multiparametric MRI, including T2, DWI and if possible DCE derived images
3. Application of deep learning methods by the time when adequately large datasets become available
4. For a semi-automated approach, redefinition of C2 and C5 features for a non compact arbitrary shape support region

# A

# emPROST 0.1 documentation

## Manifesto

### Purpose of emPROST

*emPROST* software is being developed to extract and finally compare radiomic features from prostate-MRI images. Radiomic features are a popular way to diagnose cancer or even classify tumor aggressivness and several such features have been suggested and applied in several CAD attempts reported in literature.

*emPROST* software caclulates most of these features and saves them to files (.npy files) that represent 4D arrays, where the first 3 dimensions correspond to MRI-image dimensions and the fourth refers to the 198 feature space corresponding to each pixel in the image.

Subsets of these features can subsequently be retrieved and used for processing.

**Warning:** This software has been used and tested only for T2 axial MRI images of the prostate. Moreover, no gold standard for feature values This means that some values are obtained as features, but it is not reliable to estimate the accuracy according to which these values represent the actual feature values. The best way to test would be to get a refference image and compare, but such a testing object is not available.

---

**Note:** Many parameters are hardcoded within the source code and thus failsafe, but if made available to a user interface, controls about these parameters should be implemented (for example parameter *ws* which stands for window size in should be a positive odd integer).

---

### Read before you go-how to use it

*emPROST* implements all necessary preprocessing steps, such as figuring out which images to use, under which directory they are stored, the corresponding mask files etc. However, a sequence of actions is needed:

1. Manual preprocessing for segmentations and ground truth extraction.
2. Feature calculation.
3. Data processing (classification experiments).

#### Preprocessing

Manual preprocessing requires prostate and ROI segmentations to be provided, where, in the mask images, 1 corresponds to prostatic volume or a lesion and 0 to non-prostatic volume or healthy tissue respectively. Optionally a segmentation image of the peripheral zone can be provided, in case that features just for the peripheral zone are required. These files must be saved in a folder that contains the dicom image that they

have been extracted from. This is necessary for making the correct correspondence to the files in the dataset according to the dicom header *patient ID* attribute. The root directory of all these folders will be referred to as the **mask root directory**. Summarizing, mask root directory must contain N subfolders, referring to N patients respectively, each one of which contains:

- A dicom (or *MeVisLab* dicom/tif format) file, to which segmentations correspond.
- *no_previous_mask_init.nii* file that corresponds to the prostate segmenation.
- *pz_mask.nii* (optionally) that corresponds to segmentation of the peripheral zone.
- *rois_mask_.nii* file that corresponds to the annotated lesions.

A reliable and free software to do segmentations is *itk-snap*.

---

**Note:** A *note.txt* file might be also present. In that case, this patient is automatically disregarded. This file should contain notes for the reason that this patient is not taken into account.

---

---

**Note:** All file names (in *italics*) must be precisely as shown in italics.

---

---

**Note:** Any ROIs or ROI parts that fall outside the prostate or peripheral zone segmentation are disregarded.

---

### Calculate features

After the mask root directory is manually created, feature calculation and storage routine can be invoked. This is automatic, given the necessary inputs. Inputs are

-Data root directory, which is the root directory of our dataset. -Mask root directory, described above. -Working directory. This can be empty for the moment, but it will be filled as features are calculated. It will finally contain several text files that contain information about what is stored where and several folders, each one of which will contain the original MRI image, the .npy file containing the features for the same image, text files that contain features' names, image segmentations and ROI masks. Text files containing the directories of files (images and masks corresponding to patients) are first created. Then the folders are created, named after the corresponding patient ID. -Optionally selection of features that are to be estimated, whether to use the entire prostate segmentation or just the peripheral zone and whethern to use sliding window or circumscribed rectangle ROI estimation approach for C2 and C5 feature sets can be additional inputs.

---

**Note:** A status file called *processed_cases.txt* is saved also within the working directory. In that case, patients can be processed in batches, as if the program stops, feature extraction will continue from the patient that was last processed. If this file is editted or deleted, it will start over.

---

### Preprocess and classify

This is also automatic. The user can choose between SVM and Random Forest classifiers. In case of SVM where preprocessing is needed, it is done within the routine. Results (in terms of accuracy of a randomly subsampled subset of the dataset) are saved in text files within the working directory. Results text files' are defined by the user and should reflect the experiment that they refer to. Extracted accuracies refer to pixel classification, thus for example a 60 % accuracy would mean that 60 % of all pixels in all images are classified correctly.

## Dependencies

Python version is 2.7.13. The following list of libraries should be included:

-mahotas

-pydicom

-imblearn

-scipy

-numpy

-skimage

-SimpleITK (preferably including *Simple Elastix*, but not necessary)

-re (usually already present in most *python* installations)

### FUTURE VERSIONS

It is recommended that for future implementations, a custom ROI-size independent arbitray support region C2 feature evaluation routine is developped (in the current version, *mahotas* implementations are deployed, that do not support these attributes).

Moreover, a speedup with multithreading during feature evaluation is a good idea.

Finally, as feature files are basically bitmaps with feature values, this format could be optimized as follows: - Use compression - Save original image, features and segmentations within the same file, so that they don't have to be saved in subdirectories within the working directory. This would also make visualization of features easier. A potential way to achieve this is itk vector image, which is unfortunatelly not supported in *SimpleITK*.

## NAVIGATING THE DATASET

### MAKING THE DATASET MAP

Module *separate_modalities* hosts all functions that generate the text files stored within the working directory and containing the directories of all interesting files (masks and images) for each patient. Finally, only patients about who all information is available are further processed. This selection is done internally, whereas text files contain directories for all available patients.

`separate_modalities.check_mask_integrity(`*mask_dir*, *working_directory*, *patient_ID*, *mask_format_name*, *fix_masks=False*`)`

"Masks are extracted with a in-house developped tool (SegmentationTool3) on *MeVisLab* some masks are hollow on some planes (only the contours are saved). Check for a mask if it is hollow or not in each plane. If it is so, fill holes with ones and save a corrected copy. It performs operations on a single mask file each time. This function is by default deactivated, as most masks are fixed manually. However, it is still possible to deploy.

**Parameters**

- **mask_dir** (*string*) – mask directory
- **working_directory** (*string*) –
- **patient_ID** (*string*) –
- **mask_format_name** (*string*) – which kind of segmentation files to fix (peripheral zone or entire prostate). It has not been tested on ROI segmentations. Either way, it is not necessary for ROI segmentations. Thus, it can be *no_previous_mask_init.nii* for the entire prostate or *pz_mask.nii* for the peripheral zone.
- **fix_masks** (*bool*) – If set to True, perform a check, correct problematic masks and save corrected files. Default is False.

`separate_modalities.correct_masks(`*working_directory*, *mask_root_directory*, *mask_format_name*`)`

"Read directories file from *uncorrected_mask_paths.txt*, serial feed to check_mask_integrity and update file entry for mask. *corrected_masks.txt* map is thereafter created, which contains the directories of the masks actually used during feature extraction.

**Parameters**

- **mask_dir** (*string*) – mask directory
- **working_directory** (*string*) –
- **mask_format_name** (*string*) – which kind of segmentation files to fix (peripheral zone or entire prostate). It has not been tested on ROI segmentations. Either way, it is not necessary for ROI segmentations. Thus, it can be *no_previous_mask_init.nii* for the entire prostate or *pz_mask.nii* for the peripheral zone.

separate_modalities.**file_list2**(*working_directory*, *data_directory*, *reset=False*)

Create and save file *all_modalites.txt* in the working directory, which is the major map to a raw dataset. This file contains in lines the patient ID followed by lines that refer to the sequence description, pixel spacing and file full path (dicom directory). It is readable by humans. Then the next patient ID follows and so on.

**Parameters**

- **working_directory** (*string*) –
- **data_directory** (*string*) – raw dataset root directory
- **reset** (*bool*) – If an *all_modalities.txt* file already exists, delete it and make a new map. Default is False.

separate_modalities.**fix_dictionaries**(*reference*, *dic2*)

"check if dicitonaries have the same keys. If not, delete entries to make them.

**Parameters**

- **reference** (*dictionary*) – true dictionary
- **dic2** (*dictionary*) – dictionary to edit

separate_modalities.**get_dictionary**(*modality_directories*, *reset=False*)

Return a dictionary with patient IDs as keys and modality directories as entries corresponding to them. For parallel dictionairies creation. Input is T2_directories.txt or DWI_direc, or masks_corrected

separate_modalities.**mask_list**(*working_directory*, *mask_root_directory*, *status_file_name*, *file_name_ending*)

"Create text files with patient IDs and the corresponding directories of masks or ROIs. It can generally be used to map any file referring to this patient and is stored withn the patient subdirectory under the mask root directory.

**Parameters**

- **working_directory** (*string*) –
- **mask_root_directory** (*string*) –
- **status_file_name** (*string*) – name of the minor map text file

separate_modalities.**process_files**(*working_directory*, *data_root_directory*, *mask_root_directory*, *mask_format_name*)

Entire dataset mapping and preprocessing, which includes mapping the dataset, retrieving modality and mask directories. Then check for which of all candidate patient IDs all necessary data is present. Necessary data includes in current version: * axial T2 image * prostate or peripheral zone segmentation image * Regions of interest image Finally only keep patients for which all this information is present. For these patients, create within the working directory a subdirectory named after the patient ID in which all these information are copied and feature files are going to be stored.

**Parameters**

- **working_directory** (*string*) – At this point working directory can be completely empty
- **data_root_directory** (*string*) –
- **mask_root_directory** (*string*) – It is the directory under which all patient ID subderectories are present. Those must contain an itk image referring to the prostate

or peripheral zone segmentation, an itk image referring to ROIs and a *MeVisLab* dicom/tif image corresponding to the axial T2 image of the patient, after which masks are extracted. This is necessary to retrieve patient ID and spacing information.

- **mask_format_name** (*string*) – indicates if whole prostate segmentations or peripheral zone segmentations shall be used. Can be *no_previous_mask_init.nii* for the entire prostate or *pz_mask.nii* for the peripheral zone.

separate_modalities.**retrieve_modalities**(*modality_name*, *working_directory*, *reset=False*)

Create and save files in the working directory, that are minor maps to a raw dataset. Theses files are generated by map *all_modalities.txt* and like the format of *all_modalities.txt*, contain in lines the patient ID followed by lines that refer to the sequence description, pixel spacing and file full path (dicom directory).They are readable by humans. Then the next patient ID follows and so on. The difference is that they only refer to files that correspond to a specific sequence description. This description is retrieved by *all_modalities.txt* by a regular expression matching, so that they match the pattern of a specific sequence (e.g. T2 or DWI). In case a new dataset is introduced, where sequence descriptions differ, regex matches can be editted in a single point in the code, which is in global_functions.regex_match

**Parameters**

- **modality_name** (*string*) – can be only T2 or DWI.
- **working_directory** (*string*) –
- **reset** (*bool*) – If the file already exists, delete it and make a new map. Default is False.

## ESTIMATING FEATURES

The following modules are used to extract prostate radiomic features mentioned in literature. Whenever a method described in a paper is used, this is ceited either in the docstring of the function, or in inline comments in the source code. Five major modules are meant to caclulate the features. Each one of them has a structure, in which, a major function named as the module itself (e.g. *C1.C1*, *C2.C2* etc) is called. This function then calls other functions implemented in each module and returns a dictionary with all the features that it has calculated. All these modules are contained within the directory *src/features*, a subdirectory of the main source code directory.

The following holds for these modules:

- C1 calculates first order statistics.
- C2 calculates second order statistics.
- C3 is for gradient features.
- C4 is for wavelet features. Note that extraction of features for multiple scales after laplacian pyramidal decomposition is implemented in *main1*.
- C5 is to calculate fractal features.

C2 and C5 features can be calculate either by the sliding window or by the ROI based approach.

### ROI BASED APPROACH

ROIs over which to calculate the features were defined by the ground truth. As *mahotas* library (or any library available) does not cover the case of support regions of arbitrary shape, the method to apply this was to consider a circumscribed rectangle around each ROI marked as positive. As a slice might contain multiple ROIs, the remaining healthy space was then partitioned in a set of rectangles. The boundaries of those were defined by the boundaries of the rectangles corresponding to positive ROIs extended until the end of the image and the boundaries of the image itself. This can be formulated as follows:

If N contours described by a parameter $t_i(x_i, y_i), i \in [1, N]$ respectively lie within a rectangle bounded by $x \in [x^0, x^1]$ and $y \in [y^0, y^1]$, then for each $x_i, y_i$ holds $x_i \in [x_i^0, x_i^1]$ and $y_i \in [y_i^0, y_i^1]$, where $x_i^0, x_i^1 \in [x^0, x^1]$ and

$y_i^0, y_i^1 \in [y^0, y^1]$. Then the ROIs over which the features are calculated are defined by the intersections of the lines defined by

$$x = x^0, x_1^0, \ldots, x^1$$
$$y = y^0, y_1^0, \ldots, y^1$$

**Warning:** For C2, implemented mostly by *mahotas* library, feature values were found to have a dependence on the region size (the smaller the region the smaller the value).

### SLIDING WINDOW

9x9 pixels sliding window is a more secure alternative.

### C1 FEATURES

`C1.C1(`*img*`)`

Returns first order features. See C1.window_statistics for more

*ws* is an important parameter (window size) and it is defined within the function source code. It refers to a window which is *wsxws* pixels. Preset value is 9x9.

**See also:**

documentation of *C1.window_statistics*

**Parameters**

- **`img`** (*`np.array`*) – the image about which features are calculated
- **`intensity_levels`** (*`int`*) – represents the dataset global maximum of intensity levels

**Returns** dictionary with all features

`C1.window_statistics(`*window, intensity_levels*`)`

Calculates first order statistical radiomic features.

Features are described in detail in literature review. These include first four order moments, and some special features, the reference of which can be find in the block comments in the code. All features are calclulated for a neighborhood of size *ws x ws* around each pixel. Parameter ws is defined in function *C1.C1* source code which in turn calls C1.window_statistics via *global_functions.pixelwise_features* giving as input the window, rather than the entire image. Then this function is called for a window of size ws x ws for each pixel in the image and for this window (referring to a single pixel) all features are returned in form of a dictionary.

**Parameters**

- **`window`** (*`np.array`*) – window on which features are calculated
- **`intensity_levels`** (*`int`*) – represents the dataset global maximum of intensity levels

**Returns** dictionary with first order statistics, shape of the image containing 1x4 lists

### C2 FEATURES

`C2.C2(`*img, return_array=False*`)`

returns 2nd order statistics as described in literature review. GLRLM and GLCM are calculated over a 5 grey level image to avoid sparse matrices. LBP and LDDP are calculated over a 256 grey level image. 14th Haralick feature is not calculated as mahotas implementation used here is buggy and for a sliding

window excecution hangs at arbitrary points. However, for ROI based aproach, it seems to be working. Still, it is not considered safe to use it.

For references see:

M. M. Galloway, “Texture analysis using gray level run lengths,” Com- puter graphics and image processing, vol. 4, no. 2, pp. 172–179, 1975.

R. M. Haralick, K. Shanmugam, et al., “Textural features for image clas- sification,” IEEE Transactions on systems, man, and cybernetics, no. 6, pp. 610–621, 1973.

Note that for the last reference, all 28 features are calculated.

**Parameters**

- `img` (*np.array*) – image for which the second order statistical features presented in the above papers are extracted
- `return_array` (*bool*) – if True, an array with all features for the specified pixel is returned. This is used to describe the feature vector referring to the specific pixel in a sliding window aproach. Therefore, it should be only set in sliding window aproach, otherwise an exception should be expected. Default is False (in accordance with default values of the main function).

**Returns** dictionary or numpy array with features

`C2.GLRLM`(*img, intensity_levles*)
calls *chainGLRLM* for rotated versions of input image and returns dictionary with GLRLM features

`C2.GLRLM_features`(*p*)
Returns 5 Grey Level Run Length Features as described in M. Galloway paper

M. M. Galloway, “Texture analysis using gray level run lengths,” Com- puter graphics and image processing, vol. 4, no. 2, pp. 172–179, 1975.

**Parameters** `p` (*list*) – p is the matrix containing gray level run lengths (2D)

**Returns** a list containing all 5 features

`C2.chainGLRLM`(*img, intensity_levels, width*)
returns normalized grey level run length matrix for already rotated image.

An image (img input parameter) is given as input. The image has to already be rotated. A rotated image looks like this: if the original image is an np.array with shape (512, 512), a rotated image at 45 degrees will have 1,2,..., 1024, 1023,...,2,1 row elements. This means that the pixels are given as row elements for each diagonal line. Then all subsequent occurences of all gray level are counted for each row and returned as the GLRLMat. For more information see:

M. M. Galloway, “Texture analysis using gray level run lengths,” Com- puter graphics and image processing, vol. 4, no. 2, pp. 172–179, 1975.

**Parameters**

- `img` (*np.array*) – Rotated image for which the consecutive occurences of all gray levels are caluclated
- `intensity_levels` (*int*) – the number of intensity levels present in an image
- `width` (*int*) – the width of the image in pixels

**Returns list** 2 x 2 list containing countings of consecutive occurences of all gray levels

`C2.lddp`(*image, radius1, radius2, points, ignore_zeros=False, preserve_shape=True*)
Custom implementation of 2nd order local directional derivative pattern Originally obtained from mahotas.features.lbp_transform function.

An inner and an outer radius with respect to a point, which is each image pixel are selected. Then, a set of points are obtained by interpolation on these radii, according to the number defined by *points* argument. Note that if, for example, 8 points are given, there are 8 points that are considered on the inner radius defined by equal angles starting from the central point and each one of them. If these two

points (the origin, or the centre) and each point define a straight line, also 8 points on the same lines are considered for the outer radius.

For reference see :

Guo, Zhenhua, et al. "Local directional derivative pattern for rotation invariant texture classification." Neural Computing and Applications 21.8 (2012): 1893-1904.

**Parameters**

- **image** (*np.array*) – numpy array input image
- **radius1** (*int*) – inner radius (in pixels)
- **radius2** (*int*) – outer radius (in pixels)
- **points** (*int*) – number of points to consider. It should be given regarding the inner radius, as the second set of points will be aligned to the ones lying in the inner circle.

**Returns** lddp histogram

### C3 FEATURES

C3.C3(*img, resolution*)

Returns all gradient features

All the gradient features include the following:

- Kirsch gradients
- Roberts gradients
- Sobel filter
- Unoriented gradients
- local binary patterns
- local directional derivative patterns
- Gabor filter coefficients
- Gaussian blurred images

Note that for the Gabor and Gaussian filters, scales range between 2 and 8 mm as to capture scales within which there could be a lesion. This method is mentioned in:

G. Litjens, O. Debats, J. Barentsz, N. Karssemeijer, and H. Huisman, "Computer-aided detection of prostate cancer in mri," IEEE transactions on medical imaging, vol. 33, no. 5, pp. 1083–1092, 2014.

**Parameters**

- **img** (*np.array*) – the image about which features are calculated
- **resolution** (*float*) – represents the physical dimension to which a pixel corresponds

**Returns** dictionary with all gradient features.

C3.li(*img*)

Implementation of the Li filter

Filter is described in detail in:

Q. Li, S. Sone, and K. Doi, "Selective enhancement filters for nodules, vessels, and airway walls in two- and three-dimensional ct scans," Medical physics, vol. 30, no. 8, pp. 2040–2051, 2003.

Roughly, it is a dot and line sensitive filter.

**Parameters** **img** (*np.array*) – image on which the filter is applied

**Returns** a list containing two numpy arrays. The first is the filter that refers to dots, while the second refers to lines. Each of these has the shape of the image (a value for each pixel)

## C4 FEATURES

`C4.C4(`*img*`)`
Returns Daubechies and Haar wavelet coefficients

Daubeschies wavelet coefficients are averaged on a 7 x 7 window as mentioned in:

R. Lopes, A. Ayache, N. Makni, P. Puech, A. Villers, S. Mordon, and N. Betrouni, "Prostate cancer characterization on mr images using fractal features," Medical physics, vol. 38, no. 1, pp. 83–95, 2011.

Depending on the input image, the maximum level of scale decomposition might vary. This depends on pywavelet library that is used to extract the coefficients.

`C4.pad_to_shape(`*coeffs, original_shape*`)`
"pad with zeors to get the original slice shape for each scale. return_arry[0,0] and [0,1] are the original dims

## C5 FEATURES

`C5.C5(`*img, return_array=False*`)`
Returns fractal dimension features.

The algorithms that have been deployed to extract fractal features is the popular 3D box counting. In the following reference:

R. Lopes, A. Ayache, N. Makni, P. Puech, A. Villers, S. Mordon, and N. Betrouni, "Prostate cancer characterization on mr images using fractal features," Medical physics, vol. 38, no. 1, pp. 83–95, 2011.

It is recommended that 8-tap Daubechie wavelet coefficients could be used to estimate the fractal dimension (FD) but this method has not been implemented. 3D box counting is presented in:

B. Stark, M. Adams, D. Hathaway, and M. Hagyard, "Evaluation of two fractal methods for magnetogram image analysis," Solar Physics, vol. 174, no. 1-2, pp. 297–309, 1997.

Fractal dimmensions are estimated for the image itself given as input *img* and the histogram of the same image, a method recommended in

D. Lv, X. Guo, X. Wang, J. Zhang, and J. Fang, "Computerized charac- terization of prostate cancer by fractal analysis in mr images," Journal of magnetic resonance imaging, vol. 30, no. 1, pp. 161–168, 2009.

Implementation of the algorithm has been properly adapted from an implementation by Francesco Turci, retrieved online from https://francescoturci.wordpress.com/2016/03/31/box-counting-in-numpy/ on 10 Feb 2017.

**Parameters**

- **`img`** (*`np.array`*) – Image the fractal dimension of which is estimated. Also, image histogramm's FD is estimated.
- **`intensity_levels`** (*`int`*) – represents the dataset global maximum of intensity levels
- **`return_array`** (*`bool`*) – if True, an array with all features for the specified pixel is returned. This is used to describe the feature vector referring to the specific pixel in a sliding window aproach. Therefore, it should be only set in sliding window aproach, otherwise an exception should be expected. Default is False (in accordance with default values of the main function).

**Returns** dictionary with image and histogram FDs or numpy array in sliding window mode

## CLASSIFICATION

In the current version, an automated *main* function is implemented for the module *classification*. This calls recursively excecution instructions for all possible scenaria for the experiments that were run, having hardcoded the corresponding directories. This should be editted during any external application.

`load_npy.get_feature_names(`*patient_directory*`)`
Get all features names in a list with the corresponding order that they are saved in the vectors of the .npy feature file.

**Parameters** `patient_directory` (*string*) – Where feature name files are saved.

**Returns** list: all feature names in the correct order.

`load_npy.get_features_subset(`*patient_directory, feature_file, feature_names*`)`
Get a subset of all features for an image as a 4d numpy array.

**Parameters**

- `patient_directory` (*string*) –
- `feature_file` (*string*) –
- `feature_names` (*list*) – Features that should be retrieved

**Returns** 4d numpy array: Requested features of the specific image for the volume of interest.

`load_npy.load_file(`*patient_directory, feature_file, feature_name, return_array=False*`)`
Retrieve a single feature from a feature file and either return it for further processing or save it as itk image for visualization. Visualization shows the true feature values for the volume of interest over which the feature has been calculated and sets the rest of the image to zero, so there is a common world matrix with T2 image and masks and comparison is made possible. However, those zero values do not correspond to actual feature values. Moreover, during classification or processing, no feature values outside the mask are taken into account.

**Parameters**

- `patient_directory` –
- `feature_file` –
- `feature_name` –
- `return_array` – If set to True, array is returned for further processing, otherwise an itk image file is saved for visualizing the specific feature. Default value is False.

**Returns 3d numpy array** a feature value (only if return_array is set to True)

`classification.RF_trees_number_optimization(`*working_directory, keep_features, no_xval=5, case_name='undefined_', multi_dir=False*`)`
call classify RF recursively for several numbers of trees of the random forest

`classification.case_split(`*input_string*`)`
used to be main, but with change of cluster can be submitted as a single job

`classification.classify_RF(`*working_directory, keep_features, no_xval=5, no_trees=60, case_name='undefined_', multi_dir=False*`)`
Train and evaluate a random forest model with cross validation. Save the results of each cross validation fold in a text file within the working directory.

**Parameters**

- `working_directory` (*string*) –
- `keep_features` (*string*) – full path to a text file containing all features (in lines) that should be loaded
- `file_name` (*string*) – name of the file where the results are saved
- `no_xval` (*int*) – number of cross validation folds. Default is 5.

- **no_trees** (*int*) – number of trees of the random forest. Default is 60.

**Returns** list of cross validation folds accuracies

`classification.`**classify_RF_independent**(*training_directory, test_directory, keep_features, no_trees=70, case_name='undefined_', multi_dir=False*)

Train a Random Forest model and evaluate on an independent dataset. Save the resulting accuracy in a text file within the working directory.

**Parameters**

- **training_directory** (*string/list*) – root directory or directories of the training set
- **test_directory** (*string*) – root directory of the test set
- **keep_features** (*string*) – full path to a text file containing all features (in lines) that should be loaded
- **case_name** (*string*) – name of the file where the results are saved. It is not the full name, as the file will always end with _independent_test_RF.txt. This is the initial part of the file name, that should be indicative of the case (e.g. **peripheral_zone_all_features_**).
- **no_trees** (*int*) – Number of trees used by the random forest. Default is 70.
- **multi_dir** (*bool*) – Must be true if training_directory is a list of directories. This means that data under multiple roots can be used as a training set and a model on multi-clinic data can be trained. Default is False.

**Returns** classification accuracy

`classification.`**classify_svm**(*working_directory, keep_features, no_xval=5, no_features=15, case_name='undefined', multi_dir=False*)

Scale the data with RobustScaler (see documentation of sklearn.preprocessing.RobustScaler) as SVM is sensitive to the data scales. Then apply PCA on the new dataset. Train and evaluate a svm model with cross validation. Save the results of each cross validation fold in a text file within the working directory.

**Parameters**

- **working_directory** (*string*) –
- **keep_features** (*string*) – full path to a text file containing all features (in lines) that should be loaded
- **file_name** (*string*) – name of the file where the results are saved
- **no_xval** (*int*) – number of cross validation folds. Default is 5.
- **no_features** (*int*) – number of features that are kept during PCA. Default is 15. (Usually many more features are present).

**Returns** list of cross validation folds accuracies

`classification.`**classify_svm_independent**(*training_directory, test_directory, keep_features, no_features=15, case_name='undefined', multi_dir=False*)

Train a SVM model and evaluate on an independent dataset.

Scale the data with RobustScaler (see documentation of sklearn.preprocessing.RobustScaler) as SVM is sensitive to the data scales. Then apply PCA on the new dataset. Train and evaluate a svm model. Save the resulting accuracy in a text file within the working directory.

**Parameters**

- **training_directory** (*string/list*) – root directory or directories of the training set
- **test_directory** (*string*) – root directory of the test set

- **keep_features** (*string*) – full path to a text file containing all features (in lines) that should be loaded
- **case_name** (*string*) – name of the file where the results are saved. It is not the full name, as the file will

always end with _PCA_independent_svm.txt. This is the initial part of the file name, that should be indicative of the case (e.g. **peripheral_zone_all_features_**). :param bool multi_dir: Must be true if training_directory is a list of directories. This means that data under multiple roots can be used as a training set and a model on multi-clinic data can be trained. Default is False. :param int no_features: number of features that are kept during PCA. Default is 15. (Usually many more features are present).

**Returns** classification accuracy

classification.**discriminate_dataset**(*working_directory*)

Returns a list with all full paths for all the files that shall be processed within a working directory.

classification.**get_all_features**(*working_directory*, *features_switch*)

Save a keep features text file inside the working directory. This contains in each line a feature name that should be loaded. This file is based on only one patient directory within the working directory. Therefore, it is assumed that the same features have been calculated for all patients, otherwise an exception will be raised when trying to load features that have not been calculated.

**Parameters**

- **working_directory** (*string*) –
- **features_switch** (*list*) – Which of the features to retrieve (from families C1, C2, C3, C4, C5).It is important to feed the features switch in order (eg never give f3, f5,f1 but f1, f3, f5)

**Returns** string which is full path to keep features text file

classification.**independent_patient_xval**(*working_directory*, *keep_features*, *case_name*)

Leave one out cross validation. Training set is the total of pixels of all -1 patients, test set is set of pixels of 1 patient.

classification.**load_data**(*working_directory*, *feature_file*, *keep_features*, *exclude_circumscribed=True*, *class_balance=True*)

Load and return as numpy array the selected calculated features from the specified feature file. If no positive labels are found in an image this is possibly due to registration errors and nothing is returned.

**Parameters**

- **working_directory** (*string*) –
- **feature_file** (*string*) – full path to feature file
- **keep_features** (*string*) – full path to a text file containing all features (in lines) that should be loaded
- **exclude_circumscribed** (*bool*) – If set to True, pixels where circumscribed ROIs do not correspond to actual label values are disregarded. This only has an effect to ROI based calculation of C2 and C5 features. Default value is True.
- **class_balance** – If set to True, majority class random undersampling is performed and a balanced dataset is returned. Default value is True.

**Returns** numpy array with feature vectors for all the pixels

**Returns** numpy array with the labels corresponding to the vectors in the first returned array

classification.**load_dataset**(*working_directory*, *keep_features*, *multi_case=False*)

**Returns dataset with all selected features for all feature files within a working directory. (The entire dataset,** where data points refer to vectors that correspond to pixels for evey image.)

**Parameters**

- **working_directory** (*string*) –
- **keep_features** (*string*) – full path to a text file containing all features (in lines) that should be loaded

**Returns**

classification.**multi_directory_data**(*dir_list, keep_features*)

Load and return data as an array of vectors and the corresponding labels from multiple directory roots. These arrays and labels can be pipelined to any sklearn model.

**Parameters**

- **dir_list** (*list*) – list of full root directories of datasets (the corresponding working directories, where all feature files are saved. Not the actual raw dataset directories.)
- **keep_features** (*string*) – full path to a text file containing all features (in lines) that should be loaded

**Returns** numpy array with feature vectors for all the pixels of all feature files under dir_list entries

**Returns** numpy array with the labels corresponding to the vectors in the above array

classification.**reduction_analysis**(*working_directory, case_name='undefined', features_switch=['f1', 'f2', 'f3', 'f4', 'f5'], multi_dir=False*)

Perform PCA analysis and save the eigenvalues in descending order in text files. This analysis is done both on the original data and on data that has been scaled with an outlier robust scaler, which selects values within the 25 and 75th percentile of all values. (see documentation of sklearn.preprocessing.RobustScaler)

**Parameters**

- **working_directory** (*string*) –
- **file_name** (*string*) – name of the output text file
- **features_switch** (*list*) – Which of the features to retrieve (from families C1, C2, C3, C4, C5).It is important to feed the features switch in order (eg never give f3, f5,f1 but f1, f3, f5). Default value is ['f1', 'f2', 'f3', 'f4', 'f5'].

## FURTHER FUNCTIONALITIES

Some further functionalities have been developped. *global_functions* module is essential for all other modules, carrying out some vital tasks needed globally. Other modules ( *change_to_nifti* and *registration* ) just automate some preprocessing steps but their presence is optional during automated routines. However, *registration.dicom_series_to_nii* is still essential and should allways be present.

global_functions.**ROI_based_calclulations**(*img, ROIs, the_function*)

For ROI based calculation of C2 and C5 features. Apply the function defined as input to regions defined by the circumscribed rectangles around regions of interest defined at ROIs. ROIs is a numpy array representing the mask where 0 refers to healthy tissue and 1 to lesions. There can multiple lesions of arbitrary shape. Also a 4-connected 1-pixel binary errosion and dilation is applied on ROIs to compensate for registration errors of the ROIs masks, that usually are the outputs of a registation and salt and pepper noise is frequently observed. Lesions with a total area less than 10 pixels are disregarded as they are probably a result of registration errors that are not compensated during erosion-dilation.

**Parameters**

- **numpy array img** (*2d*) –
- **numpy array ROIs** (*2d*) –
- **object the_function** (*function*) –

**Returns** features dictionary, where keys are features' names and entries are 2d numpy arrays with the feature values over each pixel.

global_functions.**custom_pixelwise_features**(*img*, *ws*, *function*, **extra_args*)

Returns features for the windows within the image calculated by function.

It is a common issue that many features have to be calculated for each pixel within the image, with respect to a neighborhood of pixels, referred to as window. Window has a size *ws x ws*. *pixelwise_features* calls iteratively a function that calculates all these features. Function *function* can have in this version arbitrarilly many arguments given as the input specified by *extra_args*. *pixelwise_features* calls then this function for every window within the image. This means, that , for example, two directly subsequent windows only differ in one value.

The important thing about *pixelwise_features* is that it takes in account boundary conditions. For a pixel that lies close to the boundary and a window of size *ws x ws* would yield an index error, is truncated to a window just fitting in the image. For example, the window of a pixel at *img[1,0]* for *ws* = *9*, would be *img[0:5, 0:4]*.

It is used for sliding window estimation of C2 and C5 features. Otherwise, *pixelwise_features* which calls an implementation from *ndimage* library, which is way more efficient is used.

**Parameters**

- **img** (*np.array*) – image for which the features are calculated
- **ws** (*int*) – window size
- **function** (*function.object*) – function that calculates the features. It only takes one input, namely window (e.g. img[0:5, 0:4]
- ***extra_args** (*any*) – Any further set of parameters that function *function* could take as input

**Returns** dictionary where keys are features names and entries are 2d arrays with feature values over the image.

global_functions.**healthy_rois**(*img*, *roi_inds*, *ROIs*, *a_function*)

For ROI based calculation of C2 and C5 features. Apply the function defined as input to regions defined by the dual of circumscribed rectangles around regions of interest defined at ROIs. Healthy ROIs are defined by the intersections of lines defining the boundaries of circumscribed rectangles around regions defined by ROIs, thus they are compact and recangularly shaped.

**Parameters**

- **numpy array img** (*2d*) –
- **roi_inds** (*list*) – indices of lines defining the boundaries of circumscribed rectangles

around regions defined by ROIs :param 2d numpy array ROIs: :param function object a_function: function to apply on healthy regions

**Returns** features dictionary, where keys are features' names and entries are 2d numpy arrays with the feature values over each pixel.

**Returns** A list with coordinates of the healthy ROI circumscribed rectangle is also returned.

global_functions.**pixelwise_features**(*img*, *ws*, *function*, **extra_args*)

Sliding window feature estimation by using *ndimage.filters.generic_filter*, which is much faster than the custom implementation.

**Parameters**

- **img** (*np.array*) – image for which the features are calculated
- **ws** (*int*) – window size
- **function** (*function.object*) – function that calculates the features. It only takes one input, namely window (e.g. img[0:5, 0:4]
- ***extra_args** (*any*) – Any further set of parameters that function *function* could take as input

**Returns** 2d numpy array with feature values over the image.

`global_functions.regex_match(necessity, string_to_match)`

"Gathering all regex matches used throughout the code to make it applicable to other datasets by modifying. Therefore, if we have to train on a new dataset where sequence description is in a new format, only this function has to be editted and all regex matches used to make the dataset mapping will be up to date. Generally regex are extensivelly used to navigate through map files and identify if something is a directory, etc. In short, it is used to find what is what.

**..note:: Downside is that this script will have to be imported to other scripts under module directory.** This makes it impossible to import any script from modules directory to *global_functions* script, otherwise mutual import will happen and an exception will be raised.

**Parameters** `necessity` (*string*) – can be *patient image, patient image, axial T2, axial DWI, windows or unix directory*

It indicates what we are looking for, which means that we are questioning if a string is a directory or an image etc. :param string_to_match:

**Returns** Bool, True if the string is what we are looking for, false otherwise.

`global_functions.set_in_range(img, indices, values)`

Crop an array to the boundaries specified by indices and fill it with specified values. New image shape and values must have the same shape.

**Parameters**

- `numpy array img` (*2d*) –
- `array indices` (*numpy*) –
- `numpy array values` (*2d*) –

**Returns** 2d numpy array with new shape and values

`global_functions.single_values_to_img(dictionary, img_shape)`

"For ROI based features, replicate img.shape times the single value over the entire ROI

`global_functions.trim_array_to_ROI(img, return_support_region=False)`

Crop an image to the boundaries specified by the circumscribed rectangle of the region defined by a mask.

**Parameters**

- `numpy masked array img` (*2d*) – mask is the region of interest.
- `return_support_region` (*bool*) – If set to true, also the boudaries of the circumscribed rectangle are returned.

Default is False.

**Returns** 2d numpy array, cropped image

**Returns** If *return_support_region* is set, also a list with coordinates of the ROI circumscribed rectangle is returned.

**Warning:** *global_functions* module should **never** import any other module from a relative path.

`change_to_nifti.change_an_image(img_file)`

Change an image from mhd or mha itk format to nii itk format. This is mostly for legacy, as in previous efforts mhds were used.

**Parameters** `img_file` (*string*) – full path to the file

`change_to_nifti.find_mhds(root_directory, change_to_nifti=True)`

"change all .mhd or .mha images under a root directory to .nii

change_to_nifti.**resize_nifti**(*img_file*)

"It is a common case that image files when transferred from DICOM format use a 64 bit float format in itk images. This function sets unsigned 8 bit integer format for mask or roi files and int 32 for other images, resulting in an important file size reduction.

**Parameters** **img_file** (*string*) – full path to the file

change_to_nifti.**set_masks_world**(*root_directory*)

"Each patient folder has 3 nii files, one for masks, one for ROIs and the axial T2. In the working directory, each of these is set to the same values according to the mask, which is set according to the T2 DICOM.

**Parameters** **root_directory** (*string*) –

registration.**dcm_series_to_nii**(*dump_directory*, *working_directory*, *overwrite=False*)

**Reads dicom files from dump directory and writes single nii file** in working directory for feeding in elastix. Also creates the patient directory within the working directory.

..note:: if a file is already there it is overwritten

**Parameters**

- **dump_directory** (*string*) – directory of the dicom files
- **working_directory** (*string*) –
- **overwrite** (*bool*) – If set to True, patient folder is removed and a new one is created. Default is False.

registration.**patient_identifier**(*mask_root_directory*, *all_modalities_file*, *patient_no*)

Obsolete, shall not be used

registration.**recursive_patient_registration**(*msk_root*)

Obsolete, shall not be used

registration.**roi_registration**(*directory*)

Fine tuned call of *Simple Elastix* to apply a 3D BSpline registration on the ROIs. In order to call it, a build of *SimpleITK* that includes *Simple Elastix* must be present. Registered file and transform are saved in the patient subdirectory within the mask root directory. It takes as input a single argument, which is a directory. This must contain the fixed and moving images, that must be named as *no_previous_mask_init.nii* and *macro_mask_init.nii* and both come from manual preprocessing. If for any reason the file is not proper for processing, a *note.txt* file mentioning the reason for this insufficiency shall also reside within the directory. In that case, where a *note.txt* file is also present, no registration is attempted.

**Parameters** **directory** (*string*) –

## FEATURE EXTRACTION ROUTINE

main1.**apply_np_mask**(*image*, *mask*)

**Apply a mask on an image. Before this function is called, it is adviseable to call change_to_nifti.set_masks_world** as it is possible that some mask files and MRI image files have different world masks and this will raise an exception.

**Parameters**

- **image file image** (*itk*) – MRI itk image file
- **image file mask** (*itk*) – mask itk image file on the specific MRI image file (which means that if MRI image is a DWI and the mask is extracted on a T2 image it will raise an exception.)

**Returns** circumscribed rectangular volume of interest on which features will be calculated. As the prostate is a small proportion of the entire image and feature calculation is computationally expensive, this speeds up excecution a lot.

`main1.calculate_features(`*img, rois, patient_directory, resolution, features_switch=['f1', 'f2', 'f3', 'f4', 'f5'], sliding_window=False*`)`

Calculates features for an image slice. All functions that implement the selected feature calculation in the selected way are called depending on the inputs.

**Parameters**

- **`numpy array img`** (*2d*) – Image slice
- **`numpy array rois`** (*2d*) – Regions of interest on the image slice labeled 0 for healthy and 1 for lesion
- **`patient_directory`** – Where to save feature names description files
- **`resolution`** – Image pixel physical spacing (retrieved from the MRI image header information, not avilable in numpy)
- **`features_switch`** (*list*) – Which of the features to calculate (from families C1, C2, C3, C4, C5). Default value is ['f1', 'f2', 'f3', 'f4', 'f5'], which calculates all features.
- **`sliding_window`** (*bool*) – If true, C2 and C5 features are calculated according to the 9x9 sliding window approach, otherwise by the circumscribed rectangle aproach. Default is False.

**Returns** numpy array: 3d numpy array with vectors containing the feature values at each pixel.

`main1.dictionaries_to_arrays(`*features_dictionary, patient_directory, feature_set*`)`

Due to change in design, had to convert all features that are reurned in dictionairies as multi dimensinal np arrays. for exapmple, a 512x 512 slice with 10 C1 featues is now a 512x512x10 array dictionary entries are writen in text files which will indicate feature values sequence. According to these files, features can be retireived later. Each patient folder contains the files that correspond to the calculated features for the specific patient.

Feature name files also contain a dummy feature which is zero everywhere. This is for initialization and it does not correspond to an actual feature.

**Param** dictionary features_dictionary: keys are features names and entries are 2d numpy arrays that contain the value of the specified feature for each pixel in the image.

**Param** string patient_directory: directory where feature name files are saved.

**Param** string feature_set: the set of features (C1, C2, etc) It defines the name of the feature name files as features_names_C1.txt etc

**Returns** numpy array: 3d numpy array with vectors containing the feature values at each pixel.

`main1.dump_patient_folders(`*working_directory, features_switch, sliding_window*`)`

Walk all over the working directory, find files for which features should be calculated and calculate the selected features for them according to the specified method (sliding window or ROI based.) Moreover, a progress file is saved in the working directory. This contains all images about which features have already been calculated. Therefore, if an exception is raised or excecution is halted for any reason, it can reset. It is also possible to expand in more modalities, as progress file's images names' depend on the modality and the patient ID.

**Parameters**

- **`working_directory`** (*string*) – Where all patient directories containing images, masks and ROIs are saved.
- **`features_switch`** (*list*) – Which of the features to calculate (from families C1, C2, C3, C4, C5). Default value is ['f1', 'f2', 'f3', 'f4', 'f5'], which calculates all features.
- **`sliding_window`** (*bool*) – If true, C2 and C5 features are calculated according to the 9x9 sliding window approach, otherwise by the circumscribed rectangle aproach. Default is False.

main1.**main2**(*working_directory, data_root, mask_root, features_switch=['f1', 'f2', 'f3', 'f4', 'f5'], whole_prostate=True, sliding_window=False*)

"Call all dataset preprocessing routines, namely separate_modalities.process_files, change_to_nifti.set_masks_world and then call dump_patient_folders to calculate all features for all files. Also select whether the whole prostate in feature estimation is used or just the peripheral zone. :param string working_directory: Where all patient directories containing images, masks and ROIs are saved. :param string data_root: root directory of the raw dataset :param string mask_root: root directory of the mask and roi files :param list features_switch: Which of the features to calculate (from families C1, C2, C3, C4, C5). Default value is ['f1', 'f2', 'f3', 'f4', 'f5'], which calculates all features. It is important to feed the features switch in order

(eg never give f3, f5,f1 but f1, f3, f5)

**Parameters**

- **whole_prostate** (*bool*) – Whether to use the entire prostate or simply the peripheral zone. Default is True (whole prostate).
- **sliding_window** (*bool*) – If true, C2 and C5 features are calculated according to the 9x9 sliding window approach, otherwise by the circumscribed rectangle aproach. Default is False.

main1.**rescale**(*img, ratio*)

Modified wavelet pyramid scheme to bring an image down to a specific scale. Applies a 2*ratio pixel kernel Gaussian filter to the image to smooth and avoid aliasing. Then input image is sampled on a grid defined by ratio. If ratio is not integer and image pixel indices are :math: *i in [0,N_x]* and :math: *j in [0, N_y]* respectively,then a sampling point is i*ratio[0]. The index according to which its value is retrieved is floor(i*ratio(0)), since the value is constant in the pixel. It is better to use scipy.ndimage.interpolate.zoom function but this implementation is necessary to run the program.

3d images and 3d ratio as input

**Param** 3d numpy array img

**Param** list ratio: ratio[0] corresponds to x sampling ratio, ratio[1] to y sampling ratio and ratio[2] to z. Since this function is made to subsample MRI images, ratio[0]==ratio[1]

**Returns** 3d numpy array: downsampled image

main1.**write_features_to_files**(*patient_directory, image_name, rois, mask, features_switch, sliding_window*)

**Calls calculate_features for each slice trimmed within the volume of interest (thus eliminating pixels both** in z and in x,y directions) and saves a file with the features calculated within the patient directory. The file is in .npy format and it is a 4d numpy array. Features can retrieved by the feature name description files that are saved in the same directory.

**Parameters**

- **patient_directory** (*string*) – where to save the feature files
- **image_name** – MRI image on which features are calculated. This name is used to name after the feature file. Thus, if extended to multiparametric MRI, there can be several feature files in the same directory corresponding to different modality images, as image name is after the sequence description and the patient ID.
- **image file rois** (*itk*) – itk image file for ROIs on the specific MRI image file (which means that if MRI image is a DWI and the mask is extracted on a T2 image it will raise an exception.) 1 refers to a lesion and 0 refers to healthy.
- **image file mask** (*itk*) – itk image file for masks on the specific MRI image file (which means that if MRI image is a DWI and the mask is extracted on a T2 image it will raise an exception.) 1 refers to the prostate or peripheral zone of the prostate volume and 0 refers to healthy.

- **features_switch** (*list*) – Which of the features to calculate (from families C1, C2, C3, C4, C5). Default value is ['f1', 'f2', 'f3', 'f4', 'f5'], which calculates all features.
- **sliding_window** (*bool*) – If true, C2 and C5 features are calculated according to the 9x9 sliding window approach, otherwise by the circumscribed rectangle aproach. Default is False.

## INDICES AND TABLES

- genindex
- modindex
- search

# PYTHON MODULE INDEX

# B

# PCMM DATASET PREPROCESSING

The PCMM dataset contains MRI scans from 3 hospitals. These are the Erasmus MC, NKI and RadboudUMC. Moreover, macroslides of the extirpated prostates and the histology of each macroslide are available for the Erasmus MC cases. The PCMM dataset is developed by a multidisciplinary consortium composed of academic hospitals to address clinical needs in the field of prostate cancer.

## MAPPING THE DATASET

The PCMM dataset contains MRI scans from 3 hospitals. These are the Erasmus MC, NKI and RadboudUMC. Moreover, macroslides of the extirpated prostates and the histology of each macroslide are available for the Erasmus MC cases. The PCMM dataset is developed by a multidisciplinary consortium composed of academic hospitals to address clinical needs in the field of prostate cancer. PCMM dataset primarily consists of 183 folders , each of which concerns a patient that is prostate cancer positive. These folders will from now on be referred as PCMM patient folders. In each of those folders, there are several subfolders, where each one contains DICOM files of multi-parametric MRI images for the specific patient. The number of the sub-folders in each PCMM patient folder varies in a wide range, from approximately 7 (folder 01.0001) to 89 (folder S20920). Moreover, as these folders originate from several different clinics, there is no consistent correspondence between sub-folders and modalities. This results in a vast directory tree, where manual navigation would be both exhausting and error prone.

To this end, an automated routine to make a map of the dataset was developed. This routine takes as input the root directory of the dataset and walks all the way down to the leaves (the DICOM files). Documentation of this routine can be found in Appendix A (*emPROST* documentation, function *separate_modalities.file_list2*). To this level, DICOM tags are read and the information concerning:

- Patient ID
- Pixel Spacing
- Series Description

is kept for further processing. As these tags are identical in each sub-folder, whenever they are read for a single DICOM file we go one level up and continue to the following sub-folder. The output is a text file that for each patient ID contains information about:

- Series Description
- Pixel Spacing
- Directory

A typical entry of this file is presented in Figure 1:

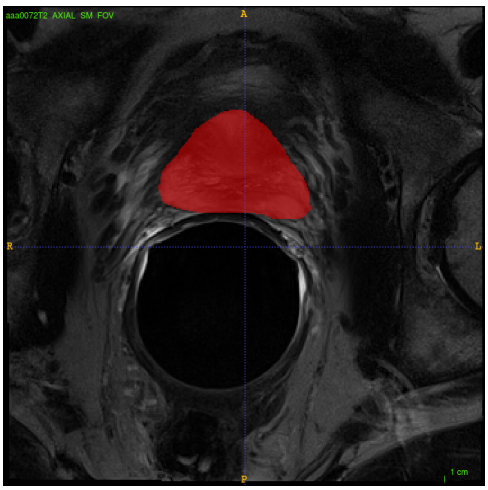
```
PC-2961304084
3Plane Loc&     0.7813& C:/PCMM/01.0001/S10/00001
Calibration Scan&       7.5&    C:/PCMM/01.0001/S20/00001
Sag T2 FRFSE&   0.4688& C:/PCMM/01.0001/S30/00001
Cor T2 FRFSE&   0.4688& C:/PCMM/01.0001/S50/00001
Ax DWI B=1000&  1.25&   C:/PCMM/01.0001/S60/00001
DYN Ax LAVA ARC Freq AP&        1.4063& C:/PCMM/01.0001/S70/00001
Ax T2 FRFSE&    0.4688& C:/PCMM/01.0001/T2/00001
```

Figure B.1: An entry of the file describing the PCMM dataset structure. First row is the patient ID, whereas first column is the series description , second column is the pixel spacing and the third one is the storage directory of the file.

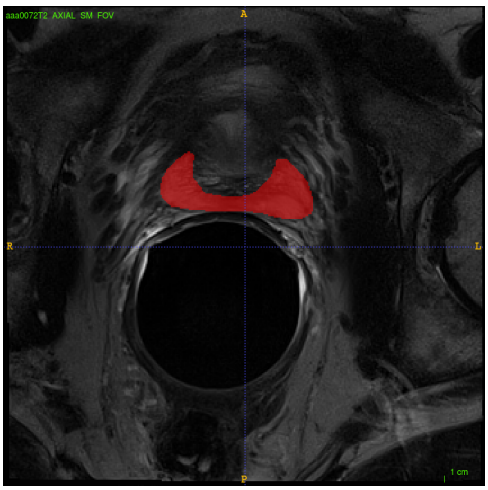
## SEGMENTATION

Since navigation in the dataset is made possible, we proceeded with segmenting the prostate on each axial T2 image. Axial T2 images are chosen, as they have the highest resolution compared to all other modalities and therefore segmentation will be easier and more accurate. Initially, we considered the use of an automated segmentation approach. There are several approaches that report great results in the PROMISE12 competition reports. PROMISE12 (Prostate MRI Image Segmentation 2012) is a competition which exactly fits our case. Out of the top results list [141], VNet by Fausto Milletari [142]("CAMP-TUM2" competition entry) was available on github. Unfortunately, the available version was not trained. Moreover, the reported accuracy (approximately 85 %) might not have been good enough for us. Therefore, we decided to proceed with manually segmenting the images.

The tool used to this purpose was *SegmentationTool3* by Reinhard Hameeteman in MeVisLab. This is not a standard MeVisLab module, but it is only available under BIGR applications (EMC/General). This module works by manually selecting a CSO list on each slice. In the end, the CSO lists are converted to masks of itk meta image format(*.mhd* or *.mha* files). The other option for manual segmentation was *ITK-Snap*. *SegmentationTool3* was however chosen over it for most cases, as it gives the option to save the CSO list which is editable. In our case, this was necessary as the segmentations were later reviewed by two expert radiologists (Ivo Schoots and Razvan Miclea), who advised some modifications in what should be considered as prostate or not. An example of a manually segmented prostate can be seen in figure B.2a, whereas the segmentation of a peripheral zone for the same prostate can be seen in B.2b.



(a) slice of patient TCIA.aaa0072 with a display of the entire prostate segmentation



(b) slice of patient TCIA.aaa0072 with a display of the peripheral zone segmentation

Unfortunately, we noted that the conversion of CSO files to mhd was prone to a bug that resulted in some slices of some masks only covering the contours of the prostate and not the entire surface. We could trace this bug either to the *SegmentationTool3* module (whose status mentions "under development") or to outdated graphics card driver software of the developing machine. Nevertheless, a python module was developed to address this issue.

This module recursively retrieves all mask files stored under a root directory, different than the PCMM root di-

```
PC-3354827525
AXIAL T2&       0.3906& C:/PCMM/S84/S50/00001

PC-2322493792
T2 TSE ax hi&   0.2734375&      C:/PCMM/S8/S5010/00001
```

Figure B.3: An entry of the file describing axial T2 file locations. For each entry, first row is the patient ID, whereas first column is the series description , second column is the pixel spacing and the third one is the storage directory of the file.

rectory (which from now on will be referred as mask root directory), iterates through all slices of all masks and fills all holes of binary images with the standard *scipy* package function *scipy.ndimage.morphology.binary_fill_holes*. Corrected files are subsequently saved in mhd format in a new directory, which will be referred to as the working directory. In total, 130 axial T2 images were segmented in this way.

## AUTOMATIC IMAGE RETRIEVAL

In the file describing the dataset structure, we can see that manual retrieval of the each-time desired modality can again be tiresome and error-prone. Therefore, a routine was developed to avoid manual file selection. This routine works by iterating through the lines of the text file described in section B.1 and makes use of regular expression patterns, to identify if an image Series Description tag matches the specifications of the required modality. A python dictionary of patterns is hard-coded, where each pattern corresponds to a modality description. This dictionary can easily be extended or limited, according to the required information or even to be adjusted to other datasets. A text file similar to the one described in section B.1 is created, containing information about the location of each matching file. This yields a separate file for each modality. By iterating over these files, new sub-folders named as the patient IDs are created- if not already present - within the working directory and the current image is saved in mhd format. The files describing specific modality image directories look as in Figure 2.

## DEFINING THE GROUND TRUTH

In order to proceed with feature evaluation and classification, we need labeling of the data. In literature, manual annotations by one or more radiologists are often mentioned as ground truth labels. Albeit convenient, this method cannot reveal new underlying information as in this case the golden standard is to achieve the performance of the potentially best human. This would limit our conclusions in tumors that are only immediately visible with naked eye. However, the radiomics hypothesis dictates that information about a tumor's phenotype could also be inferred from various computed features, not directly visible to the human eyes.

Therefore, we chose to use histology as our ground truth. Unfortunately, histological results were only available for the part of the PCMM dataset coming from Erasmus MC. The available histological slices had manually drawn tumor contours, with the corresponding Gleason Score (GS) annotations. From previous work, those slices were manually stitched together and the contours from the stitched slices were manually drawn anew on extracted prostate macro slices. This resulted in images like Figure 3:

From those images, from now on referred to as "annotated macros", the following workflow was followed in order to register those to the MRI images:

1. Crop and save each slice shown in the annotated macros in a separate jpeg file with *ImageJ*.
2. Stack all those slices and save in a mhd file. For this task, a *MeVisLab* network was used. This network, undertakes the following tasks:
   (a) As lesions are annotated with red marker, keep only the blue color channel for each of the annotated macro slices.
   (b) Repeat one slice twice to compensate in volume for the missing frozen slice.
   (c) Stack all slices along the z axis.
   (d) Do a manual rigid registration for each of the slices, based on the central slice.
   (e) Save this stack as a single mhd file, from now on referred to as the macro stack.

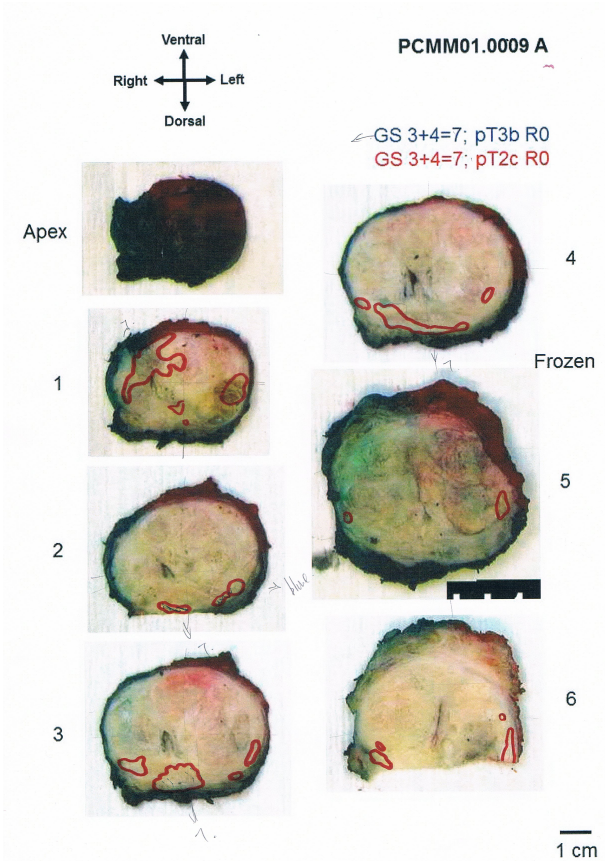



Figure B.4: Tumor contours manually drawn on extracted prostate macros.

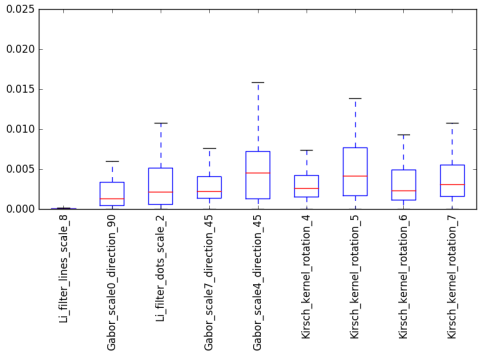
3. Manually segment the prostate region and the lesion contours on the macro stack. Those segmentations are saved in files referred to as the macro masks.
4. Do a manual affine registration (pre-registration) of the macro masks on the axial T2 masks extracted from the procedure described in section B.2. This step brings the macro masks to the same scale as the T2 masks and gives a good initial alignment, that is essential for the next step.
5. Do an automatic BSpline registration with *Elastix* of the pre-registered macro masks on the axial T2 masks. A registration of the macro masks was chosen over a registration of the macro stacks themselves on the axial T2 images, as
   - It is easier and faster for *Elastix* to give good results on binary, rather than gray scale images.
   - Histogram matching and equalization between fixed and moving images would be needed if we went for the original images. Still, the results would be uncertain.
   - A perfect correspondence between regions representing the prostate can be guaranteed. This means that all labeled regions will be inside the region about which features will be subsequently extracted.
   - During registration, annotated regions of interest are expected to undergo the smallest possible distortion, giving us the best we can do for a good correspondence between tumorous and healthy regions.
6. Finally, salt and pepper noise is sometimes observed around the registered regions of interest. This is removed in post processing with a dedicated python function that applies binary erosion and dilation for small connected components, so that true regions of interest stay unaffected.

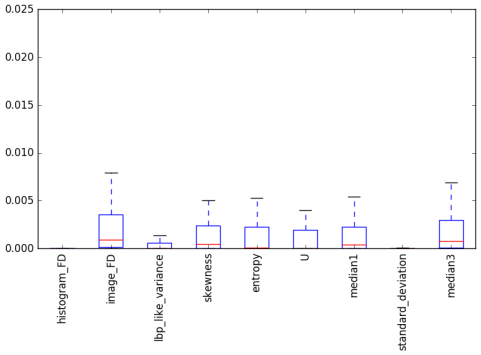
# C

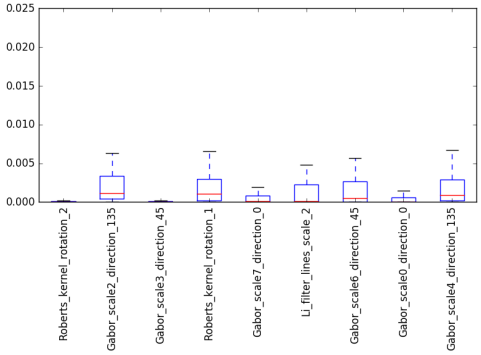
# Feature Significance Ranges

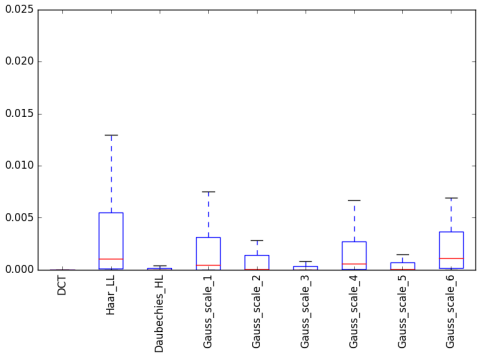
Bellow, the feature significance ranges as computed throughout all relevant experiments through the Random Forest intrinsic feature evaluation are presented. First, the values for a semi-automatic ROI based approach are shown and the sliding window approach estimated values follow.

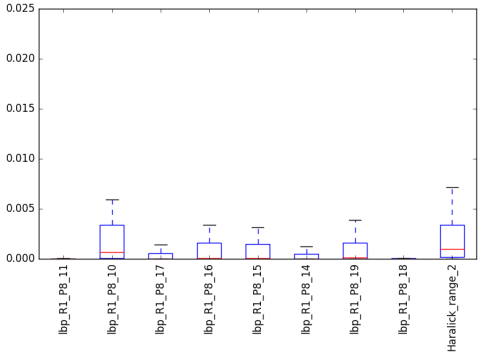
## ROI based estimated values

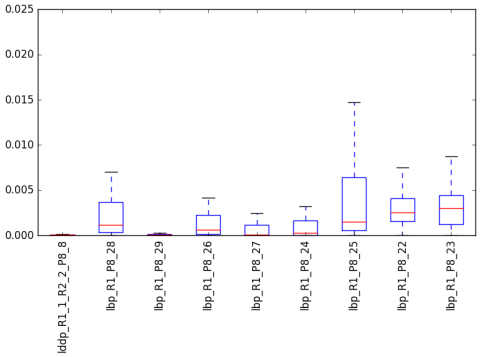

0.025
0.020
0.015
0.010
0.005
0.000
Li_filter_lines_scale_8
Gabor_scale0_direction_90
Li_filter_dots_scale_2
Gabor_scale7_direction_45
Gabor_scale4_direction_45
Kirsch_kernel_rotation_4
Kirsch_kernel_rotation_5
Kirsch_kernel_rotation_6
Kirsch_kernel_rotation_7

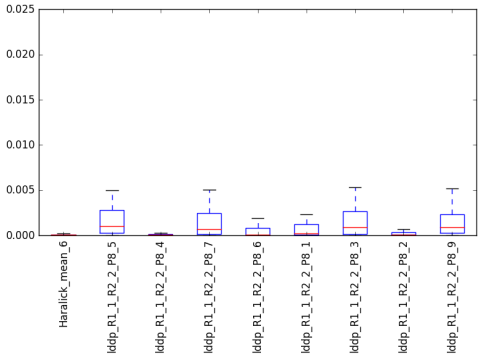

0.025
0.020
0.015
0.010
0.005
0.000
histogram_FD
image_FD
lbp_like_variance
skewness
entropy
U
median1
standard_deviation
median3

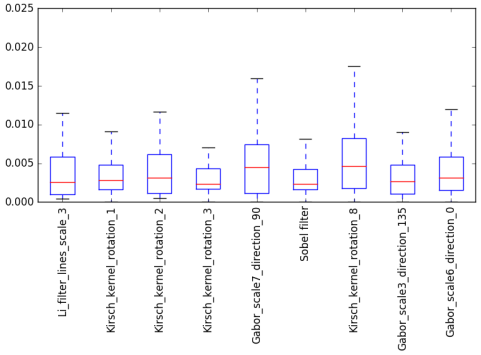

0.025
0.020
0.015
0.010
0.005
0.000
Roberts_kernel_rotation_2
Gabor_scale2_direction_135
Gabor_scale3_direction_45
Roberts_kernel_rotation_1
Gabor_scale7_direction_0
Li_filter_lines_scale_2
Gabor_scale6_direction_45
Gabor_scale0_direction_0
Gabor_scale4_direction_135

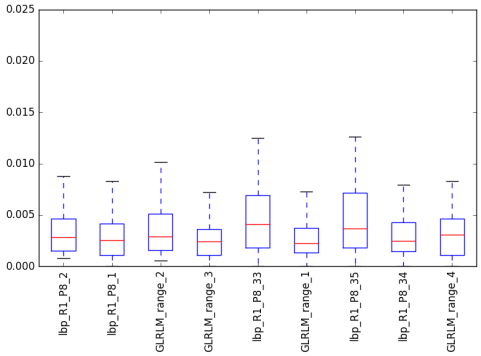

0.025
0.020
0.015
0.010
0.005
0.000
DCT
Haar_LL
Daubechies_HL
Gauss_scale_1
Gauss_scale_2
Gauss_scale_3
Gauss_scale_4
Gauss_scale_5
Gauss_scale_6

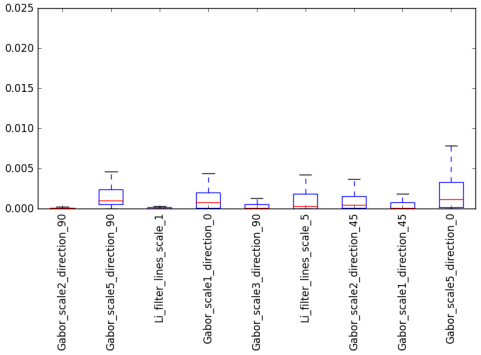

0.025
0.020
0.015
0.010
0.005
0.000
lbp_R1_P8_11
lbp_R1_P8_10
lbp_R1_P8_17
lbp_R1_P8_16
lbp_R1_P8_15
lbp_R1_P8_14
lbp_R1_P8_19
lbp_R1_P8_18
Haralick_range_2

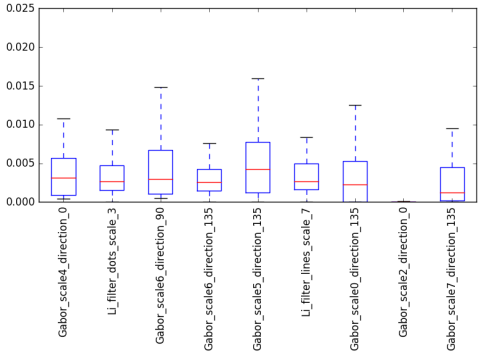

0.025
0.020
0.015
0.010
0.005
0.000
lddp_R1_1_R2_2_P8_8
lbp_R1_P8_28
lbp_R1_P8_29
lbp_R1_P8_26
lbp_R1_P8_27
lbp_R1_P8_24
lbp_R1_P8_25
lbp_R1_P8_22
lbp_R1_P8_23

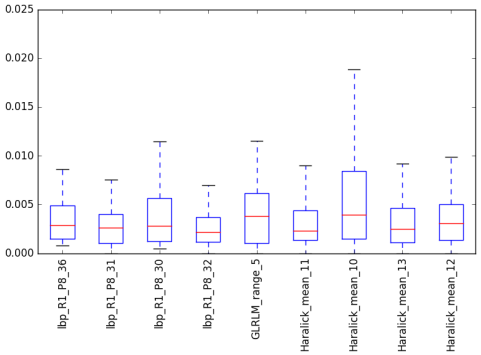

0.025
0.020
0.015
0.010
0.005
0.000
Haralick_mean_6
lddp_R1_1_R2_2_P8_5
lddp_R1_1_R2_2_P8_4
lddp_R1_1_R2_2_P8_7
lddp_R1_1_R2_2_P8_6
lddp_R1_1_R2_2_P8_1
lddp_R1_1_R2_2_P8_3
lddp_R1_1_R2_2_P8_2
lddp_R1_1_R2_2_P8_9

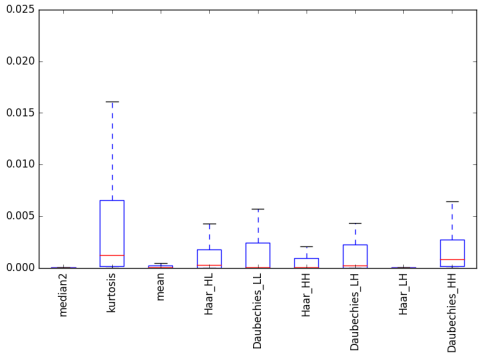

0.025
0.020
0.015
0.010
0.005
0.000
Li_filter_lines_scale_3
Kirsch_kernel_rotation_1
Kirsch_kernel_rotation_2
Kirsch_kernel_rotation_3
Gabor_scale7_direction_90
Sobel filter
Kirsch_kernel_rotation_8
Gabor_scale3_direction_135
Gabor_scale6_direction_0

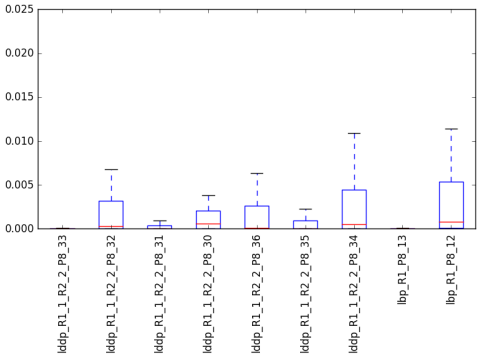

0.025
0.020
0.015
0.010
0.005
0.000
lbp_R1_P8_2
lbp_R1_P8_1
GLRLM_range_2
GLRLM_range_3
lbp_R1_P8_33
GLRLM_range_1
lbp_R1_P8_35
lbp_R1_P8_34
GLRLM_range_4

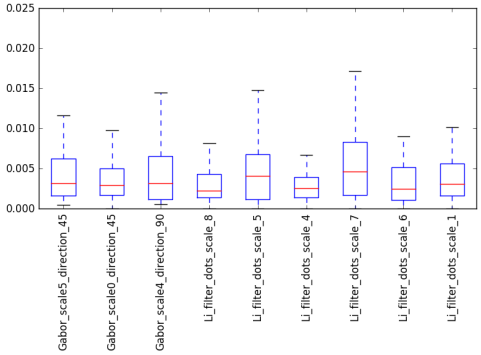

0.025
0.020
0.015
0.010
0.005
0.000
Gabor_scale2_direction_90
Gabor_scale5_direction_90
Li_filter_lines_scale_1
Gabor_scale1_direction_0
Gabor_scale3_direction_90
Li_filter_lines_scale_5
Gabor_scale2_direction_45
Gabor_scale1_direction_45
Gabor_scale5_direction_0

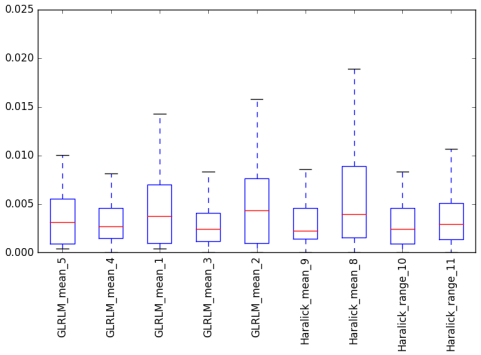

0.025
0.020
0.015
0.010
0.005
0.000
Gabor_scale4_direction_0
Li_filter_dots_scale_3
Gabor_scale6_direction_90
Gabor_scale6_direction_135
Gabor_scale5_direction_135
Li_filter_lines_scale_7
Gabor_scale0_direction_135
Gabor_scale2_direction_0
Gabor_scale7_direction_135

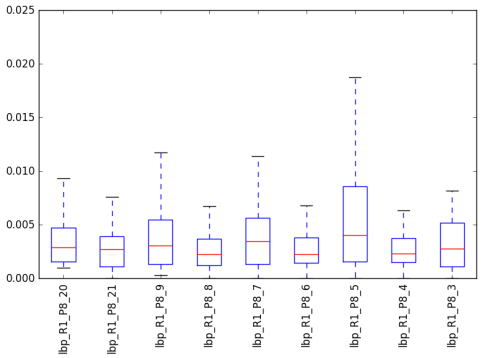

0.025
0.020
0.015
0.010
0.005
0.000
lbp_R1_P8_36
lbp_R1_P8_31
lbp_R1_P8_30
lbp_R1_P8_32
GLRLM_range_5
Haralick_mean_11
Haralick_mean_10
Haralick_mean_13
Haralick_mean_12

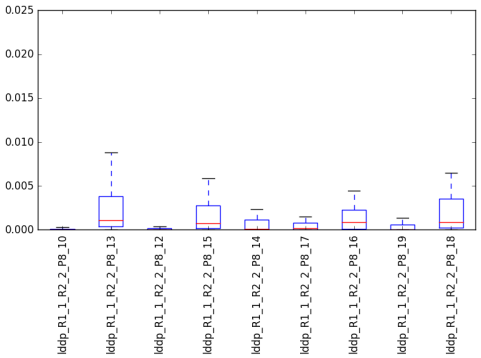

0.025
0.020
0.015
0.010
0.005
0.000
median2
kurtosis
mean
Haar_HL
Daubechies_LL
Haar_HH
Daubechies_LH
Haar_LH
Daubechies_HH

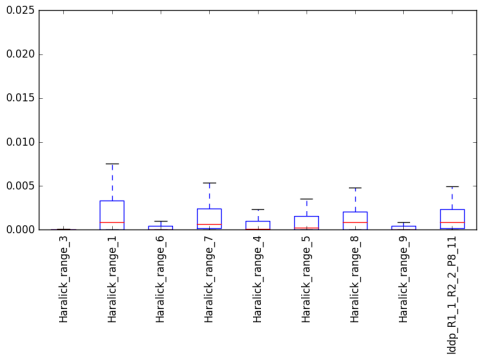

0.025
0.020
0.015
0.010
0.005
0.000
lddp_R1_1_R2_2_P8_33
lddp_R1_1_R2_2_P8_32
lddp_R1_1_R2_2_P8_31
lddp_R1_1_R2_2_P8_30
lddp_R1_1_R2_2_P8_36
lddp_R1_1_R2_2_P8_35
lddp_R1_1_R2_2_P8_34
lbp_R1_P8_13
lbp_R1_P8_12

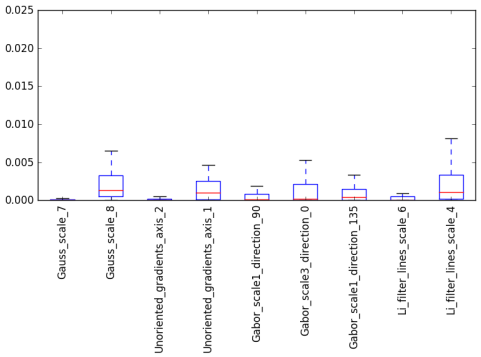

0.025
0.020
0.015
0.010
0.005
0.000
Gabor_scale5_direction_45
Gabor_scale0_direction_45
Gabor_scale4_direction_90
Li_filter_dots_scale_8
Li_filter_dots_scale_5
Li_filter_dots_scale_4
Li_filter_dots_scale_7
Li_filter_dots_scale_6
Li_filter_dots_scale_1

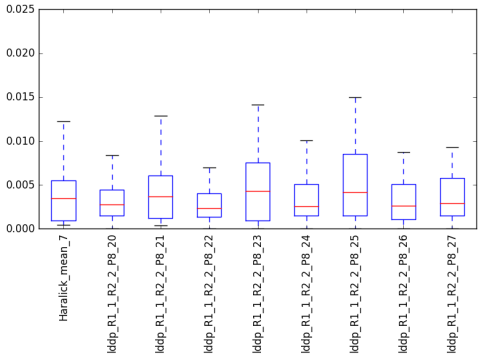

0.025
0.020
0.015
0.010
0.005
0.000
GLRLM_mean_5
GLRLM_mean_4
GLRLM_mean_1
GLRLM_mean_3
GLRLM_mean_2
Haralick_mean_9
Haralick_mean_8
Haralick_range_10
Haralick_range_11

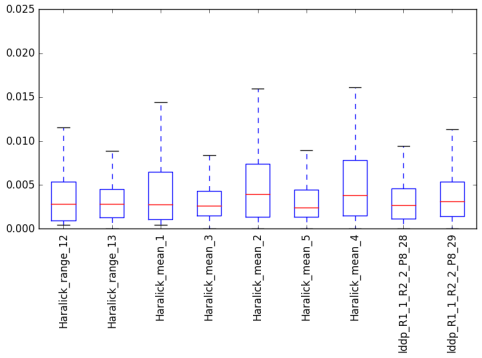

0.025
0.020
0.015
0.010
0.005
0.000
lbp_R1_P8_20
lbp_R1_P8_21
lbp_R1_P8_9
lbp_R1_P8_8
lbp_R1_P8_7
lbp_R1_P8_6
lbp_R1_P8_5
lbp_R1_P8_4
lbp_R1_P8_3


0.025
0.020
0.015
0.010
0.005
0.000
lddp_R1_1_R2_2_P8_10
lddp_R1_1_R2_2_P8_13
lddp_R1_1_R2_2_P8_12
lddp_R1_1_R2_2_P8_15
lddp_R1_1_R2_2_P8_14
lddp_R1_1_R2_2_P8_17
lddp_R1_1_R2_2_P8_16
lddp_R1_1_R2_2_P8_19
lddp_R1_1_R2_2_P8_18

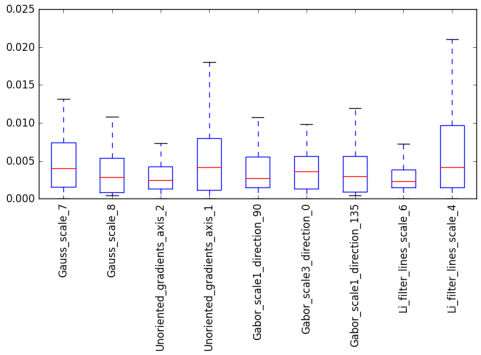

0.025
0.020
0.015
0.010
0.005
0.000
Haralick_range_3
Haralick_range_1
Haralick_range_6
Haralick_range_7
Haralick_range_4
Haralick_range_5
Haralick_range_8
Haralick_range_9
lddp_R1_1_R2_2_P8_11

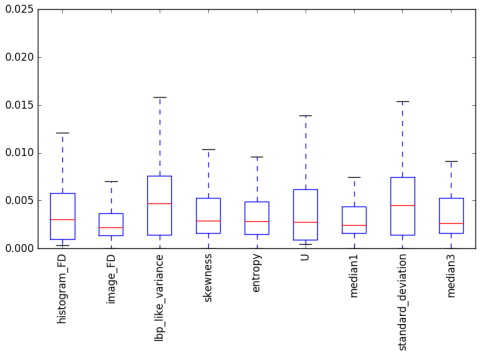

0.025
0.020
0.015
0.010
0.005
0.000
Gauss_scale_7
Gauss_scale_8
Unoriented_gradients_axis_2
Unoriented_gradients_axis_1
Gabor_scale1_direction_90
Gabor_scale3_direction_0
Gabor_scale1_direction_135
Li_filter_lines_scale_6
Li_filter_lines_scale_4

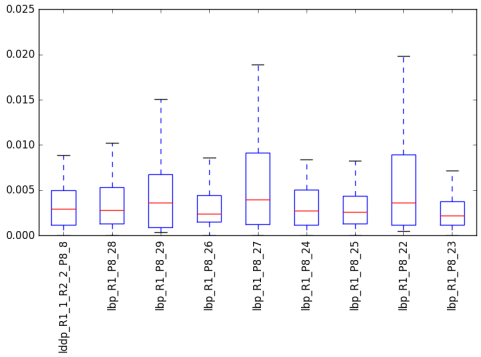

0.025
0.020
0.015
0.010
0.005
0.000
Haralick_mean_7
lddp_R1_1_R2_2_P8_20
lddp_R1_1_R2_2_P8_21
lddp_R1_1_R2_2_P8_22
lddp_R1_1_R2_2_P8_23
lddp_R1_1_R2_2_P8_24
lddp_R1_1_R2_2_P8_25
lddp_R1_1_R2_2_P8_26
lddp_R1_1_R2_2_P8_27

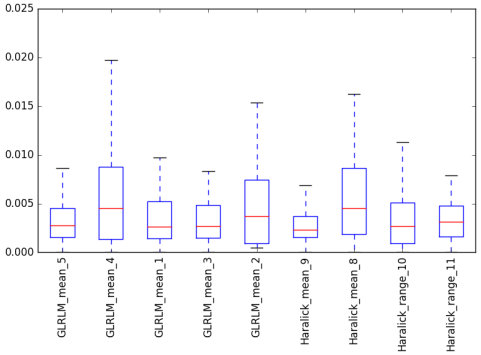

0.025
0.020
0.015
0.010
0.005
0.000
Haralick_range_12
Haralick_range_13
Haralick_mean_1
Haralick_mean_3
Haralick_mean_2
Haralick_mean_5
Haralick_mean_4
lddp_R1_1_R2_2_P8_28
lddp_R1_1_R2_2_P8_29

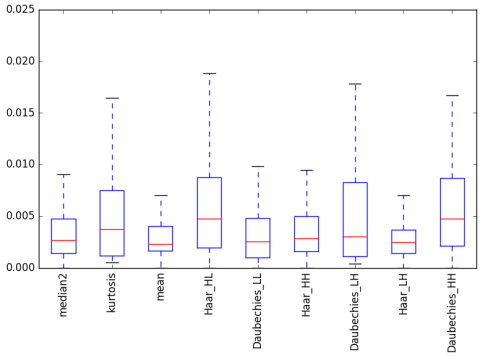
## SLIDING WINDOW ESTIMATED VALUES

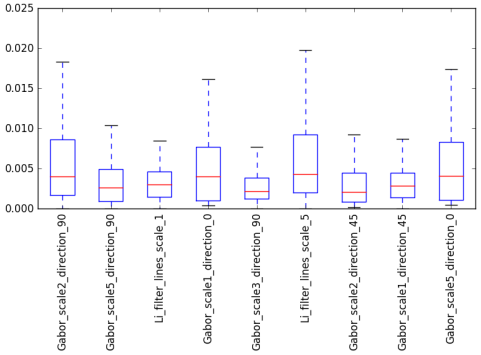

0.025
0.020
0.015
0.010
0.005
0.000
Gauss_scale_7
Gauss_scale_8
Unoriented_gradients_axis_2
Unoriented_gradients_axis_1
Gabor_scale1_direction_90
Gabor_scale3_direction_0
Gabor_scale1_direction_135
Li_filter_lines_scale_6
Li_filter_lines_scale_4

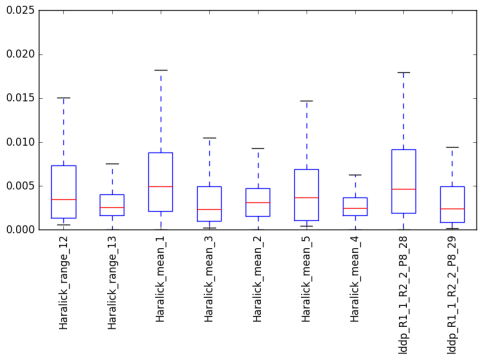

0.025
0.020
0.015
0.010
0.005
0.000
histogram_FD
image_FD
lbp_like_variance
skewness
entropy
U
median1
standard_deviation
median3

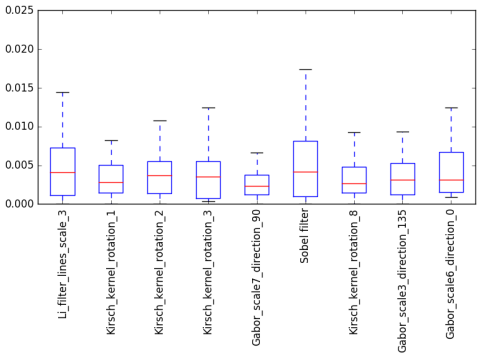

0.025
0.020
0.015
0.010
0.005
0.000
lddp_R1_1_R2_2_P8_8
lbp_R1_P8_28
lbp_R1_P8_29
lbp_R1_P8_26
lbp_R1_P8_27
lbp_R1_P8_24
lbp_R1_P8_25
lbp_R1_P8_22
lbp_R1_P8_23

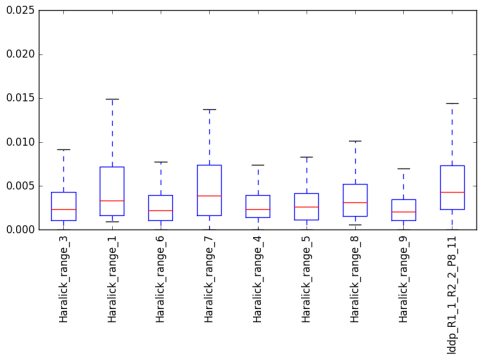

0.025
0.020
0.015
0.010
0.005
0.000
GLRLM_mean_5
GLRLM_mean_4
GLRLM_mean_1
GLRLM_mean_3
GLRLM_mean_2
Haralick_mean_9
Haralick_mean_8
Haralick_range_10
Haralick_range_11

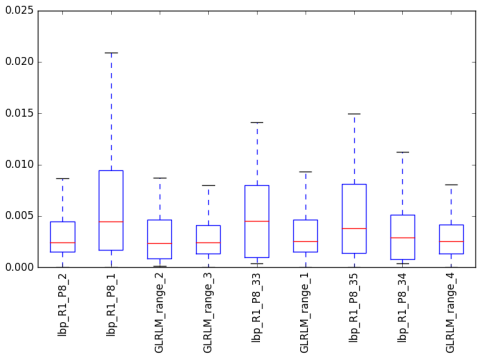

0.025
0.020
0.015
0.010
0.005
0.000
median2
kurtosis
mean
Haar_HL
Daubechies_LL
Haar_HH
Daubechies_LH
Haar_LH
Daubechies_HH

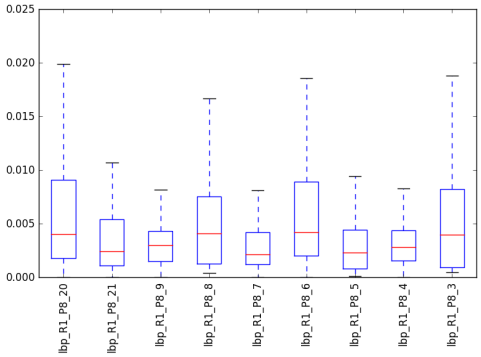

0.025
0.020
0.015
0.010
0.005
0.000
Gabor_scale2_direction_90
Gabor_scale5_direction_90
Li_filter_lines_scale_1
Gabor_scale1_direction_0
Gabor_scale3_direction_90
Li_filter_lines_scale_5
Gabor_scale2_direction_45
Gabor_scale1_direction_45
Gabor_scale5_direction_0

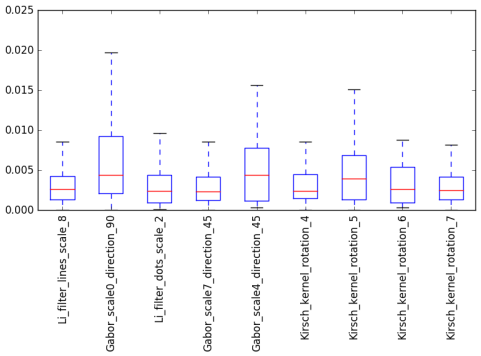

0.025
0.020
0.015
0.010
0.005
0.000
Haralick_range_12
Haralick_range_13
Haralick_mean_1
Haralick_mean_3
Haralick_mean_2
Haralick_mean_5
Haralick_mean_4
lddp_R1_1_R2_2_P8_28
lddp_R1_1_R2_2_P8_29

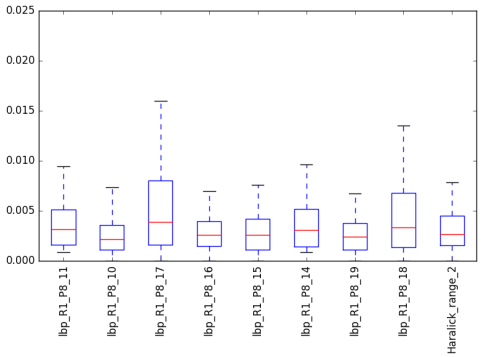

0.025
0.020
0.015
0.010
0.005
0.000
Li_filter_lines_scale_3
Kirsch_kernel_rotation_1
Kirsch_kernel_rotation_2
Kirsch_kernel_rotation_3
Gabor_scale7_direction_90
Sobel filter
Kirsch_kernel_rotation_8
Gabor_scale3_direction_135
Gabor_scale6_direction_0

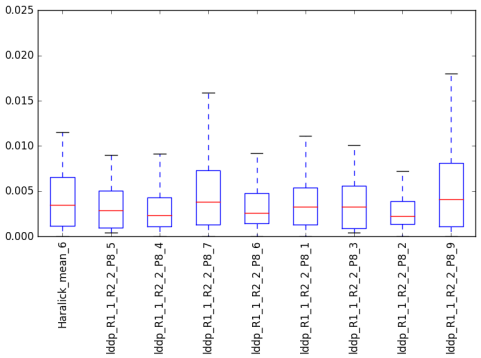

0.025
0.020
0.015
0.010
0.005
0.000
Haralick_range_3
Haralick_range_1
Haralick_range_6
Haralick_range_7
Haralick_range_4
Haralick_range_5
Haralick_range_8
Haralick_range_9
lddp_R1_1_R2_2_P8_11

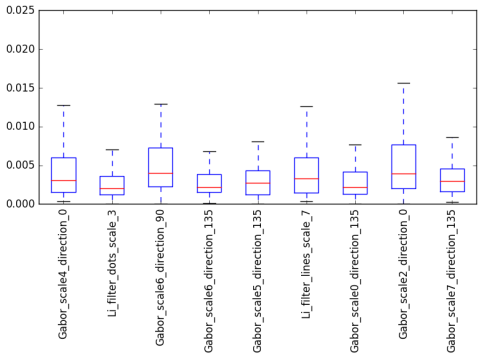

0.025
0.020
0.015
0.010
0.005
0.000
lbp_R1_P8_2
lbp_R1_P8_1
GLRLM_range_2
GLRLM_range_3
lbp_R1_P8_33
GLRLM_range_1
lbp_R1_P8_35
lbp_R1_P8_34
GLRLM_range_4

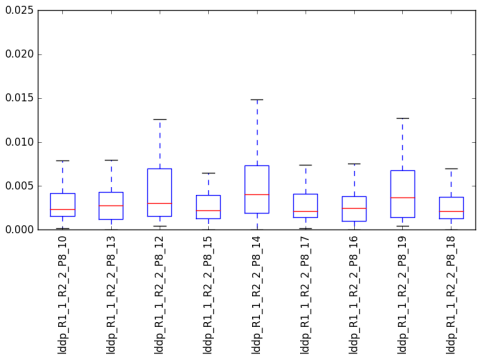

0.025
0.020
0.015
0.010
0.005
0.000
lbp_R1_P8_20
lbp_R1_P8_21
lbp_R1_P8_9
lbp_R1_P8_8
lbp_R1_P8_7
lbp_R1_P8_6
lbp_R1_P8_5
lbp_R1_P8_4
lbp_R1_P8_3

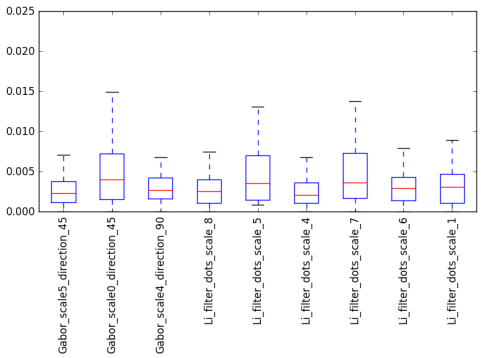

0.025
0.020
0.015
0.010
0.005
0.000
Li_filter_lines_scale_8
Gabor_scale0_direction_90
Li_filter_dots_scale_2
Gabor_scale7_direction_45
Gabor_scale4_direction_45
Kirsch_kernel_rotation_4
Kirsch_kernel_rotation_5
Kirsch_kernel_rotation_6
Kirsch_kernel_rotation_7

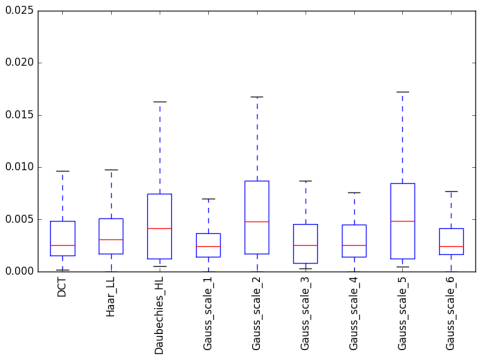

0.025
0.020
0.015
0.010
0.005
0.000
lbp_R1_P8_11
lbp_R1_P8_10
lbp_R1_P8_17
lbp_R1_P8_16
lbp_R1_P8_15
lbp_R1_P8_14
lbp_R1_P8_19
lbp_R1_P8_18
Haralick_range_2

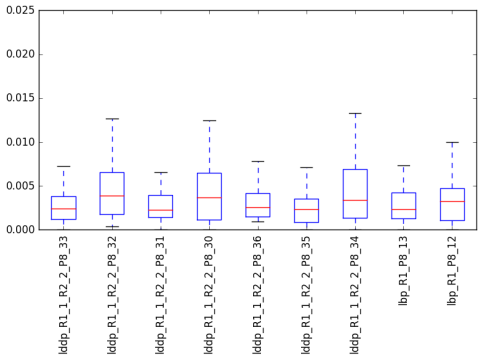

0.025
0.020
0.015
0.010
0.005
0.000
Haralick_mean_6
lddp_R1_1_R2_2_P8_5
lddp_R1_1_R2_2_P8_4
lddp_R1_1_R2_2_P8_7
lddp_R1_1_R2_2_P8_6
lddp_R1_1_R2_2_P8_1
lddp_R1_1_R2_2_P8_3
lddp_R1_1_R2_2_P8_2
lddp_R1_1_R2_2_P8_9

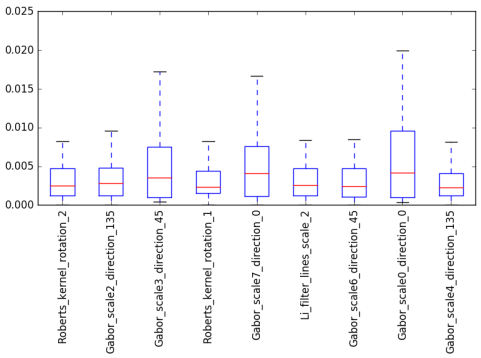

0.025
0.020
0.015
0.010
0.005
0.000
Gabor_scale4_direction_0
Li_filter_dots_scale_3
Gabor_scale6_direction_90
Gabor_scale6_direction_135
Gabor_scale5_direction_135
Li_filter_lines_scale_7
Gabor_scale0_direction_135
Gabor_scale2_direction_0
Gabor_scale7_direction_135

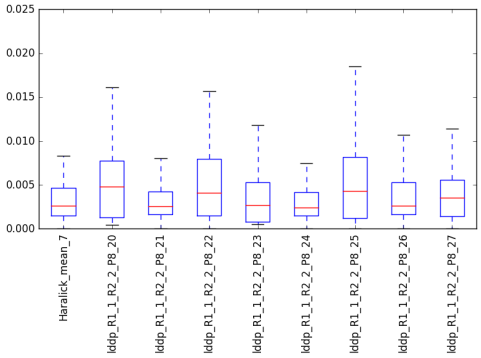

0.025
0.020
0.015
0.010
0.005
0.000
lddp_R1_1_R2_2_P8_10
lddp_R1_1_R2_2_P8_13
lddp_R1_1_R2_2_P8_12
lddp_R1_1_R2_2_P8_15
lddp_R1_1_R2_2_P8_14
lddp_R1_1_R2_2_P8_17
lddp_R1_1_R2_2_P8_16
lddp_R1_1_R2_2_P8_19
lddp_R1_1_R2_2_P8_18

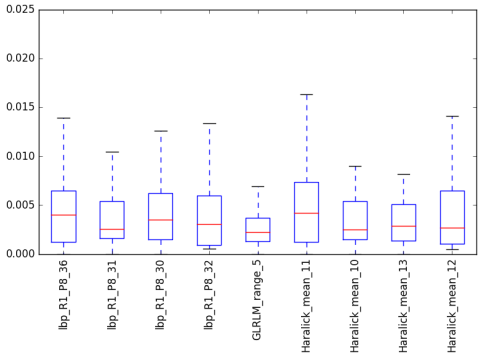

0.025
0.020
0.015
0.010
0.005
0.000
Gabor_scale5_direction_45
Gabor_scale0_direction_45
Gabor_scale4_direction_90
Li_filter_dots_scale_8
Li_filter_dots_scale_5
Li_filter_dots_scale_4
Li_filter_dots_scale_7
Li_filter_dots_scale_6
Li_filter_dots_scale_1


0.025
0.020
0.015
0.010
0.005
0.000
DCT
Haar_LL
Daubechies_HL
Gauss_scale_1
Gauss_scale_2
Gauss_scale_3
Gauss_scale_4
Gauss_scale_5
Gauss_scale_6


0.025
0.020
0.015
0.010
0.005
0.000
lddp_R1_1_R2_2_P8_33
lddp_R1_1_R2_2_P8_32
lddp_R1_1_R2_2_P8_31
lddp_R1_1_R2_2_P8_30
lddp_R1_1_R2_2_P8_36
lddp_R1_1_R2_2_P8_35
lddp_R1_1_R2_2_P8_34
lbp_R1_P8_13
lbp_R1_P8_12


0.025
0.020
0.015
0.010
0.005
0.000
Roberts_kernel_rotation_2
Gabor_scale2_direction_135
Gabor_scale3_direction_45
Roberts_kernel_rotation_1
Gabor_scale7_direction_0
Li_filter_lines_scale_2
Gabor_scale6_direction_45
Gabor_scale0_direction_0
Gabor_scale4_direction_135


0.025
0.020
0.015
0.010
0.005
0.000
Haralick_mean_7
lddp_R1_1_R2_2_P8_20
lddp_R1_1_R2_2_P8_21
lddp_R1_1_R2_2_P8_22
lddp_R1_1_R2_2_P8_23
lddp_R1_1_R2_2_P8_24
lddp_R1_1_R2_2_P8_25
lddp_R1_1_R2_2_P8_26
lddp_R1_1_R2_2_P8_27


0.025
0.020
0.015
0.010
0.005
0.000
lbp_R1_P8_36
lbp_R1_P8_31
lbp_R1_P8_30
lbp_R1_P8_32
GLRLM_range_5
Haralick_mean_11
Haralick_mean_10
Haralick_mean_13
Haralick_mean_12

# BIBLIOGRAPHY


[1] W. Kandinsky, *VERTIEFTE REGUNG (DEEPENED IMPULSE)* (Deutsche Kunstausstellung, 1928, no. 477, 1928).

[2] A. Jemal, F. Bray, M. M. Center, J. Ferlay, E. Ward, and D. Forman, *Global cancer statistics,* CA: a cancer journal for clinicians **61**, 69 (2011).

[3] K. Hodge, J. McNeal, M. Terris, and T. Stamey, *Random systematic versus directed ultrasound guided transrectal core biopsies of the prostate,* The Journal of urology **142**, 71—4; discussion 74—5 (1989).

[4] G. Carneiro, L. Oakden-Rayner, A. P. Bradley, J. Nascimento, and L. Palmer, *Automated 5-year mortality prediction using deep learning and radiomics features from chest computed tomography,* arXiv preprint arXiv:1607.00267 (2016).

[5] P.-P. Ypsilantis, M. Siddique, H.-M. Sohn, A. Davies, G. Cook, V. Goh, and G. Montana, *Predicting response to neoadjuvant chemotherapy with pet imaging using convolutional neural networks,* PloS one **10**, e0137036 (2015).

[6] T. P. Coroller, P. Grossmann, Y. Hou, E. R. Velazquez, R. T. Leijenaar, G. Hermann, P. Lambin, B. Haibe-Kains, R. H. Mak, and H. J. Aerts, *Ct-based radiomic signature predicts distant metastasis in lung adenocarcinoma,* Radiotherapy and Oncology **114**, 345 (2015).

[7] C. A. Livasy, G. Karaca, R. Nanda, M. S. Tretiakova, O. I. Olopade, D. T. Moore, and C. M. Perou, *Phenotypic evaluation of the basal-like subtype of invasive breast carcinoma,* Modern pathology **19**, 264 (2006).

[8] P. Lambin, E. Rios-Velazquez, R. Leijenaar, S. Carvalho, R. G. van Stiphout, P. Granton, C. M. Zegers, R. Gillies, R. Boellard, A. Dekker, *et al.*, *Radiomics: extracting more information from medical images using advanced feature analysis,* European journal of cancer **48**, 441 (2012).

[9] K. M. Panth, R. T. Leijenaar, S. Carvalho, N. G. Lieuwes, A. Yaromina, L. Dubois, and P. Lambin, *Is there a causal relationship between genetic changes and radiomics-based image features? an in vivo preclinical experiment with doxycycline inducible gadd34 tumor cells,* Radiotherapy and Oncology **116**, 462 (2015).

[10] B. T. Larkin, T. H. Berquist, and D. C. Utz, *Evaluation of the prostate by magnetic resonance imaging,* Magnetic resonance imaging **4**, 53 (1986).

[11] R. Stoyanova, M. Takhar, Y. Tschudi, J. C. Ford, G. Solórzano, N. Erho, Y. Balagurunathan, S. Punnen, E. Davicioni, R. J. Gillies, *et al.*, *Prostate cancer radiomics and the promise of radiogenomics,* Translational Cancer Research **5**, 432 (2016).

[12] P. Tiwari, J. Kurhanewicz, and A. Madabhushi, *Multi-kernel graph embedding for detection, gleason grading of prostate cancer via mri/mrs,* Medical image analysis **17**, 219 (2013).

[13] A. Madabhushi, M. D. Feldman, D. N. Metaxas, J. Tomaszeweski, and D. Chute, *Automated detection of prostatic adenocarcinoma from high-resolution ex vivo mri,* IEEE transactions on medical imaging **24**, 1611 (2005).

[14] H. J. Aerts, E. R. Velazquez, R. T. Leijenaar, C. Parmar, P. Grossmann, S. Carvalho, J. Bussink, R. Monshouwer, B. Haibe-Kains, D. Rietveld, *et al.*, *Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach,* Nature communications **5** (2014).

[15] R. T. Leijenaar, G. Nalbantov, S. Carvalho, W. J. van Elmpt, E. G. Troost, R. Boellaard, H. J. Aerts, R. J. Gillies, and P. Lambin, *The effect of suv discretization in quantitative fdg-pet radiomics: the need for standardized methodology in tumor texture analysis,* Scientific reports **5** (2015).

[16] R. T. Leijenaar, S. Carvalho, E. R. Velazquez, W. J. Van Elmpt, C. Parmar, O. S. Hoekstra, C. J. Hoekstra, R. Boellaard, A. L. Dekker, R. J. Gillies, *et al.*, *Stability of fdg-pet radiomics features: an integrated analysis of test-retest and inter-observer variability,* Acta Oncologica **52**, 1391 (2013).

[17] D. Fehr, H. Veeraraghavan, A. Wibmer, T. Gondo, K. Matsumoto, H. A. Vargas, E. Sala, H. Hricak, and J. O. Deasy, *Automatic classification of prostate cancer gleason scores from multiparametric magnetic resonance images,* Proceedings of the National Academy of Sciences **112**, E6265 (2015).

[18] A. Cameron, F. Khalvati, M. A. Haider, and A. Wong, *Maps: a quantitative radiomics approach for prostate cancer detection,* IEEE Transactions on Biomedical Engineering **63**, 1145 (2016).

[19] C. Parmar, R. T. Leijenaar, P. Grossmann, E. R. Velazquez, J. Bussink, D. Rietveld, M. M. Rietbergen, B. Haibe-Kains, P. Lambin, and H. J. Aerts, *Radiomic feature clusters and prognostic signatures specific for lung and head & neck cancer,* Scientific reports **5**, 11044 (2015).

[20] J. Wu, S. Poehlman, M. D. Noseworthy, and M. V. Kamath, *Texture feature based automated seeded region growing in abdominal mri segmentation,* in *2008 International Conference on BioMedical Engineering and Informatics*, Vol. 2 (IEEE, 2008) pp. 263–267.

[21] C. Parmar, E. R. Velazquez, R. Leijenaar, M. Jermoumi, S. Carvalho, R. H. Mak, S. Mitra, B. U. Shankar, R. Kikinis, B. Haibe-Kains, *et al.*, *Robust radiomics feature quantification using semiautomatic volumetric segmentation,* PLoS One **9**, e102107 (2014).

[22] J. Shi and J. Malik, *Normalized cuts and image segmentation,* IEEE Transactions on pattern analysis and machine intelligence **22**, 888 (2000).

[23] P. C. Vos, T. Hambrock, J. O. Barenstz, and H. J. Huisman, *Computer-assisted analysis of peripheral zone prostate lesions using t2-weighted and dynamic contrast enhanced t1-weighted mri,* Physics in medicine and biology **55**, 1719 (2010).

[24] E. Niaf, O. Rouvière, F. Mège-Lechevallier, F. Bratan, and C. Lartizien, *Computer-aided diagnosis of prostate cancer in the peripheral zone using multiparametric mri,* Physics in medicine and biology **57**, 3833 (2012).

[25] S. Viswanath, B. N. Bloch, M. Rosen, J. Chappelow, R. Toth, N. Rofsky, R. Lenkinski, E. Genega, A. Kalyanpur, and A. Madabhushi, *Integrating structural and functional imaging for computer assisted detection of prostate cancer on multi-protocol in vivo 3 tesla mri,* in *SPIE medical imaging* (International Society for Optics and Photonics, 2009) pp. 72603I–72603I.

[26] A. Rampun, L. Zheng, P. Malcolm, B. Tiddeman, and R. Zwiggelaar, *Computer-aided detection of prostate cancer in t2-weighted mri within the peripheral zone.* Physics in medicine and biology **61**, 4796 (2016).

[27] F. Khalvati, A. Wong, and M. A. Haider, *Automated prostate cancer detection via comprehensive multiparametric magnetic resonance imaging texture feature models,* BMC medical imaging **15**, 27 (2015).

[28] S. Viswanath, P. Tiwari, J. Chappelow, R. Toth, J. Kurhanewicz, and A. Madabhushi, *Cadonc: An integrated toolkit for evaluating radiation therapy related changes in the prostate using multiparametric mri,* in *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro* (IEEE, 2011) pp. 2095–2098.

[29] N. Betrouni, N. Makni, S. Lakroum, S. Mordon, A. Villers, and P. Puech, *Computer-aided analysis of prostate multiparametric mr images: an unsupervised fusion-based approach,* International journal of computer assisted radiology and surgery **10**, 1515 (2015).

[30] R. Lopes, A. Ayache, N. Makni, P. Puech, A. Villers, S. Mordon, and N. Betrouni, *Prostate cancer characterization on mr images using fractal features,* Medical physics **38**, 83 (2011).

[31] I. Chan, W. Wells III, R. V. Mulkern, S. Haker, J. Zhang, K. H. Zou, S. E. Maier, and C. M. Tempany, *Detection of prostate cancer by integration of line-scan diffusion, t2-mapping and t2-weighted magnetic resonance imaging; a multichannel statistical classifier,* Medical physics **30**, 2390 (2003).

[32] D. Lv, X. Guo, X. Wang, J. Zhang, and J. Fang, *Computerized characterization of prostate cancer by fractal analysis in mr images,* Journal of magnetic resonance imaging **30**, 161 (2009).

[33] X. Liu, D. L. Langer, M. A. Haider, Y. Yang, M. N. Wernick, and I. S. Yetik, *Prostate cancer segmentation with simultaneous estimation of markov random field parameters and class,* IEEE Transactions on Medical Imaging **28**, 906 (2009).

[34] F. Khalvati, A. Modhafar, A. Cameron, A. Wong, and M. A. Haider, *A multi-parametric diffusion magnetic resonance imaging texture feature model for prostate cancer analysis,* in *Computational Diffusion MRI* (Springer, 2014) pp. 79–88.

[35] S. E. Viswanath, N. B. Bloch, J. C. Chappelow, R. Toth, N. M. Rofsky, E. M. Genega, R. E. Lenkinski, and A. Madabhushi, *Central gland and peripheral zone prostate tumors have significantly different quantitative imaging signatures on 3 tesla endorectal, in vivo t2-weighted mr imagery,* Journal of Magnetic Resonance Imaging **36**, 213 (2012).

[36] D. Ampeliotis, A. Antonakoudi, K. Berberidis, E. Psarakis, and A. Kounoudes, *A computer-aided system for the detection of prostate cancer based on magnetic resonance image analysis,* in *Communications, Control and Signal Processing, 2008. ISCCSP 2008. 3rd International Symposium on* (IEEE, 2008) pp. 1372–1377.

[37] P. Vos, J. Barentsz, N. Karssemeijer, and H. Huisman, *Automatic computer-aided detection of prostate cancer based on multiparametric magnetic resonance image analysis,* Physics in medicine and biology **57**, 1527 (2012).

[38] A. Vignati, S. Mazzetti, V. Giannini, F. Russo, E. Bollito, F. Porpiglia, M. Stasi, and D. Regge, *Texture features on t2-weighted magnetic resonance imaging: new potential biomarkers for prostate cancer aggressiveness,* Physics in medicine and biology **60**, 2685 (2015).

[39] A. Wibmer, H. Hricak, T. Gondo, K. Matsumoto, H. Veeraraghavan, D. Fehr, J. Zheng, D. Goldman, C. Moskowitz, S. W. Fine, *et al.*, *Haralick texture analysis of prostate mri: utility for differentiating noncancerous prostate from prostate cancer and differentiating prostate cancers with different gleason scores,* European radiology **25**, 2840 (2015).

[40] A. B. Rosenkrantz, S. Kim, R. P. Lim, N. Hindman, F.-M. Deng, J. S. Babb, and S. S. Taneja, *Prostate cancer localization using multiparametric mr imaging: comparison of prostate imaging reporting and data system (pi-rads) and likert scales,* Radiology **269**, 482 (2013).

[41] S. Doyle, A. Madabhushi, M. Feldman, and J. Tomaszeweski, *A boosting cascade for automated detection of prostate cancer from digitized histology,* in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Springer, 2006) pp. 504–511.

[42] G. Litjens, R. Elliott, N. Shih, M. Feldman, J. Barentsz, C. Hulsbergen-van de Kaa, I. Kovacs, H. Huisman, and A. Madabhushi, *Distinguishing prostate cancer from benign confounders via a cascaded classifier on multi-parametric mri,* in *SPIE Medical Imaging* (International Society for Optics and Photonics, 2014) pp. 903512–903512.

[43] S. Wang, K. Burtt, B. Turkbey, P. Choyke, and R. M. Summers, *Computer aided-diagnosis of prostate cancer on multiparametric mri: a technical review of current research,* BioMed research international **2014** (2014).

[44] S. E. Seltzer, D. J. Getty, C. Tempany, R. M. Pickett, M. D. Schnall, B. J. McNeil, and J. A. Swets, *Staging prostate cancer with mr imaging: a combined radiologist-computer system.* Radiology **202**, 219 (1997).

[45] G. Collewet, M. Strzelecki, and F. Mariette, *Influence of mri acquisition protocols and image intensity normalization methods on texture classification,* Magnetic Resonance Imaging **22**, 81 (2004).

[46] L. G. Nyu and J. K. Udupa, *On standardizing the mr image intensity scale,* image **1081** (1999).

[47] L. Harrison, P. Dastidar, H. Eskola, R. Järvenpää, H. Pertovaara, T. Luukkaala, P.-L. Kellokumpu-Lehtinen, and S. Soimakallio, *Texture analysis on mri images of non-hodgkin lymphoma,* Computers in biology and medicine **38**, 519 (2008).

[48] Y. Artan, A. Oto, and I. S. Yetik, *Cross-device automated prostate cancer localization with multiparametric mri,* IEEE Transactions on Image Processing **22**, 5385 (2013).

[49] J. Sijbers, A. J. den Dekker, P. Scheunders, and D. Van Dyck, *Maximum-likelihood estimation of rician distribution parameters,* IEEE Trans. Med. Imaging **17**, 357 (1998).

[50] G. Lemaître, R. Martí, J. Freixenet, J. C. Vilanova, P. M. Walker, and F. Meriaudeau, *Computer-aided detection and diagnosis for prostate cancer based on mono and multi-parametric mri: A review,* Computers in biology and medicine **60**, 8 (2015).

[51] J. V. Manjón, J. Carbonell-Caballero, J. J. Lull, G. García-Martí, L. Martí-Bonmatí, and M. Robles, *Mri denoising using non-local means,* Medical image analysis **12**, 514 (2008).

[52] S. Mallat, *A wavelet tour of signal processing* (Academic press, 1999).

[53] R. D. Nowak, *Wavelet-based rician noise removal for magnetic resonance imaging,* IEEE Transactions on Image Processing **8**, 1408 (1999).

[54] N. Wiest-Daesslé, S. Prima, P. Coupé, S. P. Morrissey, and C. Barillot, *Rician noise removal by non-local means filtering for low signal-to-noise ratio mri: applications to dt-mri,* in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Springer, 2008) pp. 171–179.

[55] M. Lysaker, A. Lundervold, and X.-C. Tai, *Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time,* IEEE Transactions on image processing **12**, 1579 (2003).

[56] X. Liu and I. S. Yetik, *Automated prostate cancer localization without the need for peripheral zone extraction using multiparametric mria),* Medical physics **38**, 2986 (2011).

[57] S. Yamazoe, T. Takahara, K. Shimizu, K. Ouchi, T. Mogami, J. Harada, and K. Fukuda, *Diffusion-weighted imaging with relative signal intensity statistical thresholding for delineating prostate cancer tumors,* Magnetic Resonance in Medical Sciences **11**, 1 (2012).

[58] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. Hill, M. O. Leach, and D. J. Hawkes, *Nonrigid registration using free-form deformations: application to breast mr images,* IEEE transactions on medical imaging **18**, 712 (1999).

[59] D. Mattes, D. R. Haynor, H. Vesselle, T. K. Lewellen, and W. Eubank, *Pet-ct image registration in the chest using free-form deformations,* IEEE transactions on medical imaging **22**, 120 (2003).

[60] S. Klein, U. A. van der Heide, I. M. Lips, M. van Vulpen, M. Staring, and J. P. Pluim, *Automatic segmentation of the prostate in 3d mr images by atlas matching using localized mutual information,* Medical physics **35**, 1407 (2008).

[61] S. Martin, J. Troccaz, and V. Daanen, *Automated segmentation of the prostate in 3d mr images using a probabilistic atlas and a spatially constrained deformable model,* Medical physics **37**, 1579 (2010).

[62] J.-P. Thirion, *Image matching as a diffusion process: an analogy with maxwell's demons,* Medical image analysis **2**, 243 (1998).

[63] U. Vovk, F. Pernus, and B. Likar, *A review of methods for correction of intensity inhomogeneity in mri,* IEEE transactions on medical imaging **26**, 405 (2007).

[64] J. G. Sled, A. P. Zijdenbos, and A. C. Evans, *A nonparametric method for automatic correction of intensity nonuniformity in mri data,* IEEE transactions on medical imaging **17**, 87 (1998).

[65] A. Madabhushi and J. K. Udupa, *Interplay between intensity standardization and inhomogeneity correction in mr image processing,* IEEE Transactions on Medical Imaging **24**, 561 (2005).

[66] N. V. Chawla, N. Japkowicz, and A. Kotcz, *Editorial: special issue on learning from imbalanced data sets,* ACM Sigkdd Explorations Newsletter **6**, 1 (2004).

[67] P. Resnik and E. Hardisty, *Gibbs sampling for the uninitiated*, Tech. Rep. (DTIC Document, 2010).

[68] J. O. Barentsz, J. Richenberg, R. Clements, P. Choyke, S. Verma, G. Villeirs, O. Rouviere, V. Logager, and J. J. Fütterer, *Esur prostate mr guidelines 2012,* European radiology **22**, 746 (2012).

[69] C. M. Hoeks, T. Hambrock, D. Yakar, C. A. Hulsbergen-van de Kaa, T. Feuth, J. A. Witjes, J. J. Fütterer, and J. O. Barentsz, *Transition zone prostate cancer: detection and localization with 3-t multiparametric mr imaging,* Radiology **266**, 207 (2013).

[70] A. R. Padhani and J. E. Husband, *Dynamic contrast-enhanced mri studies in oncology with an emphasis on quantification, validation and human studies,* Clinical radiology **56**, 607 (2001).

[71] J. O'Connor, P. Tofts, K. Miles, L. Parkes, G. Thompson, and A. Jackson, *Dynamic contrast-enhanced imaging techniques: Ct and mri,* The British journal of radiology (2014).

[72] H. J. Huisman, M. R. Engelbrecht, and J. O. Barentsz, *Accurate estimation of pharmacokinetic contrast-enhanced dynamic mri parameters of the prostate,* Journal of Magnetic Resonance Imaging **13**, 607 (2001).

[73] C. Roberts, B. Issa, A. Stone, A. Jackson, J. C. Waterton, and G. J. Parker, *Comparative study into the robustness of compartmental modeling and model-free analysis in dce-mri studies,* Journal of Magnetic Resonance Imaging **23**, 554 (2006).

[74] N. F. Haq, P. Kozlowski, E. C. Jones, S. D. Chang, S. L. Goldenberg, and M. Moradi, *A data-driven approach to prostate cancer detection from dynamic contrast enhanced mri,* Computerized Medical Imaging and Graphics **41**, 37 (2015).

[75] G. Litjens, O. Debats, J. Barentsz, N. Karssemeijer, and H. Huisman, *Computer-aided detection of prostate cancer in mri,* IEEE transactions on medical imaging **33**, 1083 (2014).

[76] C. M. Hoeks, J. O. Barentsz, T. Hambrock, D. Yakar, D. M. Somford, S. W. Heijmink, T. W. Scheenen, P. C. Vos, H. Huisman, I. M. van Oort, *et al., Prostate cancer: multiparametric mr imaging for detection, localization, and staging,* Radiology **261**, 46 (2011).

[77] M. Zaider, M. J. Zelefsky, E. K. Lee, K. L. Zakian, H. I. Amols, J. Dyke, G. Cohen, Y.-C. Hu, A. K. Endi, C.-S. Chui, *et al., Treatment planning for prostate implants using magnetic-resonance spectroscopy imaging,* International Journal of Radiation Oncology* Biology* Physics **47**, 1085 (2000).

[78] R. Bammer, *Basic principles of diffusion-weighted imaging,* European journal of radiology **45**, 169 (2003).

[79] A. A. Malayeri, R. H. El Khouli, A. Zaheer, M. A. Jacobs, C. P. Corona-Villalobos, I. R. Kamel, and K. J. Macura, *Principles and applications of diffusion-weighted imaging in cancer detection, staging, and treatment follow-up,* Radiographics **31**, 1773 (2011).

[80] J. Glaister, A. Cameron, A. Wong, and M. A. Haider, *Quantitative investigative analysis of tumour separability in the prostate gland using ultra-high b-value computed diffusion imaging,* in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (IEEE, 2012) pp. 420–423.

[81] A. Wong, J. Glaister, A. Cameron, and M. Haider, *Correlated diffusion imaging,* BMC medical imaging **13**, 26 (2013).

[82] J. T. Kwak, S. Xu, B. J. Wood, B. Turkbey, P. L. Choyke, P. A. Pinto, S. Wang, and R. M. Summers, *Automated prostate cancer detection using t2-weighted and high-b-value diffusion-weighted magnetic resonance imaging,* Medical physics **42**, 2368 (2015).

[83] C. Qian, L. Wang, Y. Gao, A. Yousuf, X. Yang, A. Oto, and D. Shen, *In vivo mri based prostate cancer localization with random forests and auto-context model,* Computerized Medical Imaging and Graphics (2016).

[84] D. Portalez, G. Rollin, P. Leandri, B. Elman, P. Mouly, F. Jonca, and B. Malavaud, *Prospective comparison of t2w-mri and dynamic-contrast-enhanced mri, 3d-mr spectroscopic imaging or diffusion-weighted mri in repeat trus-guided biopsies,* European Radiology **20**, 2781 (2010).

[85] J. L. Prince and J. M. Links, *Medical imaging signals and systems* (Pearson Prentice Hall Upper Saddle River, NJ, 2006).

[86] G. Castellano, L. Bonilha, L. Li, and F. Cendes, *Texture analysis of medical images,* Clinical radiology **59**, 1061 (2004).

[87] G. J. Litjens, J. O. Barentsz, N. Karssemeijer, and H. J. Huisman, *Clinical evaluation of a computer-aided diagnosis system for determining cancer aggressiveness in prostate mri,* European radiology **25**, 3187 (2015).

[88] Q. Li, S. Sone, and K. Doi, *Selective enhancement filters for nodules, vessels, and airway walls in two-and three-dimensional ct scans,* Medical physics **30**, 2040 (2003).

[89] N. Aggarwal and R. Agrawal, *First and second order statistics features for classification of magnetic resonance brain images,* (2012).

[90] Y. Peng, Y. Jiang, C. Yang, J. B. Brown, T. Antic, I. Sethi, C. Schmid-Tannwald, M. L. Giger, S. E. Eggener, and A. Oto, *Quantitative analysis of multiparametric prostate mr images: differentiation between prostate cancer and normal tissue and correlation with gleason score—a computer-aided diagnosis development study,* Radiology **267**, 787 (2013).

[91] R. M. Haralick, K. Shanmugam, *et al., Textural features for image classification,* IEEE Transactions on systems, man, and cybernetics , 610 (1973).

[92] M. M. Galloway, *Texture analysis using gray level run lengths,* Computer graphics and image processing **4**, 172 (1975).

[93] R. A. Kirsch, *Computer determination of the constituent structure of biological images,* Computers and biomedical research **4**, 315 (1971).

[94] T. Ojala, M. Pietikainen, and T. Maenpaa, *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,* IEEE Transactions on pattern analysis and machine intelligence **24**, 971 (2002).

[95] M. R. Turner, *Texture discrimination by gabor functions,* Biological cybernetics **55**, 71 (1986).

[96] R. C. Gonzalez and R. E. Woods, *Digital image processing,* Nueva Jersey (2008).

[97] A. Graps, *An introduction to wavelets,* IEEE computational science and engineering **2**, 50 (1995).

[98] J. Nunez, X. Otazu, O. Fors, A. Prades, V. Pala, and R. Arbiol, *Multiresolution-based image fusion with additive wavelet decomposition,* IEEE Transactions on Geoscience and Remote sensing **37**, 1204 (1999).

[99] D.-M. Tsai and B. Hsiao, *Automatic surface inspection using wavelet reconstruction,* Pattern Recognition **34**, 1285 (2001).

[100] I. Daubechies, *Orthonormal bases of compactly supported wavelets,* Communications on pure and applied mathematics **41**, 909 (1988).

[101] L. Alic, M. van Vliet, C. Van Dijke, A. Eggermont, J. Veenland, and W. Niessen, *Heterogeneity in dce-mri parametric maps: a biomarker for treatment response?* Physics in medicine and biology **56**, 1601 (2011).

[102] B. B. Mandelbrot, *The fractal geometry of nature*, Vol. 173 (Macmillan, 1983).

[103] B. Stark, M. Adams, D. Hathaway, and M. Hagyard, *Evaluation of two fractal methods for magnetogram image analysis,* Solar Physics **174**, 297 (1997).

[104] S. Peleg, J. Naor, R. Hartley, and D. Avnir, *Multiple resolution texture analysis and classification,* IEEE Transactions on Pattern Analysis and Machine Intelligence , 518 (1984).

[105] C. M. Bishop, *Pattern recognition and machine learning* (springer, 2006).

[106] S. Theodoridis, A. Pikrakis, K. Koutroumbas, and D. Cavouras, *Introduction to pattern recognition: a matlab approach* (Academic Press, 2010).

[107] A. Madabhushi, J. Shi, M. Rosen, J. E. Tomaszeweski, and M. D. Feldman, *Graph embedding to improve supervised classification and novel class detection: application to prostate cancer,* in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Springer, 2005) pp. 729–737.

[108] L. v. d. Maaten and G. Hinton, *Visualizing data using t-sne,* Journal of Machine Learning Research **9**, 2579 (2008).

[109] V. Giannini, S. Mazzetti, A. Vignati, F. Russo, E. Bollito, F. Porpiglia, M. Stasi, and D. Regge, *A fully automatic computer aided diagnosis system for peripheral zone prostate cancer detection using multiparametric magnetic resonance imaging,* Computerized Medical Imaging and Graphics **46**, 219 (2015).

[110] H. Peng, F. Long, and C. Ding, *Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy,* IEEE Transactions on pattern analysis and machine intelligence **27**, 1226 (2005).

[111] V. N. Vapnik and V. Vapnik, *Statistical learning theory*, Vol. 1 (Wiley New York, 1998).

[112] L. Breiman, *Random forests,* Machine learning **45**, 5 (2001).

[113] E. Niaf, C. Lartizien, F. Bratan, L. Roche, M. Rabilloud, F. Mège-Lechevallier, and O. Rouvière, *Prostate focal peripheral zone lesions: characterization at multiparametric mr imaging—influence of a computer-aided diagnosis system,* Radiology **271**, 761 (2014).

[114] V. Ojansivu, E. Rahtu, and J. Heikkila, *Rotation invariant local phase quantization for blur insensitive texture analysis,* in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on* (IEEE, 2008) pp. 1–4.

[115] V. Shah, B. Turkbey, H. Mani, Y. Pang, T. Pohida, M. J. Merino, P. A. Pinto, P. L. Choyke, and M. Bernardo, *Decision support system for localizing prostate cancer based on multiparametric magnetic resonance imaging,* Medical physics **39**, 4093 (2012).

[116] J. F. G. Molina, L. Zheng, M. Sertdemir, D. J. Dinter, S. Schönberg, and M. Rädle, *Incremental learning with svm for multimodal classification of prostatic adenocarcinoma,* PloS one **9**, e93600 (2014).

[117] B. Turkbey, V. P. Shah, Y. Pang, M. Bernardo, S. Xu, J. Kruecker, J. Locklin, A. A. Baccala Jr, A. R. Rastinehad, M. J. Merino, *et al.*, *Is apparent diffusion coefficient associated with clinical risk scores for prostate cancers that are visible on 3-t mr images?* Radiology **258**, 488 (2011).

[118] E. Niaf, R. Flamary, O. Rouviere, C. Lartizien, and S. Canu, *Kernel-based learning from both qualitative and quantitative labels: application to prostate cancer diagnosis based on multiparametric mr imaging,* IEEE Transactions on Image Processing **23**, 979 (2014).

[119] S. B. Ginsburg, S. E. Viswanath, B. N. Bloch, N. M. Rofsky, E. M. Genega, R. E. Lenkinski, and A. Madabhushi, *Novel pca-vip scheme for ranking mri protocols and identifying computer-extracted mri measurements associated with central gland and peripheral zone prostate tumors,* Journal of Magnetic Resonance Imaging **41**, 1383 (2015).

[120] J. N. Stember, F.-M. Deng, S. S. Taneja, and A. B. Rosenkrantz, *Pilot study of a novel tool for input-free automated identification of transition zone prostate tumors using t2-and diffusion-weighted signal and textural features,* Journal of Magnetic Resonance Imaging **40**, 301 (2014).

[121] Y. S. Sung, H.-J. Kwon, B.-W. Park, G. Cho, C. K. Lee, K.-S. Cho, and J. K. Kim, *Prostate cancer detection on dynamic contrast-enhanced mri: computer-aided diagnosis versus single perfusion parameter maps,* American Journal of Roentgenology **197**, 1122 (2011).

[122] S. Ozer, D. L. Langer, X. Liu, M. A. Haider, T. H. van der Kwast, A. J. Evans, Y. Yang, M. N. Wernick, and I. S. Yetik, *Supervised and unsupervised methods for prostate cancer segmentation with multispectral mri,* Medical physics **37**, 1873 (2010).

[123] K. Gnep, A. Fargeas, R. E. Gutiérrez-Carvajal, F. Commandeur, R. Mathieu, J. D. Ospina, Y. Rolland, T. Rohou, S. Vincendeau, M. Hatt, *et al.*, *Haralick textural features on t2-weighted mri are associated with biochemical recurrence following radiotherapy for peripheral zone prostate cancer,* Journal of Magnetic Resonance Imaging (2016).

[124] S. Viswanath, R. Toth, M. Rusu, D. Sperling, H. Lepor, J. Futterer, and A. Madabhushi, *Identifying quantitative in vivo multi-parametric mri features for treatment related changes after laser interstitial thermal therapy of prostate cancer,* Neurocomputing **144**, 13 (2014).

[125] A. Prando, *Diffusion-weighted mri of peripheral zone prostate cancer: comparison of tumor apparent diffusion coefficient with gleason score and percentage of tumor on core biopsy,* International braz j urol **36**, 504 (2010).

[126] A. Cameron, A. Modhafar, F. Khalvati, D. Lui, M. J. Shafiee, A. Wong, and M. Haider, *Multiparametric mri prostate cancer analysis via a hybrid morphological-textural model,* in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (IEEE, 2014) pp. 3357–3360.

[127] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle, L. Tarbox, and F. Prior, *The cancer imaging archive (tcia): Maintaining and operating a public information repository,* Journal of Digital Imaging **26**, 1045 (2013).

[128] A. Madabhushi and M. Feldman, *Fused radiology-pathology prostate dataset.* (2016).

[129] A. Singanamalli, M. Rusu, R. E. Sparks, N. N. Shih, A. Ziober, L.-P. Wang, J. Tomaszewski, M. Rosen, M. Feldman, and A. Madabhushi, *Identifying in vivo dce mri markers associated with microvessel architecture and gleason grades of prostate cancer,* Journal of Magnetic Resonance Imaging **43**, 149 (2016).

[130] L. P. Coelho, *Mahotas: Open source software for scriptable computer vision,* arXiv preprint arXiv:1211.4907 (2012).

[131] E. Jones, T. Oliphant, and P. Peterson, {*SciPy*}*: open source scientific tools for* {*Python*}, (2014).

[132] K. J. Millman and M. Aivazis, *Python for scientists and engineers,* Computing in Science & Engineering **13**, 9 (2011).

[133] S. v. d. Walt, S. C. Colbert, and G. Varoquaux, *The numpy array: a structure for efficient numerical computation,* Computing in Science & Engineering **13**, 22 (2011).

[134] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, *scikit-image: image processing in python,* PeerJ **2**, e453 (2014).

[135] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, *Scikit-learn: Machine learning in Python,* Journal of Machine Learning Research **12**, 2825 (2011).

[136] N. Japkowicz, *The class imbalance problem: Significance and strategies,* in *Proc. of the Int'l Conf. on Artificial Intelligence* (2000).

[137] K. Marstal, F. Berendsen, M. Staring, and S. Klein, *Simpleelastix: A user-friendly, multi-lingual library for medical image registration,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2016) pp. 134–142.

[138] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet classification with deep convolutional neural networks,* in *Advances in neural information processing systems* (2012) pp. 1097–1105.

[139] *XNAT prostate mri datasets,* https://central.xnat.org/REST/projects/NCIGT_PROSTATE, accessed: 27-11-2017.

[140] *Quantitative Insights radiomics clinical application,* https://www.prnewswire.com/news-releases/quantitative-insights-gains-industrys-first-fda-clearance-for-machine-learning-driven-cancer-diagnosis-300495405.html, accessed: 27-11-2017.

[141] *Leaderboard 2012,* https://promise12.grand-challenge.org/results/, accessed: 22-05-2017.

[142] F. Milletari, N. Navab, and S.-A. Ahmadi, *V-net: Fully convolutional neural networks for volumetric medical image segmentation,* in *3D Vision (3DV), 2016 Fourth International Conference on* (IEEE, 2016) pp. 565–571.