Supervised Machine Learning Approach Utilizing Artificial Neural Networks for Automated Prostate Zone Segmentation in Abdominal MR images

---

Master Thesis

Hans-Peter Wieser, BS

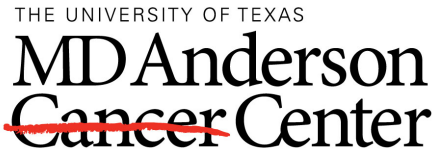
Hans-Peter.Wieser@edu.fh-kaernten.ac.at

ID: 1110310001

Klagenfurt, October 2013

Performed in corporation at the

Fachhochschule Kärnten/Carinthia University of Applied Sciences



The University of Texas MD Anderson Cancer Center

supervised by

Marvin D. Hoffland, MS | Naveen Garg, MD

(Hans-Peter Wieser)

Abstract

Being aware of the internal anatomical prostate zones and their volumes can positively affect and improve multiple clinical fields as prostate cancer staging and treatment selection. Hence, this study presents an fully-automated prostate zone segmentation algorithm from in vivo T2-weighted MRI studies with subsequent volume estimation. The proposed supervised machine learning algorithm constitute multi layer feed forward neural networks in order to solve a multi-class classification problem. To achieve input data clustering, multiple texture, distance, statistical, probabilistic as well as local neighborhood features are extracted. 3D closed surface models are assessed via triangulation from predicted prostate zones. Based on 3D closed surface models, prostate zones volumes are estimated. Optimal neural network parameters have been established, which achieve on 25 3-Tesla studies mean Dice coefficient scores of 0.78 for the central gland and 0.47 for the peripheral zone. The volumes estimated by the proposed algorithm had Person correlation coefficients (r^2 values) of 0.91 and 0.45 when compared to the ground truth. Furthermore, a semi-automated version of the proposed algorithm produces mean Dice coefficient scores on 100 MRI studies of 0.81 and 0.69. Subsequent volume estimation results in mean volume fractions of 1.19 and 0.9. Respectively, r^2 values are 0.91, 0.64. Considering only the prostate gland, the mean volume fraction averages to 1.01 and achieves a r^2 value of 0.92. Summarized, the proposed algorithm enables real time prostate zone segmentation as well as real time prostate zone volume estimation and outperforms state of the art clinical volume estimation techniques as the Myschetzky, Ellipsoid and Prolate spheroid techniques.

Key words: feed forward neural network, machine learning, supervised learning, MRI, fully-automated, prostate zone segmentation, prostate zone volume estimation

Kurzfassung

Diese Arbeit befasst sich mit der automatischen Echtzeit Segmentierung der Prostatazonen in MRT Bildern. Durch die bereitgestellte Segmentierung, kann klinisches Personal bei Prostata relevanten Fragestellungen maßgeblich unterstützt werden. Der vorgestellte Algorithmus basiert auf dem Prinzip des überwachten maschinellen Lernens und beinhaltet künstliche neuronale Netzwerke, welche im verwendeten Algorithmus ein Pixel-Klassifikationsproblem lösen. Zusätzlich ermöglicht das implementierte Programm die Volumenschätzung der segmentieren Prostatazonen. In 100 Studien erzielt der implementierte Algorithmus im Verhältnis zur tatsächlichen Prostatazone eine mittlere Überlappung von 0.81 für die zentrale Prostatazone und 0.69 für die periphere Prostatazone. Eine nachfolgende Volumenschätzung der segmentierten Prostatazonen erreicht im Bezug zu den tatsächlichen Prostatazonenvolumina Pearsonsche Korrelationskoeffizienten von 0.91 für die zentrale- und 0.64 für die periphere Prostatazone.

Suchbegriffe: künstliches neuronales Netzwerk, maschinelles Lernen, überwachtes Lernen, MRT, Prostatazonensegmentierung, Prostatavolumenschätzung

Contents

Chapter 1

Introduction

The human visual cortex (VC) is located in the back of the brain and is responsible for processing visual information, which is received from the optic nerve. The VC enables image understanding and object recognition and is part of the visual sensory system [1], [2]. As the the lateral geniculate nucleus (LGN) in the brain consists of approximately six hierarchical layers, theoretical models assume that visual perception also works in a hierarchical manner [2], [3]. The task of recognizing objects and understanding images is performed by our VC in milliseconds [4]. Humans have learned since their births the ability to learn to recognize objects, e.g., faces. Everyday we are required to use this easily executable and remarkable ability [5]. However, in contrast it is hard to transfer the complex cognitive abilities and model-based knowledge, which are learned over years, to a computer program [6]. Nowadays intense research, including this work, is attempting to develop an automatic algorithm, which acts like the human visual cortex [7], [8], [9], [10], [11], [12], [13].

According to the Organisation for Economic Co-operation and Development (OECD), the number of imaging devices in each country worldwide is increasing steadily [14]. Thus, the number of magnetic resonance imaging (MRI) devices increases to that effect and correlates with the number of imaging exams [14]. It is likely that this trend will continue in the next years due to the rising world population [15]. MRI exams are made to offer medical doctors insight into internal structures of the human body to detect abnormalities without going in vivo. In the early 90s physicians and medical personal started using imaging devices broadly. This could be associated as a reason for the decrease of cancer death rates over the last 20 years [16]. Detecting abnormalities in images is achieved by visual observation of medical doctors. This plays a key role for the image interpretation and is therefore a vital part of the diagnosis process [17].

As detecting or recognizing abnormalities in images is a high-level image processing step, several former steps are necessary, e.g., segmentation [18]. The simplest way to obtain a segmented image is a manual segmentation [6]. This time consuming manual task is practically not feasible because of the high number of imaging exams [14].

Based on this and the fact that detecting abnormalities is important for diagnostic processes, researchers are seeking automated algorithms for medical image segmentation, which is performed by software programs [6], [13], [19], [20]. Solving this problem is the

key to success in the automated detection of abnormalities [6].
Being aware and knowing the segmentation of a region of interest in an image leads to the possibility of calculating properties out of the segmentation. If these properties differ from their normal values (e.g. high eccentricity), abnormalities can be detected [18]. Therefore, an accurate initial segmentation is necessary to achieve accurate results in further high level image analysis [6]. The following Figure 1.1 represents the computer vision pipeline and provides the basic processing steps to understand images [18]. Each step of Figure 1.1 forms what is called a domain:

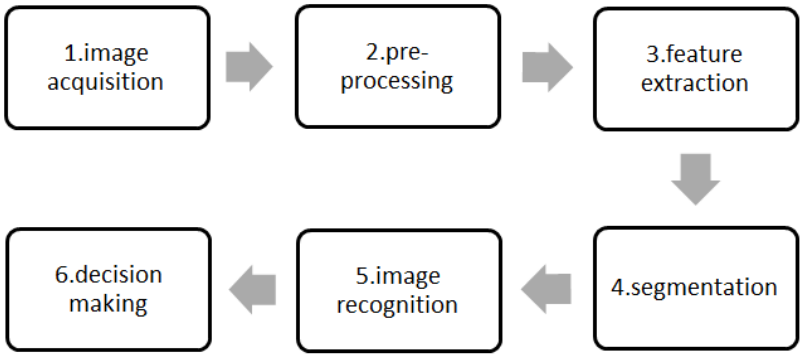


Figure 1.1: Computer vision pipeline

With the exception of step 1, the remaining steps are usually performed intrinsically by medical doctors through visual observation [17]. In order to develop an automated solution for which no expert interaction is required, this work will focus on the domains 2, 3 and 4 with an in-depth look into step 4 (Segmentation). The domains and tasks of points 5 and 6 are not covered, because they require accurate results in former steps. Hence, this work focuses on accurate and reliable image segmentation in order to establish a good initial image segmentation to form either a basis for high level image processing tasks (domain 5 and 6) or to support medical doctors in recognizing abnormalities.

Segmentation of structures or tissues in medical images can affect and improve several medical routines (e.g. diagnosis, evaluation, treatment planing/selection) positively if the segmentation is accurate and reliable. Therefore, the main requirement for an automated segmentation algorithm is that it needs to segment the organ of interest like an expert in reasonable time. Only in this case, the algorithm is eligible for clinical usage [13], [21].

An automated segmentation algorithm, which is able to segment all kind of organs in multi-modality images, currently does not exist for clinical usage due to the involving complexity. Thus, current solutions concentrate on automated segmentation of one specific organ/tissue of interest [22], [23]. Because of MD Anderson's participation at an automated prostate zone segmentation challenge[1] and the fact that prostate cancer is the most frequently diagnosed cancer after skin cancer amongst men in the United States, this approach focuses on the automated segmentation of prostate gland and consequently, the prostate zones [16].

[1]Results of the NCI-ISBI 2013 Prostate Challenge, (accessed October 2013), http://challenge.kitware.com/midas/community/

Due to the technical developments in MRI over the past ten years, MRI is accepted as the best imaging modality to visualize anatomical prostate structures [21]. A reason for this is that the MR technique enables the visualization of soft tissue and prostate tissue consists of soft tissue [24]. Therefore, the proposed automated segmentation algorithm was developed and tested on in vivo T2-weighted MRI studies.
This work is important for a number of reasons: automated prostate zone segmentation provides information about the size, shape, position and volume of the prostate zones. The prostate volume correlates with the presence of prostate cancer and is therefore important for diagnostic issues as a predictor for prostate cancer [25], [26]. The provided information (size, shape, etc.) increases the knowledge about the prostate, which could affect and improve multiple clinical routines as for instance, prostate cancer staging, prostate evaluation as well as treatment selection and planing. Furthermore, it could also influence interventional techniques as MRI-guided biopsies. As a consequence, it can reduce diagnostic uncertainties [21].

This research work concentrates on the automated segmentation of prostate zones by utilizing artificial neural networks in a supervised manner. There are multiple studies about artificial neural networks in general, but no corresponding literature can be found for the proposed prostate zone segmentation utilizing neural networks. Hence, the next section presents similar work.

1.1 Published Research in Prostate Segmentation

A considerable amount of literature has been published on prostate segmentation and on prostate zone segmentation. Existing approaches differ from each other by the required user interaction (automated, semi-automated and interactive) and the utilized methods. This section presents related research work and points out the accuracy of each method by citing the mean Dice coefficient (DC), which can be seen as the mutual overlap of the segmentation and ground truth (see Section 4.7 for Dice coefficient).

The research team around G. Vincent et al have utilized an active appearance/shape model to segment the entire prostate in T2 weighted magnetic resonance (MR) images automatically. They reported a mean DC score of 0.88 [27]. Toth R. et al [28] presented a medial axis based statistical shape model for prostate segmentation and reported a mean DC of 0.93. In a further research work Toth R. et al [29] combined the existing semi-automated active shape model segmentation with subsequent volume estimation of the prostate and reported a DC score of 0.84 and a Pearson correlation coefficient of $r^2 = 0.82$ for the corresponding volume estimation. Furthermore, S. Ghose et al [30] presented a texture enhanced active appearance model in which Haar wavelet approximation coefficients have been utilized to extract texture features. The texture features have improved the segmentation results of the entire prostate gland. The proposed method was tested on MR images and transrectal ultrasonography (TRUS) images. They achieved mean DC scores of 0.95 on MRI studies and 0.81 on TRUS studies.

Further approaches are atlas based segmentations. A prominent example of these was reported by Geert Litjens et al [31]. They proposed a semi-automatic multi-atlas algorithm and achieved a mean DC of 0.78. M. Rusu *et al.* [32] presented a semi-automatic statistical atlas construction via anatomically constrained registration for the prostate and utilized the atlas for segmenting the prostate zones. To generate the atlas, MR image information was combined with corresponding histological images. Respectively, the achieved mean DC scores are 0.89 and 0.77 for the two segmented prostate zones.

M. Yang et al [33] reported an automated prostate segmentation approach using discriminant boundary features. Due to the variable shape of the prostates, their approach included scale invariant features transformation. The DC score between the automated segmentation and the ground truth is 0.94 for the entire prostate gland.

The described individual scores of each research team cannot be compared and evaluated adequately because of three reasons. Firstly, they utilized different interaction modes; secondly, results are based on different datasets; and thirdly, some segmented the prostate gland and some segmented the prostate zones. To counteract this, the *Prostate MR Image Segmentation Challange 2013*[2] and the *NCI-ISBI Automated Prostate Segmentation of Prostate Structure Challange 2013*[3] was realized and presents a comparison of different approaches by utilizing the same dataset, but different interaction modes. Not all previously described research teams have participated at these challenges.

1.2 Structure of the Work

Chapter 2 provides information about the prostate seen from a medical viewpoint. The anatomy, function as well as prostate imaging and diagnosing prostate cancer are described. Chapter 3 contains fundamental image processing basics to enable the reader to better understand the upcoming sections and the proposed methods.
Chapter 4 describes methods in order to achieve automated prostate structure segmentation. The utilized steps are presented in detail. Chapter 5 shows results and an evaluation of the proposed algorithm. The thesis ends with a discussion in Chapter 6. In the discussion chapter are encountered problems, limitations as well as an outlook given.

[2]MICCAI Grand Challange of Prostate MR Image Segmentation 2012, (accessed October 2013), http://promise12.grand-challenge.org/

[3]NCI-ISBI 2013 Challenge - Automated Segmentation of Prostate Structures, (accessed October 2013), http://goo.gl/OBdxpq

Chapter 2

Medical Background of the Prostate

This chapter provides insight into the medical background of the prostate. Subsection 2.1 describes the anatomy and the function of the prostate. Afterwards, Subsection 2.2 explains the common medical imaging techniques to visualize the prostate. The last Subsection 2.3 presents information on how prostate cancer is currently diagnosed.

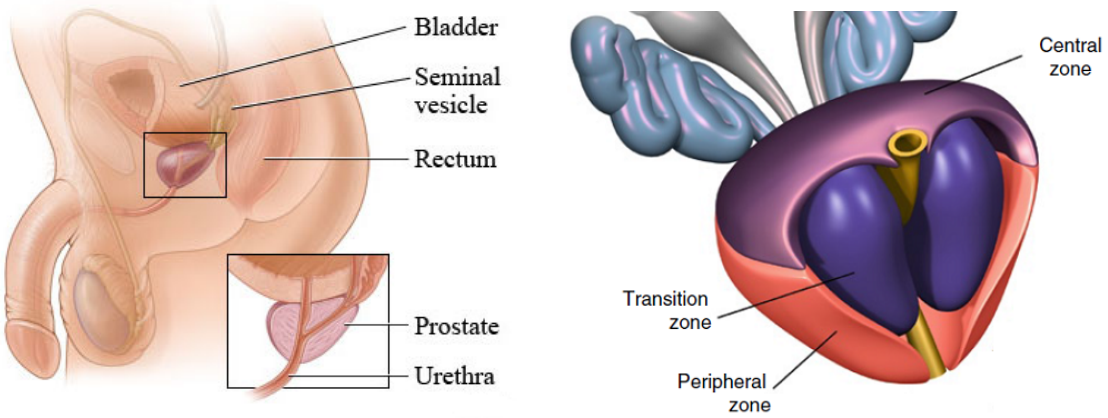
2.1 About the Prostate

In 2013, the American Cancer Society estimates that there will be 238,590 new cases of prostate cancer in the United States. Approximately 29,720 people will die of prostate cancer in 2013. Moreover, prostate cancer is the most frequently diagnosed cancer in men after skin cancer [16]. For unknown reasons, prostate cancer occurs 70% more often in white people than in African Americans [16]. Similar to other types of cancer, prostate cancer is most effectively treated when diagnosed early. Hence, special attention should be directed to prostate cancer and its diagnosis [21].

Prostates can only be found in males and are part of the reproductive system. Each male has normally one prostate located in the men's abdomen. To be more specific, the position of the prostate is below the urine bladder and in front of the rectum, as shown in Figure 2.1(a). Figure 2.1(b) represents the different zones of the prostate, which will be explained later in this section.

As the prostate is located near the rectum, it can be palpated from the rectum. This procedure plays a role in diagnosing prostate cancer, which is explained in section 2.3. The size of the prostate can show high variation during lifetime. In early ages the prostate has usually the size of a walnut or chestnut. In men older than 50, the prostate can be much larger. If the prostate increases its volume caused by illness, urinating and ejaculation problems can occur. The prostate's average weight is 30-40g. A prostate having a volume $> 40cm^3$ is considered large. As mentioned in the introduction, the prostate volume is a predictor for prostate cancer [25], [26].

The prostate is an exocrine gland and is responsible for secreting a fluid during ejaculation to extend the lifetime of sperm. This function is important for the fertilization process, because the female's vagina is acidic to protect herself for infection. Therefore, the secreted fluid enables sperms longer lifetimes in the vagina by protecting them from the acidic



(a) (b)

Figure 2.1: Anatomy of the prostate: (a) abdominal position of the prostate [4], (b) different zones of the prostate [34]

environment. Furthermore, the prostate's epithelial cells secrete a prostate specific antigen (PSA), which is an important marker for prostate cancer diagnosis (see Section 2.3). The secreted fluid and the sperm form together the semen, which is carried out of the body through the urethra. The urethra is a tube, which ranges from the bladder to the end of the genitals. The urethra goes through the center of the prostate and merges with the ejaculatory ducts [34], [35], [36].

As shown in Figure 2.1(b) the prostate consists of three principle zones. In a healthy prostate the central zone forms 25% of the prostate volume. The central zone and transitional zone form together the inner part of the prostate, which is known as central gland (CG). The peripheral zone (PZ) is the third zone of the prostate and lies partly around the CG. The combined area of CG and PZ is known as prostate gland. Over two-thirds of prostate carcinomas occur in the PZ. Thus, this work targets the segmentation of CG and PZ. The yellow tube in Figure 2.1(b) represents the urethra [34].

Patients with early or mid-stage prostate cancer do not show any staginess symptoms. Possible symptoms indicating prostate cancer are blood in the urine/semen and ejaculation/urinating problems. Unfortunately, there are no typical prostate cancer symptoms because the described symptoms can also be caused by other diseases. The only well-known risk factor for prostate cancer is an increasing age. 97% prostate cancer cases are diagnosed in men older than 50 years [16], [35].

[4] My Health Alberta, ©Healthwise, incorporated, (accessed October 2013), http://goo.gl/1cAch0

2.2 Prostate Imaging

Currently, established modalities to visualize the prostate are ultrasonography (US), MRI, magnetic resonance spectroscopy (MRS), computed tomography (CT) and positron emission tomography (PET). Due to rapid technical developments in imaging devices, prostate cancer can be visualized better than ever before. There also exist several new techniques such as dynamic contrast-enhanced MRI and diffuse-weighted imaging, which should be evaluated over the next years [37], [38], [39].

1.Ultrasonography: The most common technique to visualize the prostate is TRUS, which is simple and readily available. Echoes caused by emitted sound waves generate the gray-scale image information. This technique can be used either for viewing the prostate itself or for guided prostate biopsies. TRUS is commonly used to estimate the prostate's volume. [37]

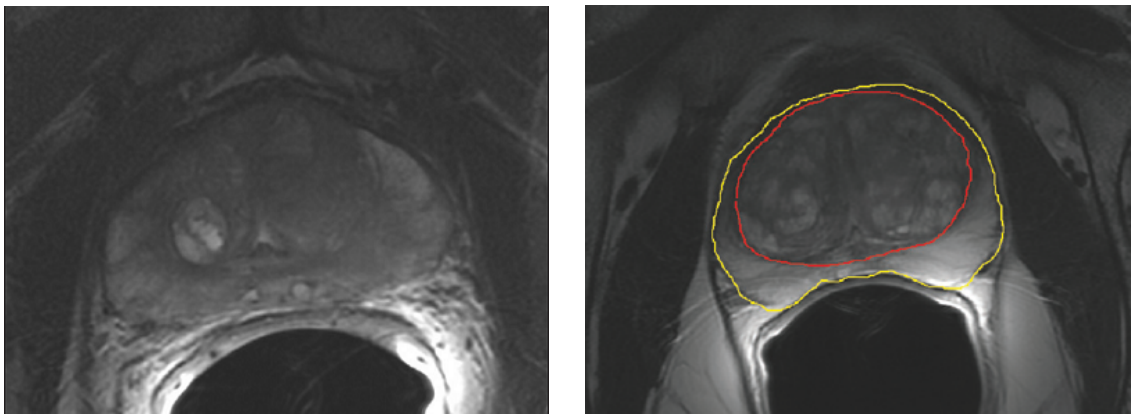
2.Magnetic Resonance Imaging: MR images provide a clear look at the prostate [35]. Especially, T2 weighted MR images provide sufficient soft tissue resolution to visualize the prostate structures. In contrast to T2 weighted images, appears the prostate in T1 weighted MR images as one homogenous region [38]. MR imaging is used for prostate cancer localization and staging. Referring to the C. Tempany and F.Franco [21], MRI will become more important for detecting and diagnosing prostate cancer [40]. Beyond that, MR imaging has also been proposed successfully for MRI-guided prostate biopsy and has shown promising results [41], [42].

3.Magnetic Resonance Spectroscopy: Tumor growth involves increasing cell metabolism, which can be analyzed by MRS. For example, MR images provide information about the location, size and position of the tumor and in contrast MRS provides information about the cellular activity within tissues. MR is usually performed before MRS to superimpose the T2 weighted MR with the MRS image. Several studies have discovered that combining MR with MRS, which means combining anatomical and metabolic information, leads to higher accuracy in diagnosing prostate cancer. [37], [38]

4.Computed Tomography: CT imaging plays a less important role in detecting and staging prostate cancer due to poorly visualized intraprostatic anatomy. CT scans can help to diagnose metastatic diseases (bone metastases and lymph node involvement), but MRI should be used as it is superior to CT imaging. [37], [43]

5.Positron Emission Tomography: PET is a functional imaging technique and visualizes glucose metabolism of cells. Due to the fact that cancer has increased metabolism rates, it can be detected and staged by PET scans. But research results have shown mediocre results, because of the difficulty to discriminate between benign and malign tissue regions. Both regions present high metabolism rates. Researchers try to figure out the best PET tracer for the optimal PET use for prostate cancer. [37], [44]

In conclusion MRI is accepted as best imaging modality to visualize the intraprostatic anatomy [21]. Two examples of MR images of the prostate are illustrated in Figure 2.2.



(a) (b)

Figure 2.2: T2-weighted prostate MR images seen from a axial view [21]: (a) despite high soft tissue resolution it is still hard to distinguish between prostate zones, (b) expert contours of prostate gland shown in yellow and CG is represented in red. The area between these two boundaries represents PZ.

2.3 Diagnosis of Prostate Cancer

In patient cases with indications of prostate cancer, digital rectal exam (DRE) and PSA blood tests are performed as a first step to detect prostate cancer [37]. During a DRE, the examining doctor puts a gloved finger into the patient's rectum to palpate prostate abnormalities. A PSA test is understood to measure an enzyme produced by the prostate, which allows drawing conclusions about abnormalities (see Section 2.1) [45]. Due to the fact that solid pathological diagnoses can neither be made by DRE nor by PSA, it is necessary to carry out biopsies, if either DRE or PSA shows abnormalities. A TRUS-guided biospy is typically performed as a second step followed by pathological examinations [21], [35]. To receive accurate results from pathological examinations, it is important to extract tissue samples from the affected region(s). If tissue is extracted from a non-affected, healthy region despite the fact the observed patient has actual prostate cancer, false-negative results are the outcome. Thus, prostate biopsies play a key-role in the diagnostic process of prostate cancer. Being aware of the internal anatomical prostate zones brings advantages in performing diagnostic prostate biopsies. This issue is addressed by segmenting the internal anatomical prostate zones. Recent results have shown that MRI-guided biopsies have a higher detection rate of prostate cancer in patient with suspected prostate cancer and previously negative TRUS-guided biopsies [21], [41], [42]. Automated prostate zone segmentation in MR images, provides information about the prostate, which could affect and improve multiple fields as evaluating prostate cancer, treatment selection and reducing diagnose uncertainties. Furthermore, it can improve the accuracy of MRI-guided biopsies or other interventional techniques.

Chapter 3

Image Processing Background

This chapter describes image processing basics, which are used in this thesis in order to help the reader to understand utilized methods in Chapter 4. Section 3.1 concentrates on object segmentation in images, which is part of the computer vision pipeline (see Figure 1.1). The following two sections address supervised learning and classification of image pixels. Thus, Section 3.2 contains information about the proposed learning method to train a classifier for segmentation. The last Section 3.3 explains the classification process and the theory behind the implemented classifier in detail.

Most of the following information in this chapter is extracted from [46] and [47]. If further information is used, corresponding sources are given.

3.1 Object Segmentation in Images

Dividing or partitioning a digital image $\boldsymbol{I}$ into disjoint regions with similar properties (e.g. intensity or color) is understood as image segmentation [6], [48]. An image $\boldsymbol{I}$ is segmented in K disjoint regions r_k through:

$$\bigcup_{k=1}^{K} r_k = \boldsymbol{I},\ for\ \forall k:\ r_k \subseteq \boldsymbol{I}\ \ and\ \ \forall k \neq j:\ r_k\ \cap\ r_j = \emptyset \tag{3.1}$$

In literature, image segmentation is described as a low-level processing step within the computer vision pipeline (see Figure 1.1). But in this work an advanced version of image segmentation namely object segmentation is utilized. It is defined as the identification of different objects in an image. For example, consider cell segmentation in a microscopic image. The goal of object segmentation is to segment each cell in the image from the background and distinguish it from other cells. Transferred to this approach, the goal of object segmentation is to segment each zone of the prostate from the background and distinguish them from other zones.

In the following work i and j are discrete values in the range of the image dimensions N and M and represent pixel position indices (see Equation 3.2).

$$S = \{(i,j)|1 \leq i \leq N, 1 \leq j \leq M\} \tag{3.2}$$

The output of object segmentation is a label image $\boldsymbol{L}(i,j)$, whereby each pixel position contains a discrete value k in the range from 0 to K that unambiguously belongs to one of the K-objects in the image (see Equation 3.3):

$$\boldsymbol{L}(i,j) = k, \quad for \quad \boldsymbol{L}(i,j) \in \{0, K\} \quad and \quad \forall(i,j) \in \ S; \tag{3.3}$$

$\boldsymbol{L}(i,j) = 0$ usually determines background and $\boldsymbol{L}(i,j) = k$ *for* $k = 1,\ldots,K$ represents objects of interest (in this work different prostate zones) [49]. Consequently, $\boldsymbol{I}$ and $\boldsymbol{L}$ have the same image dimensions. The simplest label image is represented by a binary label image, containing 0 for background and 1 for foreground. In literature, several methods exist to segment an image as for instance threshold, histogram, watershed, level sets and clustering methods [6], [18], [48], [49]. Contrary to these methods this work utilizes a different approach to achieve object segmentation, which is particularly pixel classification whereby, the input image is a MR image $\boldsymbol{I}(i,j)$ and the predicted output by the algorithm $\boldsymbol{L}(i,j)$ represents the corresponding label image. The output label can contain up to three discrete values, which are as follows: 1 represents PZ, 2 indicates central gland CG and 0 stands for background. Thus, object segmentation or in other words the identification of different objects (in this case different prostate zones) is achieved through pixel classification. Considered in more detail, object segmentation is accomplished in this work by a supervised classification approach, which is described in detail in the next two sections.

3.2 Supervised Learning

Generally, there are three types of learning for pixel labeling in the fields of image processing, pattern recognition and machine learning. The first one is called unsupervised learning in which the learning algorithm learns mapping input to the correct output on its own. Unsupervised learning can be seen as detecting structures or clusters in data. The second type of learning is called reinforcement learning, which deals with learning "what to do". An algorithm should learn how to map situations to actions in a way to maximize a numerical reward [50]. For instance, consider pet obedience schools where dogs learn to sit. Every time a dog sits on command, it will receive rewards. The third type of learning is known in literature as supervised learning. Supervised learning is understood to find a function which maps input to the correct output by telling the algorithm during the training the correct output [6], [46].
Supervised learning methods can be divided into training and prediction phase. Hence, these methods require training with expert selected labels, which are determined in the following expert labels. The fact of having expert drawn labels available, it is hypothesized that automated prostate zone segmentation would be amenable to a supervised

machine learning approach. This means that a classifier is trained based on expert labels in order to use it subsequently to process new images. At this point, definitions are invented to explain how a classifier-model is trained and then in further consequence utilized for prediction.

A set $\mathcal{D}$ is comprised of input vectors $\mathbf{x}_w$, as well as of output vectors $\mathbf{y}_w$ and is defined as follows:

$$\mathcal{D} = \{(\mathbf{x}_w, \mathbf{y}_w)\}_{w=1}^{u} \;\; with \;\; \mathbf{x}_w = (x_1, x_2, ..., x_{v-1}, x_v)^T \;\; and \; \mathbf{y}_w = k \;\; for \;\; k \in \{1, ..., K\} \tag{3.4}$$

Whereby u in equation 3.4 represents the total number of in-output pairs. One in-output pair is denoted by $(\mathbf{x}_w, \mathbf{y}_w)$. The input vector $\mathbf{x}_w$ is a v-dimensional vector. Where each dimension represents one feature. Features are extracted from images in order to generate $\mathbf{x}_w$. This procedure is described later in the feature extraction Section 4.3.
In most models the output $\mathbf{y}$ is a v-dimensional vector containing values k in the range $\mathbf{y} \in \{1, ..., K\}$. Furthermore, $\mathbf{y}$ is in literature also called target vector. If $\mathbf{y}_w$ contains discrete values, the problem is categorical and known as a classification problem (see Section 3.3). If $\mathbf{y}$ contains real-valued or continuous values, the problem is determined to be a regression problem.

As described in Equation 3.4 one in-output pair consists of one input vector $\mathbf{x}_w$ containing v-features and one target value $\mathbf{y}_w$, which represents the corresponding class. $\mathbf{y}_w$ is denoted in vector notation because some models require the target vector in form of the $1 - of - K$ coding scheme. For instance consider six classes ($K = 6$), the target vector for class 2 is either determined as $\mathbf{y}_w = (2)$ (single target vector) or in the $1 - of - K$ coding scheme as $\mathbf{y}_w = (0, 1, 0, 0, 0, 0)^T$ (multi target vector).
For more than one in-output pair – which is usually the case – the input for training a classifier is a $u \times v$ dimensional matrix denoted as $\mathbf{X}$. Matrix $\mathbf{X}$ is also called the design matrix, because it is used to design the model. In this case the model is a classifier. Depending on utilizing single or multi target vectors, the model's output $\mathbf{Y}$ is either a u-dimensional vector or a $u \times k$ dimensional matrix. $\mathbf{X}$ and $\mathbf{Y}$ are illustrated in Equation 3.5.

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_u^T \end{pmatrix} = \left.\begin{pmatrix} \overbrace{\begin{matrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1v} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2v} \\ \vdots & & & \ddots & \\ x_{u1} & x_{u2} & x_{u3} & \cdots & x_{uv} \end{matrix}}^{v \text{ features}} \end{pmatrix}\right\} u\text{-items} \quad and \quad \mathbf{Y} = \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_u \end{pmatrix} \tag{3.5}$$

Note that $\mathbf{X}$ and $\mathbf{Y}$ must have the same u-dimension for the training phase. Combining $\mathbf{X}$ and $\mathbf{Y}$ results in the training set $\mathcal{D}$, which is in this work used to train a classifier:

$$\mathcal{D} = \{(\mathbf{x}_w, \mathbf{y}_w)\}_{w=1}^{u} \quad \widehat{=} \quad \mathcal{D} = (\mathbf{X}, \mathbf{Y}) \tag{3.6}$$

The goal of the training phase is to learn a model h that maps the input to the correct output or respectively directly into the decisions (see Section 3.3). The precise form of model h is learned during the training phase (see Equation 3.7). Applying this methodology to the utilized approach means that the basic idea of this work can be described as follows: the goal is to train a model based on MR images and expert labels and then predict the prostate zones on new MR images. As it utilizes a supervised classifier, it must be trained in order to process unlabeled data. Therefore, Equation 3.7 describes the process of both phases:

$$\textit{training phase}: \quad \mathcal{D} = (\mathbf{X}, \mathbf{Y}) \longrightarrow h \qquad \textit{prediction phase}: \quad \hat{\mathbf{Y}} = h(\mathbf{X}) \tag{3.7}$$

The form of model h is assessed in the training phase based on $\mathcal{D}$. Afterwards, model h can be applied to novel unlabeled input for which no ground truth is available. This procedure is called the prediction phase. The ability to predict output from novel input – input seen for the first time – is enabled through generalization. Generalization is involved through the fitted model and is a central goal of pattern recognition. Making predictions on known data (training set) is easy because the answer just needs to be extracted from the training set (look-up table).
Prediction with novel input is made with $\hat{\mathbf{Y}} = h(\mathbf{X})$, whereby the hat-symbol indicates estimates because the goal is to predict or estimate output which is similar to the ground truth. Finally, a different look at supervised learning is for instance the following: A training set $\mathcal{D}$ is used to parameterize a model h for mapping $\mathbf{x}_w$ to $\mathbf{y}_w$ [6]. The more input vectors $\mathbf{x}_w$ that are assigned to the correct output $\mathbf{y}_w$ by the model, the more accurate segmentation results can be achieved. The next sections concentrate on assessing model h.

3.3 Classification of Image Pixels

Classification is part of the decision theory and is about grouping objects together to classes. Objects with similar properties or values should be assigned to the same class. In the field of pattern recognition, this means that each input is assigned to a discrete value or class. By dealing with discrete output values, this approach sets itself apart from regression, in which the output consists continuous variables. Furthermore, combining probability theory with decision theory enables optimal decision making and is therefore a essential part of classification. Classification problems can be broken down into two stages. The first stage is called inference stage and the second one is called decision stage. In general, there are two different models to solve classification problems, which

are namely generative models and discriminant models. For further understanding, it is important to know that decision problems can be treated as classification problems. The following section explains generative as well as discriminant models and provides a transition to artificial neural networks:

1. Generative models: These models try to solve the inference stage by obtaining the probability distributions $p(\mathbf{x}|\mathbf{y}, \mathcal{D})$, $p(\mathbf{y}, \mathcal{D})$ and $p(\mathbf{x}, \mathcal{D})$. Then the posterior class probabilities $p(\mathbf{y}|\mathbf{x}, \mathcal{D}, \theta)$ using Bayes' theorem (see Equation 3.8) are determined, whereby θ are model parameter learned in the training phase. Given a new input feature vector $\mathbf{x}$ and a training set $\mathcal{D}$, the conditional probability distribution over all classes is denoted as $p(\mathbf{y}|\mathbf{x}, \mathcal{D})$. This notation points out, that the probability for class $\mathbf{y}$ is conditional on the new input feature vector $\mathbf{x}$ and on the training set $\mathcal{D}$, by writing them on the right side of the conditional bar $|$. $p(\mathbf{y}, \mathcal{D})$ presents the class prior probability and $p(\mathbf{x}, \mathcal{D})$ represents the likelihood of the data and is in literature also known as evidence. In order to form Equation 3.8, generative models need to capture the joint probability $p(\mathbf{x}, \mathbf{y})$ in order to derive $p(\mathbf{x}|\mathbf{y}, \mathcal{D})$, $p(\mathbf{y}, \mathcal{D})$ and $p(\mathbf{x}, \mathcal{D})$ by marginalizing or summing out. This procedure can be time consuming for large datasets.

$$p(\mathbf{y}|\mathbf{x}, \mathcal{D}, \theta) = \frac{p(\mathbf{x}|\mathbf{y}, \mathcal{D})\ p(\mathbf{y}, \mathcal{D})}{p(\mathbf{x}, \mathcal{D})} \quad with \quad p(\mathbf{x}, \mathcal{D}) = \sum_{K}\ p(\mathbf{y}|\mathbf{x}, \mathcal{D})\ p(\mathbf{y}, \mathcal{D})) \tag{3.8}$$

The denominator in Bayes' theorem can be seen as normalization to ensure the posterior class probabilities sum to one. When the posterior probabilities are determined, the decision stage can be solved by assigning each input to the maximum a posteriori probability utilizing Equation 3.9. This procedure corresponds to assigning input to the most probable class or to the "best guess". Equation 3.9 is also known as maximum a posteriori estimation (MAP).

$$\hat{\mathbf{y}} \ = \ \hat{h}(\mathbf{x}) = \arg\max_{k=1}^{K} p(y = k|\mathbf{x}, \mathcal{D}) \tag{3.9}$$

These models are denoted as generative models because they are full probabilistic models and form input as well as output distributions from any variable. Hence, sampling from a model is possible, which enables consequently generation of synthetic input data. Generative models are usually trained by maximizing the joint likelihood $\sum_{u=1}^{w} \log p(y_u, \mathbf{x}_u|\boldsymbol{\theta})$, whereby $\boldsymbol{\theta}$ represents the model parameters.

2. Discriminative models: If classification is the task of interest, then it is sometimes (depending on the input) time wasting to compute the joint probabilities, when just the posterior probabilities are needed. Therefore, discriminative models form in contrast to generative models, class posteriori probabilities directly in order to solve the inference stage. In the next stage an optimal class assignment using the learned posterior probabilities is performed to realize classification (see Equation 3.9). A special approach to accomplish classification is to find a function in the following called

discriminant function, which maps an input vector $\mathbf{x}$ directly to an output class $\mathbf{y}$. This approach solves the inference stage by learning a function. Thus h in Equation 3.7 can be seen from now on as discriminative function: $h : \mathbf{x} \longrightarrow \mathbf{y}$. Discriminative models are usually trained by maximizing the conditional log likelihood $\sum_{u=1}^{w} \log p(y_u|\mathbf{x}_u, \boldsymbol{\theta})$

In addition, there is a further distinction between available pattern recognition models for classification and pattern recognition models for regression. These models can be subdivided into two groups called parametric models and non-parametric models. The former models have a fixed number of parameters and can be used as a consequence fast. The model's drawback is, that it strongly relies on the input data distribution, which results in stiffness if novel data needs to be processed. In contrast there are non-parametric models, whereby the number of parameters increase with the amount of training data. Non-parametric models are more flexible, but computational expensive for large datasets. Table 3.1 represents an overview of existing pattern recognition models.

Table 3.1: List of classification and regression models

Model	Classif/Regre	Gen/Discr	Param/NonParam
Naive Bayes Classifier	Classif	Gen	Param
K-nearest Neighbor Classifier	Classif	Gen	NonParam
Classification and Regression Trees	both	Discr	NonParam
Support Vector Machine	both	Discr	NonParam
Linear Regression	Regre	Discr	Param
Logistic Regression	Classif	Discr	Param
Neural Network	both	Discr	Param
Deep Belief Network	both	Gen	source

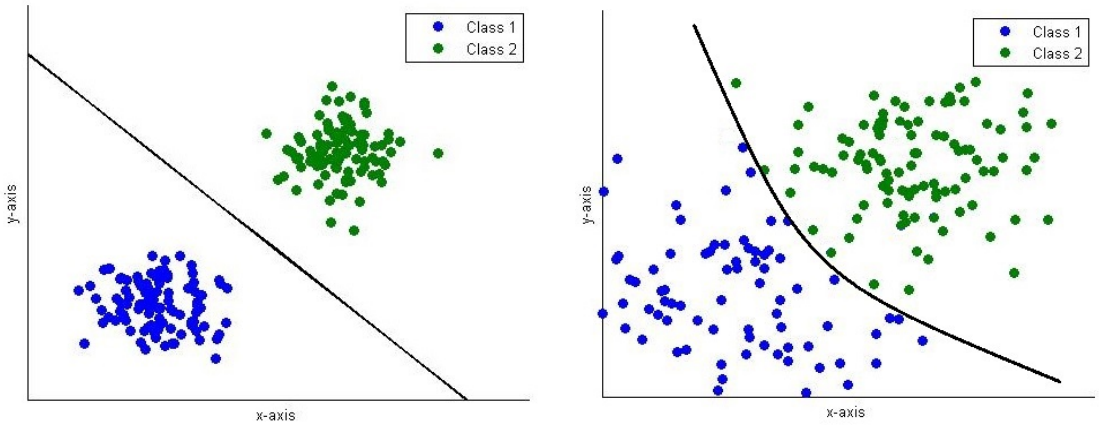
List of different classification and regression models. The columns are defined as follows: first column contains the model name; second column presents if the model can be used for classification, regression or both; third column determines whether the model is a generative or discriminative model; fourth column displays if the model is either a parametric or non-parametric model

This work utilizes primary neural networks for classification, which are discriminative models as illustrated in Table 3.1. For this reason, the rest of the chapter is dedicated to discriminative models and functions. However, this work also contains a deep belief network approach to initialize a neural network, which is described afterwards in Section 3.3.2.

As the goal of classification is assigning each input vector (feature vector) $\mathbf{x}$ to a discrete class $k = 1, ...K$ (target), the input space has to be divided into decision regions by discriminant functions when using discriminative models.

Unsupervised classification is related to density estimation and tries to cluster data itself into classes during the training phase. Complete unsupervised classification does

not provide satisfying segmentation results in this work. Opposite to this is supervised classification, in which the models are not seeking to model distributions of the input variables. Supervised classification is commonly used in the field of computer vision, because current supervised classification models produce in general more precise results compared to results obtained from unsupervised classification. Because of having valuable expert labels available, supervised classification is utilized.
The simplest classification problem is represented by one-dimensional input, which is comprised of two classes. A further assumption is that the input is linear separable. Based on this, the input space can be separated by a simple threshold. Linear and non-linear separable two-dimensional input data comprised of two classes is illustrated in Figure 3.1. Two dimensional input data correspond to a feature set consisting of two features (x_1, x_2). Therefore, the x-axis in Figure 3.1 can be seen as the first feature (x_1) and accordingly the y-axis can be seen as the second feature (x_2). In conclusion, a v-dimensional input corresponds to a v-dimensional feature vector.



(a) (b)

Figure 3.1: 2D Classification between two classes: (a) linear discriminative function, (b) non-linear discriminative function

To explain how classification can be achieved, as exemplarily shown in Figure 3.1, linear regression models are introduced and illustrated in Equation 3.10. A linear regression model is a weighted linear combination of inputs and fits a linear discriminant function to the input data.

$$y(\mathbf{x}, \theta) = h_\theta(\mathbf{x}) = \theta_0 + \sum_{w=1}^{u} \theta_w x_w \tag{3.10}$$

θ_0 represents a bias and u indicates the total number of feature vectors (input vectors). θ_w indicates the weight of feature x_w. Adding to the existing feature vector, a feature

x_0 whose value equals one ($x_0 = 1$) enables a more compact notation of linear regression models (see Equation 3.11):

$$y(\mathbf{x}, \theta) = h_\theta(\mathbf{x}) = \sum_{w=0}^{u} \theta_w x_w = \boldsymbol{\theta}^T \mathbf{x} \tag{3.11}$$

with the feature vector $\mathbf{x} = (x_0, x_1, ..., x_u)^T$, the parameter vector $\boldsymbol{\theta} = (\theta_0, \theta_1, ..., \theta_u)^T$ and $\boldsymbol{\theta}^T\mathbf{x}$, whereby $\boldsymbol{\theta}^T\mathbf{x}$ represent the inner scalar product of $\boldsymbol{\theta}^T$ and $\mathbf{x}$. Linear regression models can be used to solve regression problems, but they are fairly limited in what they can learn to do. To solve classification problems, logistic regression models should be applied. Logistic regression models are derived from linear regression models by adding an outer-function $\sigma(.)$ to 3.11. Mostly, the outer function is represented by a sigmoid function. In literature, the sigmoid function is also referred as logistic function. The sigmoid function is defined as follows:

$$\sigma(a) = \frac{1}{1 + e^{(-a)}} \tag{3.12}$$

Combing Equation 3.11 and 3.12 leads to the logistic regression model, which is illustrated in Equation 3.13. These models can fit a linear discriminant function for classification purposes (see Figure 3.1(a)). The sigmoid function in 3.13 maps the output of the linear regression model to a number between 0 and 1. Therefore, the output can be treated as posterior probability for the "positive" class, considering a single classification problem for which two classes exist ($k_1 = 0$ or $k_2 = 1$).

$$p(\mathbf{y}_{k=1}|\mathbf{x}, \mathcal{D}, \theta) = y(\mathbf{x}, \theta) = \frac{1}{1 + e^{(-\theta^T\mathbf{x})}} = \sigma(\boldsymbol{\theta}^T\mathbf{x}) \tag{3.13}$$

Accordingly, the probability for class two is given by $p(\mathbf{y}_{k=2}|\mathbf{x}, \mathcal{D}, \theta) = 1 - p(\mathbf{y}_{k=1}|\mathbf{x}, \mathcal{D}, \theta)$. Thus, the inference stage is solved using Equation 3.13. Afterwards the decision stage can be solved by MAP (see Equation 3.9). Obviously, the input space in Figure 3.1(a) is well separated and can be discriminated by a linear logistic regression function of the following form: $y(\mathbf{x}, \boldsymbol{\theta}) = \sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$. Thereby, an input vector $\mathbf{x}$ is assigned to the most likely class, which results in class 1 if $y(\mathbf{x}, \boldsymbol{\theta}) < 0$ and to class 2 if $y(\mathbf{x}, \boldsymbol{\theta}) \geq 0$. This notation automatically assigns an input vector to the most likely class, because the sigmoid function outputs 0.5 for an input with value zero $\sigma(0) = 0.5$. Dealing with more than two classes is in literature known as multi-class classification. As the label image $\mathbf{L}(i, j)$ can contain three different values, this thesis addresses a multi-class classification problem.

Despite the fact that logistic regression models include the word "regression", they just can be used for classification. Linear regression and logistic regression models fit a linear discriminant function or in other words they provide a linear decision boundary. If a linear discriminative function cannot separate the input space, then a non-linear discriminative function can be applied (see Figure 3.1(b)) to reduce the misclassification rate. Linear

regression models have linear discriminative functions (e.g. line for 2D feature vectors) and analogous have non-linear regression models non-linear discriminative functions (e.g. polynomial function for 2D feature vectors). Considering a three dimensional linear separable input space, the discriminative function is represented by a hyper-plain. Classifying the input space into more than 2 classes results in more than one discriminative function and leads to increased complexity.

Most of the real world data including the data utilized in this thesis are not linear separable. Hence, the goal is to fit a non-linear classification model. To solve non-linear classification problems, as for instance the example illustrated in Figure 3.1(b), an adaption of the existing logistic regression model is necessary. The input $\mathbf{x}$ in the logistic regression model (3.13) is replaced with $\boldsymbol{\phi}(\mathbf{x})$ to model non-linear relationships. $\boldsymbol{\phi}(\mathbf{x})$ is in literature known as *basis function expansion*. The form of logistic regression models for non-linear classification problems looks then as follows:

$$y(x, \theta) = \sigma\left(\sum_{w=0}^{u-1} \theta_w \phi_w(\mathbf{x})\right) = \sigma\left(\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x})\right) \tag{3.14}$$

with $\boldsymbol{\phi} = (\phi_0, \phi_1, ..., \phi_{u-1})^T$ and $\boldsymbol{\theta} = (\theta_0, \theta_1, ..., \theta_{u-1})^T$. $\mathbf{x}$ represents a feature vector. Thus, $\boldsymbol{\phi}(\mathbf{x})$ is a transformed feature vector by a linear or non-linear basis function. If the basis function is simply the "identity" of $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{x}$, then Equation 3.14 can be transferred in Equation 3.13. Utilizing a non-linear basis function $\boldsymbol{\phi}(.)$ leads to a non-linear model in total.

Many machine learning algorithms as for instance neural networks, support vector machines, classification and regression trees concern about estimating the basis functions $\boldsymbol{\phi}(.)$ from input data to form Equation 3.14. All previously mentioned algorithms are just different ways to estimate $\boldsymbol{\phi}(.)$. Therefore, estimating $\boldsymbol{\phi}(.)$ represents the core of this work. Depending on the input $\mathbf{x}$ it is sometimes appropriate to fit a polynomial function $\boldsymbol{\phi}(\mathbf{x}) = (1, x, x^2, ..., x^v)$ to the data to receive a non-linear logistic regression model. This would result, e.g., when using 50 features and fitting a quadratic function to the data, in a 1250 dimensional feature vector in total. Furthermore, fitting a cubic model would end up in a 20000 dimensional feature vector. Consequently, this approach can be computational expensive for high dimensional feature vectors. Neural Networks overcome this limitation and for that reason they are utilized. Neural Networks are derived from logistic regression models and subsequently Equation 3.14 is considered to be a starting point to explain Neural Networks in Section 3.3.1.

Classification can be applied in the field of image processing, on the one hand to the whole image (image classification) or on the other hand to each pixel in an image (pixel classification). The latter one is proposed in this thesis. Image classification and pixel classification are utilizing different in- and outputs sets. Thus, both are described briefly in the following to point out differences and to position this work in the field of pattern recognition.

***1.Image Classification*:** Image classification can also be seen as image recognition, because images are recognized by classification. Thereby, usually one decision is made

for the whole image, e.g., is a car in the image or not? Image recognition represents a high-level image processing step (see Figure 1.1) within the computer vision pipeline. In general, to achieve image recognition, it is firstly necessary to extract features in order to reduce the feature set's dimensionality, which represents the input for a classifier. If features are not extracted, the input for a classifier is given by the raw gray values from the image itself. Considering an 512×512 MR image, the feature set would consist of 262144 dimensions. Processing such a high dimensional feature set is not feasible. As previously described to overcome this, features needs to be extracted to reduce dimensionality.
An example for image classification is the recognition of hand written digits[5]. Hence, a classifier has to make a decision, which digit the image represents. Respectively, this results in 10 classes, whereby each class outlines one digit. Images in the Mixed National Institute of Standards and Technology database (MNIST) have a dimensionality of 28×28 pixel, which results in a total pixel number of 768. There exist several approaches to handle this "low" 768-dimensional feature set. This means that in this special case the whole image represents the classifier's input. Thus, no former feature extraction needs to be done because in this case features are basically given by the raw gray pixel values. This procedure is only possible because of the low dimensional feature set, which is in turn enabled by the small image size (28×28). In conclusion, using this approach for 512x512 MR images to segment the prostate zones is not feasible.
***2.Pixel Classification*:** Using a classifier for pixel classification in a N×M dimensional image, respectively N×M classifications have to be made in order to segment an image. Using the gray value of the pixels in an image as a feature, results in an one dimensional feature vector. Performing classification on just an one dimensional feature set leads to poor segmentation results in this thesis. Therefore, similar to image classification, features have to be extracted to enable achieving meaningful segmentations. In contrast, features are not extracted from the entire image, features are extracted from each pixel itself in order to cluster the input space.
An example for pixel classification is the work proposed by the research team around Farabet et al in [51]. They trained an algorithm in a supervised fashion to enable full scene labeling of new input images. Hence, multi scale features are extracted for each pixel from a Laplacian pyramid, whose input is an image. The output is a label image containing labels for each object. The basic methodology in [51] is similar to this work, but differs in the feature extraction and classification process.
To recognize objects and abnormalities in images, it is essential to have a good initial segmentation. In the next step, describing features can be calculated out of the segmentation (e.g. area, compactness, contrast) and on the basis of this, a classifier can recognize abnormalities through classification. Applying this procedure on the prostate could be as follows: Firstly, an initial segmentation of the prostate zones needs to be established in order to decide on behalf of this in a subsequent step if the prostate is cancerous or not. Summarized this work addresses establishing the initial prostate zone segmentations for which pixel classification is utilized. This work does not cover the recognition of abnormalities within the prostate.

[5]MNIST Database for Handwritten Digits, (accessed October 2013), http://yann.lecun.com/exdb/mnist/

3.3.1 Artifical Neural Networks

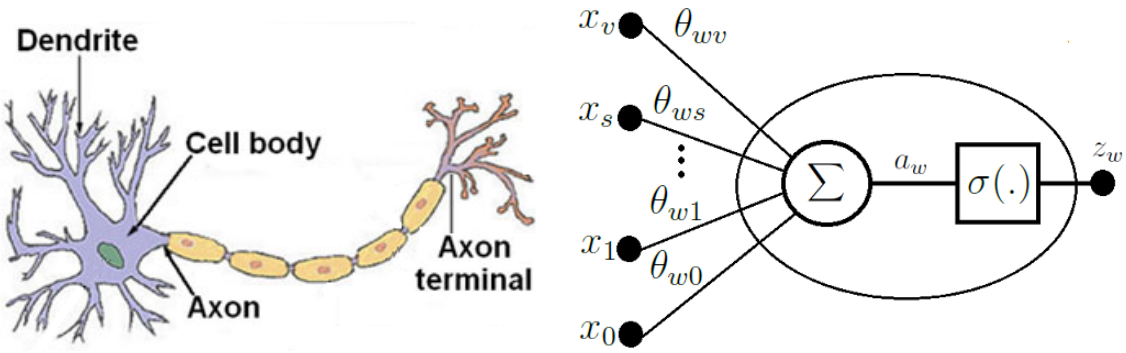
A way to implement a probabilistic classifier is to model the joint probability p($\mathbf{y}$,$\mathbf{x}$), condition on $\mathbf{x}$ and derive p($\mathbf{y}|\mathbf{x}$). This procedure is called generative approach. Another way to fit a model of form p($\mathbf{y}|\mathbf{x}$) is to model p($\mathbf{y}|\mathbf{x}$) directly. This approach is called the discriminative approach and is described in detail in this section.
A artificial neural network (ANN) is inspired by the human brain, which is comprised of about 100 billion neurons. Each neuron is connected to about 10,000 other neurons. Neurons are emitting signals via their axons. If the total signal input received from other neurons by its dendrites exceeds a certain threshold, the neuron itself is activated and emits a signal. This ability enables neuron interaction and creates thereby intelligent thoughts. An ANN is comprised of interconnected artificial neurons, which send activation signals to each other to fulfill a specific task as for instance classification. Other available applications for ANNs are function approximation, data processing, clustering and time series prediction [52], [53].

There are multiple types of neural networks as for instance Recurrent, Radial Basis ANNs and Self Organizing Maps. Probably the most common type of ANNs are Feed Forward Artificial Neural Network (FFANN), which are used for pixel classification in this work. FFANNs can vary in their topology as they can consist of multiple layers, which are in turn made up of multiple neurons. In literature FFANNs are also known as multi-layer perceptron (MLP).
FFANNs are series of logistic regression models stacked on top of each other. Logistic regression models are a generalized form of linear regression models and have already been described in Equation 3.14. Logistic regression models are considered as a starting point to explain FFANNs.

Equation 3.15 represents again logistic regression models for classification problems. Whereby, the linear combination of fixed non-linear basis functions $\phi_w(x)$ enables modeling non-linear discriminant models. If the outer function $\sigma(.)$ is removed, then Equation 3.15 can be used for regression. If the outer function $\sigma(.)$ is taken into account, then Equation 3.15 can be used for classification. From now on $\sigma(.)$ is denoted as activation function.

$$y(\mathbf{x}, \mathbf{w}) = \sigma\left(\sum_{w=1}^{u} \theta_w \phi_w(\mathbf{x})\right) \tag{3.15}$$

The goal of FFANNs is to extend the existing model in 3.15 in a way that the basis functions $\phi_w(\mathbf{x})$ depend on parameters, which can be adjusted by the coefficients θ_w during a training. In more detail, FFANNs are using non-linear basis functions, which are itself generated by a linear combination of the inputs. Whereby, the parameters in the linear combinations of the inputs are adaptive. Thus, the neural network model can be characterized as a series of functional transformations.



(a) (b)

Figure 3.2: Comparison of biological neuron and logistic artificial neuron; (a) structure of biological neuron, (b) structure of artificial neuron [6], whereby $x_0, ..., x_t$ corresponds to dendrites and z_w to the axon

To derive the neural network model several former steps are necessary. Firstly $w = 1, ..., u$ linear combinations of the input variables $x_s = x_1, ..., x_v$ have to be constructed as follows:

$$a_w = \sum_{s=1}^{v} \theta_{ws}^{(2)} x_s + \theta_{w0}^{(2)} \qquad with \;\; x_0 = 1 \;\; \Rightarrow \;\; a_w = \sum_{s=0}^{v} \theta_{ws}^{(2)} x_s \tag{3.16}$$

the superscript (2) indicates the parameters correspond to the second layer of the neural network model, which is also seen as the first hidden layer. Furthermore, the parameters $\theta_{ws}^{(2)}$ are in literature also referred as weights and the parameters $\theta_{w0}^{(2)}$ as biases. Defining an additional input variable x_0 whose values is 1 results in a more compact notation of Equation 3.16. The outputs of v-linear combinations are determined as activations a_w. In the second step, each linear combination a_w is transformed by a differentiable, non-linear activation function namely $\sigma(.)$ which results in:

$$z_w = \sigma(a_w) \tag{3.17}$$

The result of 3.17 corresponds to the non-linear basis functions $\phi_w(x)$ in 3.15. $z_1, ..., z_u$ are determined as hidden neurons, whereby z_w represents the w_{th} hidden neuron. In general and in this work, the sigmoidal function is used as activation function. Another possible activation function would be the 'tanh' function, which is of course also differentiable and non-linear. Possessing this property is important later on for the network training via error backpropagation.

As described earlier in this section, FFANNs are comprised of neurons, which are "activated" when the total input reaches a threshold. Equation 3.17 represents such artificial neurons. Hence, z_w can be seen as artificial neuron or logistic artificial neuron and is

[6] National Cancer Institute, (accessed October 2013), http://training.seer.cancer.gov/brain/tumors/anatomy/neurons.html

shown in Figure 3.2(b). Furthermore, Figure 3.2(b) illustrates how logistic artificial neurons work, whereby inputs $x_0, ..., x_v$ can be seen as "dendrites" of an artificial neuron and analogous the output z_w of artificial neurons can be seen as the neuron's axon. In contrast, Figure 3.2(a) shows a biological neuron.

FFANNs are usually comprised of three layers namely input, hidden and output layer. Sometimes in literature, the input is not referred as a layer, because it does not consist of adaptive weights. But in this thesis the input layer is referred as a layer. To generate the output layer, the values z_w have to be again linearly combined as already shown in 3.15 to form the following equation:

$$a_k^{(3)} = \sum_{w=1}^{u} \theta_{kw}^{(2)} z_w \quad + \quad \theta_{k0}^{(2)} \quad with \;\; z_0 = 1 \;\; \Rightarrow \;\; a_k^{(3)} = \sum_{w=0}^{u} \theta_{kw}^{(2)} z_w \tag{3.18}$$

whereby $k = 1, ..., K$ represents the number of outputs. The parameter $a_k^{(3)}$ characterizes the output-unit activation from $w = 1, ..., v$. Equation 3.18 corresponds to the third layer of the FFANN model as indicated by the superscript (3). Again $\theta_{k0}^{(2)}$ are bias parameters of the second layer. Defining $z_0 = 1$ enables writing the bias parameters into the sum.

The first Equation (3.15) in this chapter presents non-linear models for classification, considering an outer-function $\sigma(.)$. This function is again an activation function and is now applied to Equation 3.18. For classification purposes this function is again the sigmoid function so that the outputs are finally determined as:

$$y_k = \sigma(a_k) \qquad with \qquad \sigma(a_k) = \frac{1}{1 + \exp\left(-a_k\right)} \tag{3.19}$$

whereby $k = 0, ..., K$ represents the total number of outputs. It is necessary to apply Equation 3.19 because the posterior probability for each class is desired. As dealing with 3 different classes, each output neuron represents the posterior probability for each class. y_1 represents the posterior probability for background and analogous y_2 and y_3 the posterior probability for CG and PZ. In conclusion, combining all previous steps to one overall network function for a three layer feed forward neural network results in:

$$y_k(\mathbf{x}, \mathbf{y}) = a_k^{(3)} = \sigma\left(\sum_{w=0}^{u} \theta_{kw}^{(2)} \, \sigma\left(\sum_{s=0}^{v} \theta_{ws}^{(1)} x_s\right)\right) \equiv \sigma\left(\sum_{w=0}^{u} \theta_{kw}^{(2)} \, \phi\left(\sum_{s=0}^{v} \theta_{ws}^{(1)} x_s\right)\right) \tag{3.20}$$

Looked at simply, the network model in Equation 3.20 is a non-linear function from a set of input variables $\mathbf{x} = (x_0, x_1, ..., x_v)^T$ to a set of output variables y_k influenced by a vector $\boldsymbol{\theta}$ of adjustable parameters. This function is illustrated in form of a network diagram in Figure 3.3. Because the information is flowing one way – from left to right without any loops or cycles – this network model represents a FFANN.

In Figure 3.3 the 2nd layer is also called hidden layer because it is not connected to the environment through inputs or outputs. If there are more than one hidden layers, the network is determined as deep belief feed forward artificial neural network (DBFFANN).

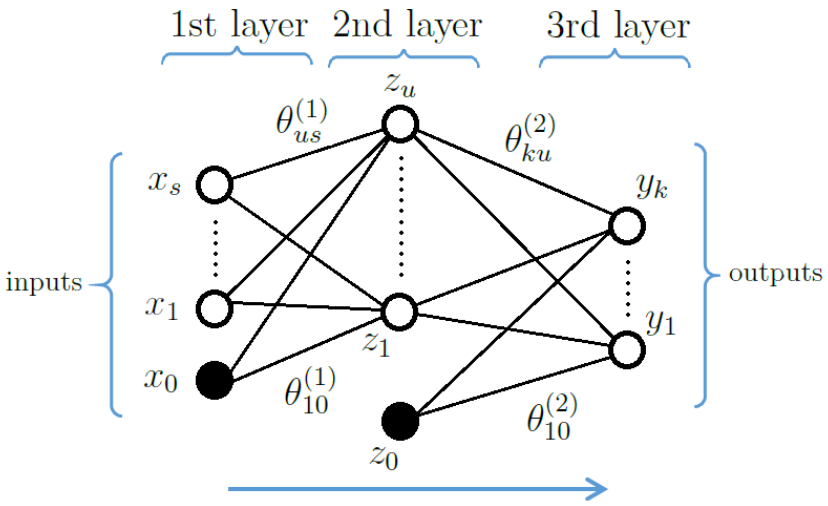


Figure 3.3: Feed forward neural network

Here, each hidden layer models a new representation of its input and consequently finds a pattern from the former layer. For each added hidden layer, a weighted linear combination of form 3.18 has to be calculated. Afterwards, an element-wise transformation using a non-linear activation function, is be applied to generate a DBFFANN.

To illustrate how a complete FFANN model looks like, Equation 3.21 provides exemplary a $3 \times 3 \times 1$ FFANN model for a non-linear single classification problem.

$$\begin{aligned} a_1^{(2)} &= \sigma\left(\theta_{10}^{(1)} x_0 + \theta_{11}^{(1)} x_1 + \theta_{12}^{(1)} x_2 + \theta_{13}^{(1)} x_3\right) \\ a_2^{(2)} &= \sigma\left(\theta_{20}^{(1)} x_0 + \theta_{21}^{(1)} x_1 + \theta_{22}^{(1)} x_2 + \theta_{23}^{(1)} x_3\right) \\ a_3^{(2)} &= \sigma\left(\theta_{30}^{(1)} x_0 + \theta_{31}^{(1)} x_1 + \theta_{32}^{(1)} x_2 + \theta_{33}^{(1)} x_3\right) \\ y_{k=2}(\mathbf{x}, \mathbf{y}) = a_1^{(3)} &= \sigma\left(\theta_{10}^{(2)} a_0^{(2)} + \theta_{11}^{(2)} a_1^{(2)} + \theta_{12}^{(2)} a_2^{(2)} + \theta_{13}^{(2)} a_3^{(2)}\right) \end{aligned} \tag{3.21}$$

Equation 3.21 represent the model's hidden units: $a_1^{(2)}$,$a_2^{(2)}$ and $a_3^{(2)}$. The input of each hidden unit is formed by the input feature vector $x = (x_0, x_1, x_2, x_3)$ weighted with $\boldsymbol{\theta}$ and transformed by the sigmoid function $\sigma(.)$

Once a neural network model of form 3.20 is generated, the next step is to find the "best" model parameters/weights during a network training. "Best" model parameters are understood as model parameters which are firstly found in a reasonable time and secondly provide the lowest misclassification rate. To achieve this, a cost function has to be defined, which is responsible for adjusting the model parameters $\boldsymbol{\theta}$ during the training phase.

Seen logistic regression models (see Equation 3.14) from a much more general view is given by the probabilistic framework. The probabilistic framework provides better inter-

pretation of training a network. When mapping an input $\mathbf{x}$ to an output $\mathbf{y}$, the output implicitly contains an error.

$$y(\mathbf{x}) = \sigma\left(\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x})\right) + \epsilon = \sigma\left(\sum_{w=1}^{u} \theta_w \phi_w(\mathbf{x})\right) + \epsilon \tag{3.22}$$

$\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x})$ represents the inner scalar product of the $\boldsymbol{\phi}$-transferred input vector and the model's weight vector. ϵ is a residual error between the linear prediction and the true response. In general, it is assumed that the underlaying distribution of $y(\mathbf{x})$ is Gaussian or normal distributed around the mean μ involving a certain variance σ^2. This means, every output y_w arises from a Gaussian distribution around the mean μ and a certain variance σ^2. Combining the linear model and the Gaussian distribution leads to:

$$p(y|\mathbf{x}, \theta) = \mathcal{N}(y|\boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}), \sigma^2) \tag{3.23}$$

Equation 3.23 points out that the model is a conditional probability density, whereby $\boldsymbol{\theta}_M = (\boldsymbol{\theta}, \sigma^2)$ determines the complete model parameter from a probabilistic viewpoint. Furthermore, it confirms the statement that discriminative models construct p(y|x) directly. Assuming the training set $\mathcal{D} = (\mathbf{X}, \mathbf{Y})$ is independent and identically distributed, the likelihood function can be constructed as follows:

$$p(\mathbf{Y}|\boldsymbol{X}, \boldsymbol{\theta}_M) = \prod_{w=1}^{u} p(\mathbf{y}_w|\mathbf{x}_w, \boldsymbol{\theta}_M) \tag{3.24}$$

Maximizing the likelihood function is understood to be the same as minimizing the sum-of-squared error function, which is illustrated in Equation 3.25. Therefore, a neural network model can be trained by either maximizing the likelihood function or minimizing the sum-of-squared error function. In this work, the sum-of-squared error function is minimized. Hence, it must be minimized for given input vectors from $\mathbf{x}_1, ..., \mathbf{x}_u$ and target vectors from $\mathbf{y}_1, ...\mathbf{y}_u$ as follows:

$$E(\mathbf{w}) = \frac{1}{2} \sum_{w=1}^{u} \|y(\mathbf{x}_w, \boldsymbol{\theta}) - \mathbf{y}_w\|^2 \tag{3.25}$$

The input and output size of neural networks is generally determined by the dimensions of the dataset and is therefore fixed. Thus, the neural network model can be adjusted by the number of hidden neurons and the number of hidden layers, which are therefore representing free parameters. Beside training a neural network model, it is important to determine the optimum values of these free parameters to achieve the best generalization performance. Finding the optimum setting corresponds to find the optimum balance between underfitting and overfitting.

The most effective algorithm to train ANNs is known as error backpropagation or just backpropagation. It is a supervised learning method, which uses gradient descent in order to minimize a sum-of-squared error function. Furthermore, it can also be seen as minimizing the error by adapting the weights. The algorithm can be divided into two phases: propagation phase and weight-update phase. These two phases are repeated until a certain error convergence is reached. The activation function of artificial neurons needs to be differentiable in order to use the backpropagation algorithm. The backpropagation algorithm represents a local optimization technique, whereby the algorithm stops when a local minimum on the error surface is found. This can lead to acceptable solutions if the local minimum is nearby the global minimum or the local minimum is itself the global minimum. If none of these scenarios appear, the output is a poorly trained network. To counteract against this, different weight initialisation techniques for FFANNs can be applied to reach local minimums nearby the global minimum considering the error surface. A fast algorithm for error backpropagation for neural networks can be found in detail in [54].

How fast the backpropagation algorithm converge among other things depends on the weight initialization. If the weights are initialized in a good manner, less iterations are necessary because the convergence of the error is reached within less iterations. For example two methods for weight initialization are either initialization with zeros or with random numbers. Another third technique is to approximate the weights in a pre-training step and then fine tune the weights in the training phase using backpropagation. Approximating the weights in a pre-training step is achieved by utilizing a Restricted Boltzmann Machine (RBM). Stacked RBMs are determined as deep belief network (DBN). DBNs can be used, inter alia, for approximating the weights of a FFANN. Once the DBN has approximated the weights, it can be transferred to a multi layer feed forward neural network, which is then trained by backpropagation. The process of transferring DBNs to FFANNs is in literature known as "unrolling", whereby the weights of a DBN are used as initial weights for a FFANN.

In this work, weights of the FFANN are initialized by three methods: zero weight initialization, random weight initialization and weight initialization using DBNs. Hence, DBNs are explained next.

3.3.2 Deep Belief Networks

Stacking RBMs enables the formation of a DBN. DBNs are used in this work for learning the weights of a FFANN. RBMs are parameterized generative stochastic models and posses the ability to learn the probability distribution of unknown input. Hence, training a RBM is understood as adjusting its parameter in a certain manner to form a probability distribution, which fits the input data as well as possible. As a DBN is comprised of stacked RBMs, each layer of a DBN corresponds to one RBM. In addition, unrolling a n-layer DBN to a FFANN results in a n-layer FFANN. In general, the areas of application of DBNs are image recognition, speech recognition and document classification [54].
RBMs are specific models derived from Boltzmann Machines with the restriction by being

a bidirectionally graph model. RBMs are comprised of two layers namely input and hidden layer. The input layer consists of visible units and analogous the output layer consists of hidden units. Visible units correspond to observations as for instance each pixel in an image represents one visible unit (see MNIST). In contrast, hidden units try to model dependencies from visible units and can also be seen as non-linear feature detectors. Units within a same layer do not have connection between each other [55], [56].

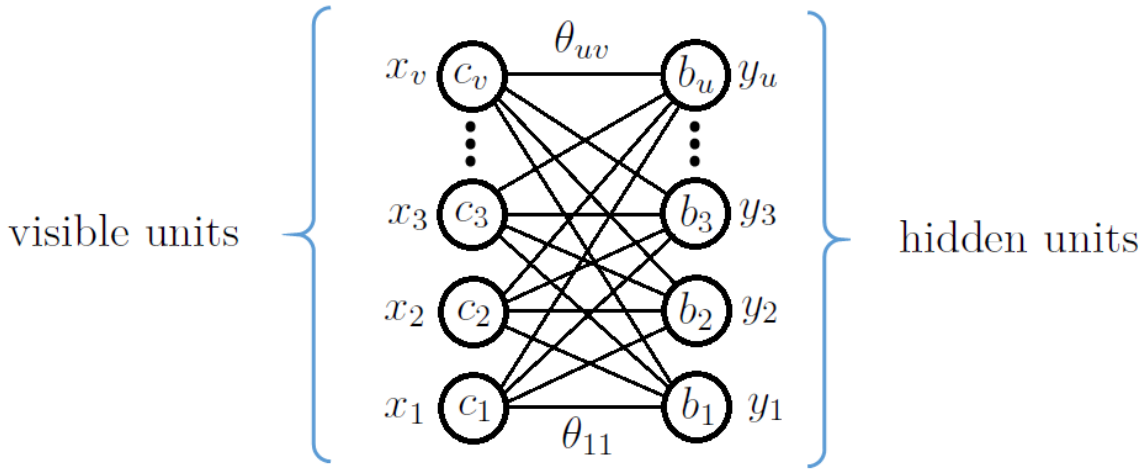


Figure 3.4: Restricted Boltzmann Machine with t visible units and k hidden units

Figure 3.4 represents a RBM, which consists of v visible units $\mathbf{x} = (x_1, ..., x_v)$ and u hidden units $\mathbf{z} = (z_1, .., z_u)$. The undirected weights, which connect visible and hidden units, are determined as $\boldsymbol{\theta} = (\theta_{11}, \theta_{12}, ...\theta_{1v}, \theta_{21}, \theta_{22}, ..., \theta_{uv})$. Furthermore, each unit has a bias, denoted as $\mathbf{c} = (c_1, ..., c_v)$ for visible and $\mathbf{b} = (b_1, .., b_u)$ for hidden units. RBMs are trained in a unsupervised greedy wise fashion using a contrastive divergence learning procedure, which is similar to the backpropagation algorithm. [55], [56]

A RBM models the joint probability distribution as follows:

$$p(\mathbf{x}, \mathbf{y}) = \frac{e^{-E(\mathbf{x},\mathbf{y}))}}{Z} \quad with \quad E(\mathbf{x}, \mathbf{y}) = -\sum_{w=1}^{u}\sum_{s=1}^{v} w_{ws} y_w x_s \; - \sum_{s=1}^{v} c_s x_s \; - \sum_{w=1}^{u} b_w y_w \tag{3.26}$$

whereby, $E(\mathbf{x}, \mathbf{y})$ represents the energy function and $Z(\mathbf{x}, \mathbf{y})$ is denoted as "partition function". The partition function is defined as the sum over all possible configurations:

$$Z(\mathbf{x}, \mathbf{y}) = \sum_{(\mathbf{x},\mathbf{y})} e^{-E(\mathbf{x},\mathbf{y})} \tag{3.27}$$

Hinton and Salakhutdinov have argued that a pre-training step utilizing RBMs to learn the weights for a FFANN helps to overcome problems observed when training a FFANN [57]. Therefore, a part of the work is dedicated to this approach.

As multi layer artificial feed forward neural networks are utilized in this work; firstly a multi layer DBN (each layer is one RBM) needs to be pre-trained to use the weights for initialization. An example for a DBN is shown in Figure 3.5.

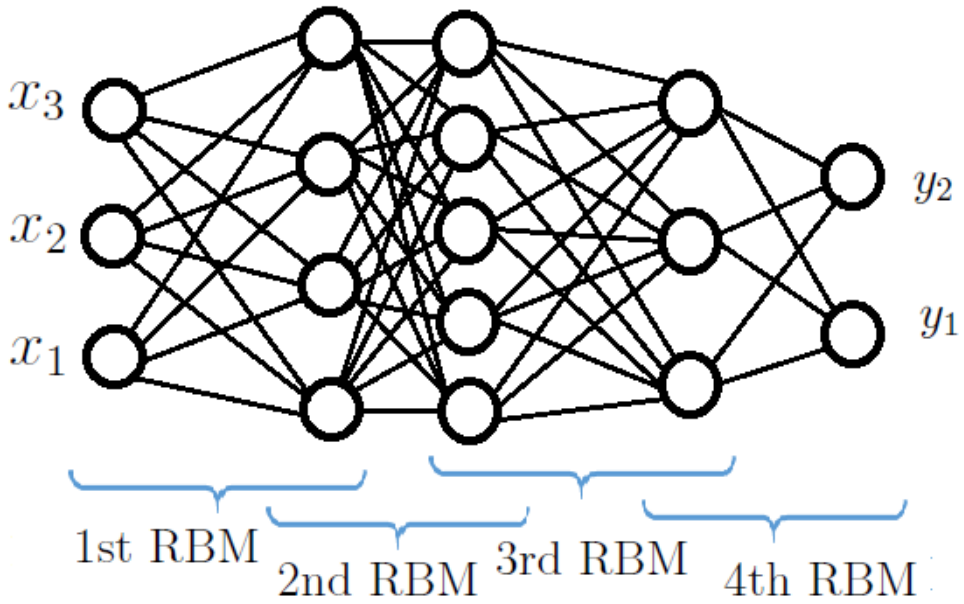


Figure 3.5: Example of a Deep Belief Network existing of 3 hidden layers and one input and one output layer

The model in Figure 3.5 consists of one input layer, three hidden layers and one output layer, whereby the output of a RBM is the input of the next RBM. The illustrated model is in total comprised of four RBMs. Once the weights are learned in the pre-training step, the weights in Figure 3.5 are then used to initialize a model of Figure 3.3.

Chapter 4

Materials and Methods

This chapter describes the appliance of the theory explained in Chapters 2 and 3 in order to enable automated prostate zone segmentation in MR images by using artificial neural networks. Hence, this chapter is structured similar to the computer vision pipeline, which is illustrated in Figure 1.1. Firstly, Section 4.1 explains the image acquisition used in this work and includes information about the utilized MR images. Next, the preprocessing Section 4.2 encloses essential basic image processing steps to receive robust segmentation results in reasonable time. Feature extraction is applied to the preprocessed MR images to generate input space clustering. Thus, Section 4.3 describes the extracted features. Section 4.4 concentrates on the decision making and classification process for the sake of realizing object segmentation and explains the proposed algorithm in detail. Furthermore, it accommodates information about, how FFANN-parameters are determined. Section 4.5 focuses on post-processing, which is performed to receive subsequently the object segmentation result. As the knowledge of the prostate's volume affect diagnostic processes positive, Section 4.6 describes the volume estimation out of the segmentation result. Lastly, for model evaluation purposes, Section 4.7 explains the utilized error metrics.

Algorithm development was done in *MATLAB*[7] utilizing the Image Processing Toolbox. Because *MATLAB* enables fast prototyping and provides a well developed image processing environment.

4.1 Image Acquisition

The algorithm was developed and tested on MR images provided by the National Cancer Institute via the public Cancer Imaging Archive[8]. Anonymized T2-weighted MR images from 50 patient studies were utilized, whereby one half was acquired with a 1.5-Tesla Philips Achieva device (from Boston Medical Center (BMC), United States) and the other half was acquired with a 3-Tesla Siemens TIM device (from Radboud University

[7]MATLAB and Image Processing Toolbox Release 2013a, The MathWorks, Inc., Natick, Massachusetts, United States.

[8]Cancer Imaging Archive, (accessed October 2013), https://public.cancerimagingarchive.net/ncia/

Nijmegen Medical Centre (RUNMC), Netherlands). BMC used an endo-rectal receiver coil whereas RUNMC utilized a surface coil for image acquisition. Each study represents adjacent axial cross-section cuts and is comprised of a $M \times M \times N$ Digital Imaging and Communications in Medicine (DICOM)-image stack, whereby image dimensions are $M \in \{400, 512\}$. The depth of the image stack N varies between 15 and 38. The xy-pixel spacing of the image stacks ranges from 0.4 mm to 0.75 mm and the z-spacing (slice thickness) is reported from 3 mm to 4 mm. Studies were imported into *MATLAB* using the provided "DICOM-read" function by the Image Processing Toolbox.

In addition to the 50 studies, corresponding labels[9] representing CG, PZ and background were provided through the Automated Segmentation of Prostate Structures Challenge. The challenge was hosted by the Cancer Imaging Program of the National Cancer Institute in collaboration with the International Society for Biomedical Imaging. The labels were marked by *Dr. Nicolas Bloch* (at Boston University School of Medicine) / *Mirabela Rusu* (Case Western U.) as well as by *Dr. Henkjan Huisman* / *Geert Litjens* / *Futterer* from RUNMC Netherlands.

Labels were published in the nearly raw raster data (NRRD)[10] file format. NRRD is designed to support scientific image processing involving N-dimensional raster data. Each NRRD-file is comprised of a header followed by the raw raster data. The header contains meta information about the raw raster data as for instance dimensions, encoding, space direction, spacing and type of the raw raster data. As a DICOM image stack of one study has the dimension $M \times M \times N$, the corresponding label stack must have the same dimensions. Therefore, a third dimension named l was added to the existing Equation for labels in 3.3. This results in a label stack of form:

$$\boldsymbol{L}(i,j,l) = k, \ for \ \boldsymbol{L}(i,j,l) \in \{0,1,2\} \ \ and \ \ \forall (i,j,l) \in \ \{(i,j,l) | 1 \leq i \leq M, 1 \leq j \leq M, 1 \leq l \leq N\} \tag{4.1}$$

The raw raster data of each NRRD-file is provided in 4.1, whereby $k = 1$ represents PZ, k = 2 stands for CG and k = 0 corresponds to background. NRRD files are imported into *Matlab* using a "NRRD-read" function provided by *Jeff Mather*[11]. The design matrix $\mathbf{X}$ is generated out of the image stacks and analogous matrix $\mathbf{Y}$ is formed based on the label stacks. $\mathbf{X}$ and $\mathbf{Y}$ represent together the dataset $\mathcal{D}$ whereby, $\mathcal{D}$ is the core of the training phase.

After importing an image stack into *Matlab*, regardless of whether it is a training or prediction study, a modality transformation and a gray value transformation in form of a windowing function are applied. Parameters for these transformations are extracted

[9] NCI-ISBI 2013 Challenge - Automated Segmentation of Prostate Structures, (accessed October 2013), http://goo.gl/OBdxpq

[10] NRRD Nearly Raw Raster Data, (accessed October 2013), http://teem.sourceforge.net/nrrd/

[11] NRRD Format File Reader, (accessed October 2013), http://goo.gl/IdQu65

from the corresponding DICOM header. As values in the DICOM image represent stored values (SV), they need to be transformed to receive real world values (RWV). Hence, each pixel in a DICOM image is transformed through a modality transformation to obtain RWV as follows:

$$\text{RWV(i,j,l)} = \textit{slope} \times \text{SV(i,j,l)} + \textit{intercept} \tag{4.2}$$

Whereby *slope* and *intercept* are two device specific tags from the DICOM header. The modality transformation is necessary because values in the DICOM image can be stored as device specific values so that only the device itself is able to interpret the values. To overcome this, equation 4.2 is applied to transform the SV to meaningful RWV. In the next step, a windowing function is applied in order to visualize the prostate zones best possible for the human eye. Thus, RWVs are transferred to display values (DISP). Window Center (Win_{center}) and Window Width (Win_{width}) are again two tags out of the DICOM header and specify a gray value range, which should be visualized with high contrast. Therefore, gray values inside the range are mapped between one and zero according to Algorithm 1. In contrast, gray values outside the window are mapped either to 0(black) or 1(white).

Algorithm 1 Windowing Function

Require: $RWV(i,j,l)$, Win_{width}, Win_{center}

if $RWV(i,j,l) \leq Win_{center} - (Win_{width} * 0.5)$ **then**
  $DISP(i,j,l) = 0$
else if $RWV(i,j,l) > Win_{center} + (Win_{width} * 0.5)$ **then**
  $DISP(i,j,l) = 1$
else
  $DISP(i,j,l) = ((RWV(i,j,l) - (Win_{center} - 0.5))/(Win_{width} - 1)) + 0.5$
end if

As the prostate is the organ of interest, it is best visible for the observer after applying the windowing function. The output values abbreviated as DISP in Algorithm 1 are called display values and are representing gray values utilized for any further processing in this work.

4.2 Preprocessing

The second step in the computer vision pipeline illustrated in Figure 1.1 is preprocessing. Preprocessing is applied in this work because of three reasons. Firstly it is to obtain prior knowledge from a training set, whereby mainly label stacks of the training set are utilized for gaining prior knowledge. Preprocessing is applied secondly to reduce the amount of input data. Reducing the amount of input data can also be seen as lowering the computational complexity, which enables as a consequence segmentations in reasonable time. The third reason for preprocessing is to normalize the input data in order to

achieve subsequently robust segmentation results. This section describes how these three preprocessing steps are realized.

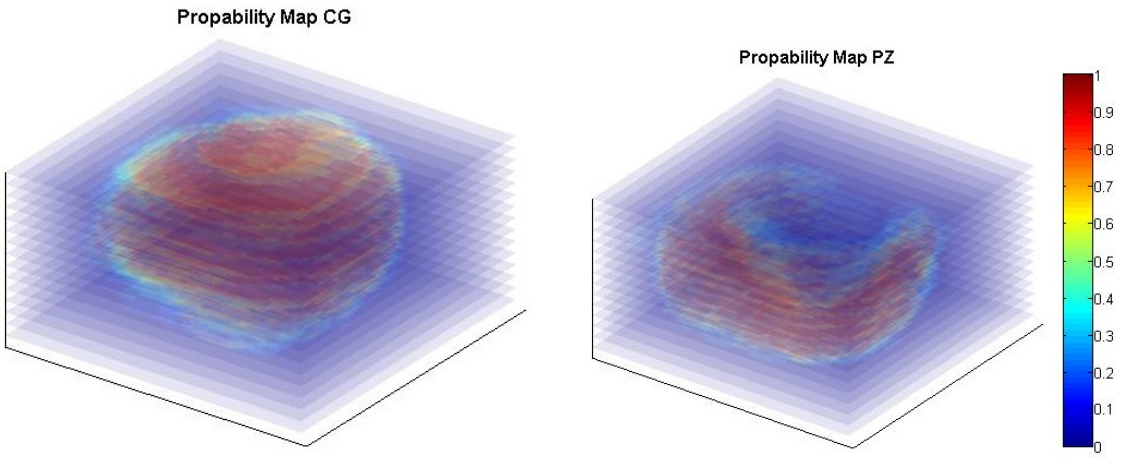
Because the prostate is the organ of interest in the provided MR image stacks, it is located in the middle of the image and in the middle of the image stack (approximately at $M/2 \times M/2 \times N/2$). Thus, the image stack borders are not relevant for prostate segmentation issues and hence, border pixels are not considered. To know which border pixels can be rejected for prostate segmentation, a maximum bounding box is learned based on the label stacks. For each case the bounding box, which includes the prostate zones completely, is assessed. Next, the maximum bounding box over all label stacks is determined. Moreover, a certain tolerance is added to ensure that large prostates in the prediction set lie inside the learned maximum bounding box. The maximum bounding box is then used to exclude unimportant border pixels in order to reduce the number of pixels and respectively the number of inputs for the FFANN model. In a further preprocessing step to reduce the amount of input data, the remaining cropped image stacks are down-sampled by factor 2. To preserve edges and suppress noise, the down-sampled image stacks are median filtered using a kernel size of 3×3. For example a case with image stack dimensions of $500 \times 500 \times 20$ is cropped with a bounding box, which results in $300 \times 300 \times 14$ dimensional image stack. Afterwards the stack is down-sampled which yields to a image stack of dimensions $150 \times 150 \times 14$. In this example the number of pixels in the image stack is reduced from 5 000 000 to 315 000. Hence, 315 000 pixel-classifications have to be made in this example.

In addition to a learned maximum three dimensional bounding box, an average mass point of the prostate gland is learned based on label stacks. Therefore, the mass point from each label stack (from the training set $\mathcal{D}$) is determined in order to assess finally the average mass point. The average mass point is utilized as a reference point for feature extraction, which is described in the next section.

Finally, two probability maps are learned from the training label stacks. This process can also be seen as learning an average central gland (CG) and an average peripheral zone PZ. Considering both prostate zones, two probability maps are learned. These maps determine for every pixel position in a stack how probable it is that the current pixel is a CG pixel or a PZ pixel. Because of dealing with different stack dimensions, the probability maps are normalized through a mapping on a cube of sides 1.

Figure 4.1 represents the probability map of the CG on the left 4.1(a) an the PZ on the right 4.1(b). Both maps are learned based on 20 label stacks out the training set. In Figure 4.1, red indicates high probability and blue low probability.

The maximum bounding box, the average mass point, the probability maps and the learned weights of the FFANN model through the training phase form together the obtained prior knowledge utilized in this work to achieve more accurate segmentation results.



(a) (b)

Figure 4.1: Probability map of the CG in (a) and of the PZ in (b) assessed from 20 training label stacks. Red indicates high probability and blue low probability.

4.3 Feature Extraction

Using just the raw gray values (DISP(i,j,l)-values) of the MR image stacks as a feature results in poor classification and thus in poor prostate segmentation (see end of Section 3.3). This is because no prostatezone-specific gray value clustering appears in the input space and accordingly a classifier cannot differentiate between prostatezones-pixels or background-pixels. To overcome this issue, describing features for each pixel are extracted to generate input space clustering in order to enable a classifier partitioning the image stack into meaningful regions. Feature extraction is mainly used for dimensionality reduction, but in this work it is utilized to create a separable input space. Hence, relevant information describing the prostate zones is extracted in order to segment the zones through classification. If features would be extracted optimal, the classification process would result in a trivial solvable linear decision problem.
In total are 28 describing features for each pixel extracted. Referring to Equation 3.5, the design matrix $\mathbf{X}$ is therefore a $28 \times number\ of\ pixels$ dimensional matrix. Features are extracted utilizing a feature extraction function $f^{\mathbf{I}}$ on each pixel in an image stack $\mathbf{I}_p(i,j,l)$ as illustrated in Equation 4.3. The feature extraction function is applied to both – training and prediction studies – to generate input data for a FFANN.

$$\mathbf{X} = f^{\mathbf{I}}\left(\sum_{p=1}^{P} \mathbf{I}_p(i,j,l)\right) \tag{4.3}$$

P represents the total number of training studies or prediction studies. Thus, feature extraction is applied to every pixel in all image stacks. The 28 utilized features represent hand-engineered low-level features. Each feature is extracted based on a different method. All methods are combined in one overall feature extraction function $f^{\mathbf{I}}$ in Equation 4.3.

Consequently, the rest of this section is dedicated to the feature extraction in order to generate input space clustering.
The first feature is basically device dependent and is called the magnetic field strength feature. It determines whether the image stack was acquired with a 3 Tesla or 1.5 Tesla device. Image stacks acquired with a 3-Tesla device receive a value of 1 and in contrast image stacks acquired with a 1.5-Tesla device receive a value of 0. Accordingly, every pixel in an image stack has the same magnetic field strength feature value. The next six features determine different positions from the pixel of interest within the image stack. These features are denoted as position features. The first three position features are the x, y, and z positions of each pixel within the image stack. For instance, the first pixel in the stack is assigned a value of (1,1,1) and the last pixel in the stack position is allocated to (100,100,14), when considering an image stack of $100 \times 100 \times 14$ dimensions. The next three out of six position features determine the position from the pixel of interest considering a spherical coordinate system. Thus, each pixel is described using (r, θ, φ) radial distance r, polar angle θ and azimuthal angle φ. (r, θ, φ) are shown in Figure 4.2. It is believed that position features are utilized to learn the shape and the position of the prostate.

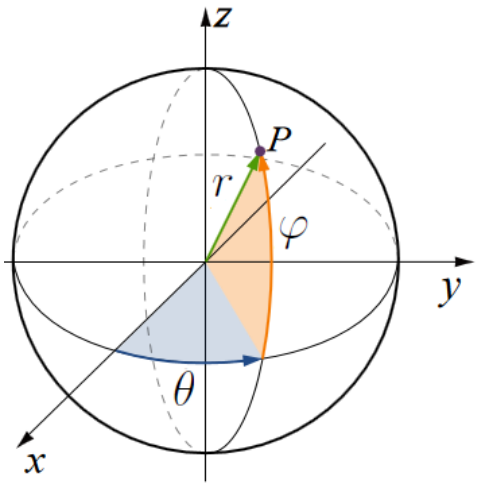
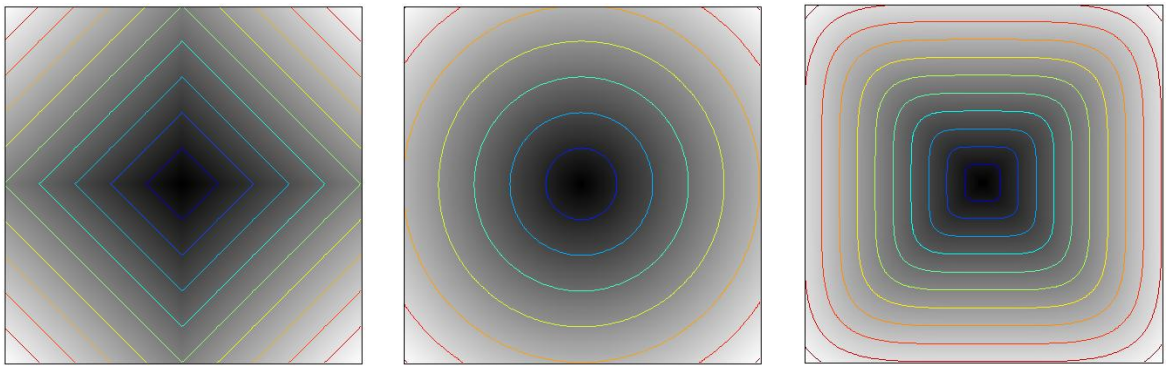


Figure 4.2: Three dimensional spherical coordinate system, whereby the position of a point P is determined with (r, θ, φ)

The next six features are distance features considering xy-pixel spacing and slice thickness. As explained in section 4.2 an average mass point is learned and used as reference point for distance calculations. X, y and z distances from the pixel of interest to the average mass point are respectively three features. The next three out of six distance features are derived from the Minowski distance ϑ. The Minowski distance between two points P and Q is defined as follows in 4.4:

$$\vartheta(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} \quad with \quad P = (x_1, ..., x_n) \quad and \quad Q = (y_1, ..., y_n) \in \mathbb{R}^n \quad (4.4)$$

whereby p represents an integer value, which determines different distance metrics. Setting p in Equation 4.4 to 1 results in the Manhattan Distance (City Block Distance) and setting the value p to 2 provides the equation to calculate the Euclidean distance. Both are utilized as features. The third and last distance feature is assessed by setting p to 3, which yields in a cubical distance profile. Two-dimensional examples of all three utilized Minowski distances are illustrated in Figure 4.3. According to that, distances are calculated with respect to the image center. It is supposed, that the utilized distance features have the highest impact on the decision making process.



(a) Minowski distance p=1 (b))Minowski distance p=2 (c) Minowski distance p=3

Figure 4.3: Two dimensional examples of utilized distance metrics derived from Equation 4.4. For visualization purposes, iso-contours are added. (a) Minowski distance with p=1 results in the Manhattan distance; (b) Minowski distance with p=2 provides the Euclidian distance; (c) Minowski distance with p = 3

Next, the first of the six gray value features contains basically the SV(i,j,l), before the transformation functions (see Algorithm 1 and Equation 4.2) are applied. The remaining five gray value features are different gray value features derived from each pixel DISP value itself. Hence, the second gray value feature is a median filtered version derived from DISP(i,j,l). The third feature contains the gray value of a contrast enhanced version and the fourth feature is represented by a local standard deviated version. Next, the fifth feature includes information about the global variance based on a specific three dimensional neighborhood and ultimately the sixth feature comprises the local entropy.

The upcoming features are containing gray values, depending on a certain neighborhood size (NHS). Possible NHSs are 1, 3, 5 and 7, which results in either 1(1×3×1), 27(3×3×3), 125(5×5×5) or 343(7×7×7) neighborhood gray values by taking the gray value of the pixel of interest into account, too. Respectively, either 1, 27, 125 or 343 gray values are considered as features. For now a NHS of 1 is pre-assumed which means that the display value of the pixel of interest is accounted as a feature. As previously mentioned 28 features are extracted in total which is based on a pre-assumed NHS of 1. For instance, considering a NHS of 3 would result in 27+27 = 52 features. Which NHS value provides the best segmentation results is shown in the the result Chapter 5.

The next six from the in total 28 features are mean gray value features deducted from surrounding six-connected neighbor cubes. Hence, the just described pixel of interest is surrounded by its neighbors, which are forming a cube with the pixel of interest as a center. Each side of the cube is touched to another neighbor-cube of the same NHS. This results in six neighbor cubes and corresponds to six mean gray values computed from each neighbor cube. Consequently, the six mean gray values represent six features. By utilizing these features, surrounding information is integrated into the decision making process for each pixel decision. It is believed, that this procedure improves segmentation results.

Once all of the above described features are extracted, each of these is mapped to values in the range between 0 and 1 to treat them as possibilities and to make them invariant to image resolution, image stack dimensions, image stack parameters (xy pixel spacing, slice thickness). Afterwards, the bottom 1% of all feature values are mapped to 0 and analogous the top 1% are mapped to 1. The rest of the feature values are stretched via a mapping function to enhance feature clustering. This procedure improves classification results. Two features are remaining to get to the total 28 describing features, considering a NHS of 1. These two features are probability features, describing how likely it is, that the current pixel belongs either to CG or PZ. Therefore, the former learned probability maps in the preprocessing step are utilized to form the two remaining features.

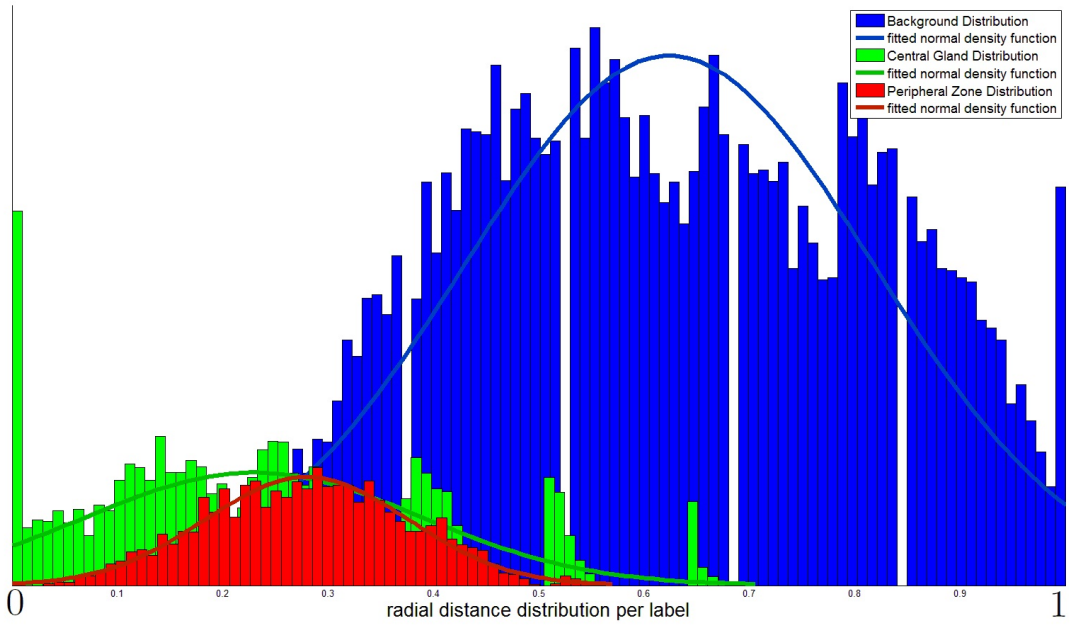


Figure 4.4: Distribution of the radial distance feature of one exemplary study, whereby blue indicates the radial distance distribution of background pixels respectively, green visualizes the radial distance distribution of CG pixels and red represents the radial distance distribution of PZ pixels. Radial distance feature is the range between 0 to 1 and causes clustering. Furthermore, a normal density function is fitted to each distribution for illustration purposes.

To determine if the proposed feature extraction causes input space clustering, the radial distance feature of one study is exemplarily illustrated in Figure 4.4. The x-axis in Figure 4.4 represents the radial distance in the range between 0 an 1. The y-axis shows the incidence of the radial distance feature. In addition, a normal density function is fitted to each distribution. The prostate's mass point is denoted by 0. Therefore, close to 0 many green CG pixels can be found, because the mass point is surrounded by CG pixels. Starting from the prostate's mass point and going outwards to the CG border, blue background pixels begin to appear in a certain distance. Red PZ pixels occur in between. Deducted from Figure 4.4, CG pixels occur approximately as often as PZ pixels. Hence, both zones have almost the same area in this study. The radial distance feature enables a way to separate CG pixels from others. In contrast, the feature distribution of the PZ does not reveal such clear clustering (see Figure 4.4). But it is supposed that the FFANN model can capture PZ pixels in combination with other features, as for instance position features.

After 28 features (pre-asuming a NHS of 1) from each pixel of all training studies are extracted, matrices $\mathbf{X}$ and $\mathbf{Y}$ are formed to generate the training set $\mathcal{D}$ in order to train a FFANN whereby, $\mathbf{Y}$ contains the corresponding labels (see Equation 3.4, 3.5 and 3.6 in Section 3.2). To predict a new study, matrix $\mathbf{X}$ is extracted from prediction studies utilizing the gained prior knowledge. Then, $\mathbf{X}$ is fed into the trained FFANN model to predict $\hat{\mathbf{Y}}$. Thus, the next section describes the pixel decision making process in detail.

4.4 Decision Making

Decision making is performed through classification utilizing a FFANN model. Because of fast prototyping reasons, the current *DeepLearnToolbox* version, developed and provided by R.B. Palm [9] on GitHub[12], is utilized. The toolbox includes amongst other things a framework about deep belief network and vanilla neural networks. The latter is mainly used in this work, whereby 'vanilla' indicates that once a network is trained, it has fixed weights. This means that during the prediction phase, weights are not going to be changed. By using more than one hidden layer, they can be treated as multi layer FFANNs. Section 4.4.2 describes the proposed algorithm in detail.
In this work a FFANN model is trained in a supervised fashion using training data including expert labels to enable subsequently the prediction of new input. To discover the best FFANN setting, several experiments are carried out. Firstly, the focus is on finding the optimum number of hidden layer and hidden neurons and secondly finding an optimum weight initialization method. There are further FFANN parameters like batch size and epochs, but they only become important when the previous parameters are found. The next section 4.4.1 describes how the numbers of hidden layer and hidden neurons are found.

[12] *DeepLearnToolbox* by R.B.Palm, (accessed October 2013), https://github.com/rasmusbergpalm

4.4.1 Network Settings

The input and output size of neural networks are in general determined by the dimensions of the dataset and consequently fixed. In this work, the output is fixed to three output neurons because of three classes (CG, PZ and background). Respectively, the number of input neurons is set to 28, because 28 features are utilized. Thus, a neural network model can only be adjusted by the number of hidden neurons and the number of hidden layers, which are therefore called free parameters. Apart from training a neural network model, it is important to determine the optimum values of these "free parameters" in order to achieve the best generalization performance. Unfortunately, there is no common rule on how to initialize values of these "free parameters". Hence, they need to be established experimentally. Finding the optimum parameters corresponds to finding the optimum balance between underfitting and overfitting.
For instance, a 5 layer feed forward artificial neural network (FFANN) is considered in this work to consist of one input layer, three hidden layers and one output layer. Whereby, 28-10-10-10-3 determines 28 input neurons in the first layer, 10 hidden neurons in the second layer, 10 hidden neurons in the third layer, 10 hidden neurons in the fourth layer and 3 output neurons in the fifth layer. As 28 features are extracted for each pixel, the number of input neurons of the FFANN is fixed to 28, considering a NHS of 1. The second, third and fourth layer form together the hidden layers. The output layer corresponds to three classes (CG, PZ and background) and can also be seen as the posterior probabilities for the corresponding class. In general, there are three different types of neural network shapes for hidden layers:

- straight shape corresponds to 50-50-50
- linear shape corresponds to 50-30-10
- exponential shape corresponds to 50-10-5

To discover the optimal shape, three experiments are carried out. The first one is to determine the optimal number of hidden neurons. The second experiment is to determine the optimal number of hidden layers and lastly to find the optimal network shape. In the next step, it is necessary to assess if random weight initialisation or weight initialisation using DBNs leads to more accurate results. After these experiments are performed, optimal parameters for a FFANN model are found.

Furthermore, it is necessary to establish at which point the model is trained with enough training cases to perform steadily reasonable on new data. Therefore, multiple training iterations are proposed, whereby each iteration utilizes a increasing number of training studies.

Once these parameters are specified, it is necessary to assess the number of epochs and the batch size. The optimal batch size and optimal number of epochs are defined as tradeoff between the mean squared error and the computation time. One epoch is understood as one sweep through of the whole training set to the FFANN. It is supposed that segmentation results improve when the model is trained with an increasing number of epochs. As the chosen mode for backpropagation is batch learning, the input space is

divided into batches. Each batch is fed into the FFANN, which means that the weights will be updated by the backpropagation algorithm after every batch and not in comparison to online training after every input vector. Moreover, an experiment to determine the optimal NHS is performed as well.

Finally, the optimal parameterized FFANN model is tested on 25-3Tesla studies and on 25 1.5-Tesla studies. Furthermore, the FFANN model is trained and tested on 100 MD Anderson Cancer Center[13] studies. For this purpose, radiologists drew contours manually using a provided *Matlab* framework.

4.4.2 Two Layer Topology

This section represents the proposed algorithm pipeline and describes therefore the core of this work. Because the feature vector is comprised of six distance features (see Section 4.3), which strongly depend on the prior obtained average mass point and the fact that prostates show high natural variability in shape and size, the classification output of one FFANN model does not produce accurate results. It is considered that distance features are important features, which have a higher impact on the decision making process than others as for instance gray value neighborhood features. To overcome the inaccuracy, a second FFANN model is attached to the first FFANN model, which results in a consecutive two layer topology. The first FFANN is determined as localization layer because the aim is to localize the prostate in the image stack. The second FFANN is denoted as the labeling layer, whereas this model should label the prostate zones. Because two FFANN models are utilized, each one needs to be trained. The main advantage of the two layer topology is the improvement of precision concerning distance features in the second layer. This fact especially affects studies containing either an abnormal large prostate or an abnormal small prostate, because both are treated after the first layer as a normal sized prostate. The proposed algorithm pipeline (see Figure4.5) contains the training as well as the predication phase of the localization and labeling layer.

The following text explains the proposed algorithm, which is comprised in total of of nine steps. In addition, each step is marked in Figure 4.5 for illustration purposes. Steps 1-4 represent the algorithm's training phase. In the first step, prior knowledge is obtained from $\mathbf{Y}_{1st}$. This procedure involves acquiring the maximum bounding box, the average mass point and the probability maps. Afterwards, in step 2 training studies are preprocessed, which is associated with reducing the amount of input data and performing gray value transformations. Hence, training studies are cropped by the previously learned maximum bounding box. Then features including the probability maps are extracted. Thereby, the average mass point is used as a reference point (see Equation 4.3) in order to generate the training set $\mathcal{D}_{Localization} = (\mathbf{X}_{1st}, \mathbf{Y}_{1st})$ for the first FFANN. The first FFANN is also denoted as localization layer. Next, the first FFANN model (localization layer) is trained utilizing $\mathcal{D}_{Localization}$. Summarized, step 2 covers the training of the localization layer. These just described steps are party analogous for the second FFANN.

[13]MD Anderson Cancer Center, (accessed October 2013), http://www.mdanderson.org/

Next, steps 3 and 4 illustrate the training of the second FFANN model. Whereby, step 3 focuses on preprocessing. But during this step, the training studies are cropped with their current bounding box. The current bounding box is assessed from each label stack itself. Afterwards, features are extracted utilizing their current mass point as a reference point for distance features. This procedure is different to the first layer, in which training studies are cropped by the learned maximum bounding box over all studies and distance features are computed using the average mass point over all studies. Using from each predicted study its current bounding box and current mass point for feature extraction represents therefore the main difference between the localization and labeling layer. Adding the obtained probability maps in step 4 completes the training set $\mathcal{D}_{Labeling} = (\mathbf{X}_{2st}, \mathbf{Y}_{2st})$ for the second FFANN. Then the second layer is trained with $\mathcal{D}_{Labeling}$. Summarized, steps 1-4 describe the training phases of both layers.

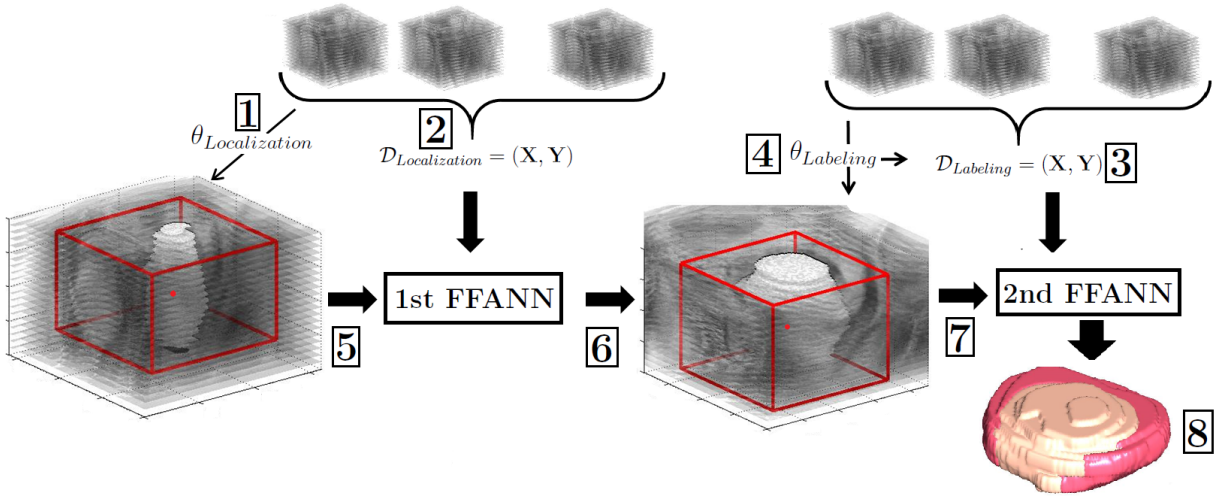


Figure 4.5: Illustration of the algorithm pipeline, which consists of 9 steps. In the first step prior knowledge including the maximal bounding box, the average mass point and the probability maps are obtained. Step 2 covers preprocessing and training set generation as well as the training of the first FFANN model (localization layer). Analogous, prior knowledge is obtained from the same data set for the second layer. Afterwards, a training set is generated for the second layer, which is different to the first layer. Then training of the second FFANN model (labeling layer) is performed (step 4). The remaining steps represent the processing of new input. Features are extracted from new input in step 5 and fed into the first FFANN. The predicted prostate zones are then cropped with the current bounding box (step 6). Next, features are extracted utilizing the predicted mass point and fed into the second FFANN (step 7). After applying postprocessing, the output of the second FFANN, namely the final algorithm result is received (step 8).

Steps 5-8 in Figure 4.5 describe the algorithm's prediction phase. The prediction phase can only be executed after the training phase was performed successfully beforehand. The following passage explains the prediction of one study. The algorithm is only able to predict one study at a time. For predicting n-studies, steps 5-8 need to be repeated

n-times.
A new study $\mathbf{I}(i,j,l)$ is cropped using the maximum bounding box learned from the training studies in step 1. This results in an image stack of form: $\mathbf{I}_{MBB}(i,j,l)$. Then, preprocessing and feature extraction $\mathbf{X}_{1pred} = f^{\mathbf{I}}(\mathbf{I}_{MBB}(i,j,l))$ is performed in order to generate the prediction set $\mathbf{X}_{1pred}$ for the first FFANN. Thereby, distance features are extracted with respect to the average mass point, which was assessed in step 1. Consequently, the prediction set $\mathbf{X}_{1pred}$ is fed into the first layer (step 5) as to predict a label stack through $\hat{\mathbf{Y}}_{1st} = h_{1st}(\mathbf{X}_{1pred})$. Thus, $\hat{\mathbf{Y}}_{1st}$ represents the predicted label stack of the first layer.
Utilizing the current bounding box based on $\hat{\mathbf{Y}}_{1st}$, the input image $\mathbf{I}(i,j,l)$ is cropped again, which results in $\mathbf{I}_{BB}(i,j,l)$. The image stack $\mathbf{I}_{BB}(i,j,l)$ becomes usually smaller than the image stack cropped with the maximum bounding box $\mathbf{I}_{MBB}(i,j,l)$ from the first layer. Furthermore, step 6 covers the establishment of the mass point from $\hat{\mathbf{Y}}_{1st}$. The recently determined mass point and the cropped image stack $\mathbf{I}_{BB}(i,j,l)$ are then utilized for feature extraction. Step 7 covers extracting features of a prediction study in order to generate the input set for the second FFANN, which has the following form: $\mathbf{X}_{2pred} = f^{\mathbf{I}}(\mathbf{I}_{BB}(i,j,l))$. Next, $\mathbf{X}_{2pred}$ is fed into the second layer. Finally, step 8 provides the predicted labels by the second FFANN $\hat{\mathbf{Y}}_{2st} = h_{2st}(\mathbf{X}_{2pred})$. Subsequently, postprocessing (see Section 4.5) is performed to receive the final segmentation result.

The proposed algorithm does not need any user interaction and represents consequently a fully automated algorithm to achieve prostate zone segmentation. The algorithm takes a special position compared to semi-automatic and manual approaches, which were provided at the *Automated Segmentation of Prostate Structures* - challenge in April 2013.

The algorithm in Figure 4.5 is determined as *A2-Mode*, which stands for automated mode using two layers. There are two more modes in the implemented version for comparison purposes. However, the main focus is still on the *A2-Mode*. Considering only the result of the first FFANN is denoted as *A1-Mode*, which stands for automated mode using the first layer. The *A1-Mode* mode produces the most inaccurate results compared to the two remaining modes, because of the natural variability of the prostate. Nevertheless, when using the *A2-Mode*, the output of the first FFANN affects the performance of the second FFANN massively. Hence, special attention is directed to the robustness of the first layer. If the cropping between the first and the second layer is done imprecisely, the second layer produces poor results. Finding parameters for the first layer can also be seen as finding parameters for the second layer. The *A2-Mode* is used for experiments to determine the optimal FFANN parameters.
The third and last mode is denoted as the *I Mode*, which stands for interactive mode and represents therefore a semi-automated version. The *I-Mode* requires the determination of the bounding box and the mass point manually through an user. Thus, the *I Mode* does not represent an automated mode like the *A1-* and *A2-Mode*. Because of the user interaction the *I-Mode* produces the most accurate segmentation results. Regarding to Figure 4.5 the *I-Mode* only utilizes steps 3,4,7 and 8. Because the goal is to develop an automated algorithm, the *I-Mode* is only used for comparison purposes.

Looking at the *A2-Mode* mode from a different view, the goal of the first FFANN corresponds to the user interaction using the *I-Mode*, which is namely defining the current bounding box and the current mass point from the prediction study. Accordingly, the *A2-Mode* produces results, which lie, depending on the accuracy, between the *A1-Mode* and the *I-Mode*.

In conclusion the algorithm's aim is to train a FFANN based on training studies utilizing expert labels and to predict afterwards new cases. Therefore, experiments are carried out to capture the performance of different algorithm setups based on various training sets.

4.5 Postprocessing

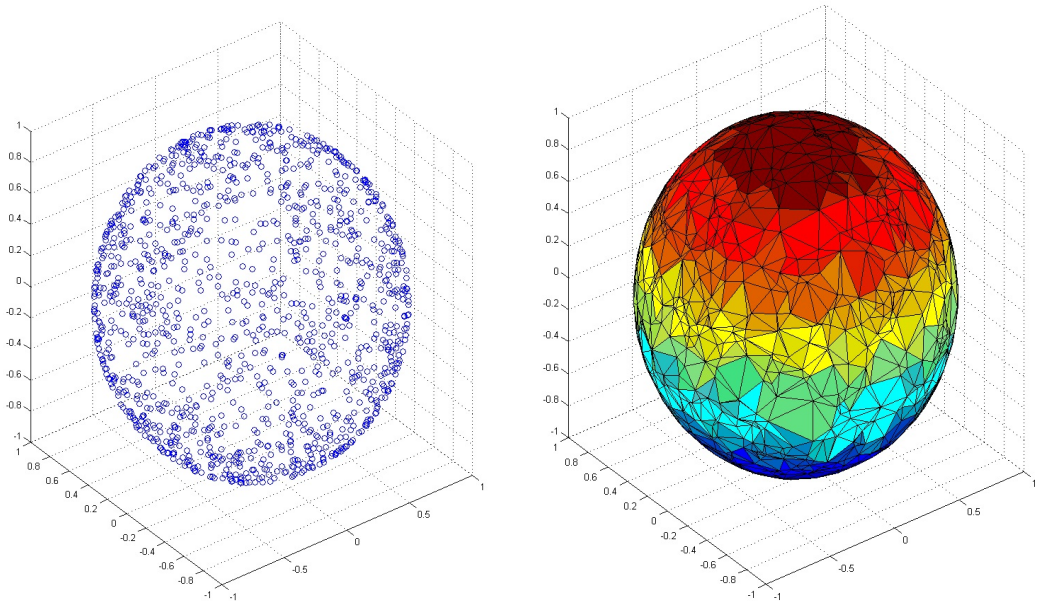
Each slice in the predicted image stack is usually comprised of labels for CG and PZ. Hence, the predicted image stack is split into two corresponding binary image stacks. Afterwards, postprocessing is applied to them due to three reasons; firstly, to get rid of prediction outliers, secondly to close wholes in the predicted prostate zones and thirdly to smooth the prostate zone shape. The first two issues are solved by applying morphological opening (erosion followed by dilation) on each slice of the two binary image stacks. The third issue is addressed by applying a three dimensional Gaussian smoothing utilizing a filter kernel of size [3 3 3]. Then both binary label stacks are merged to one and re-scaled to the original label dimensions $M \times M \times N$, which finally represents the output of the proposed algorithm. Because of the participation on the NCI-ISBI prostate segmentation challenge, the output label stack was saved as NRRD-file, for which a "NRRD-save" function was implemented in *Matlab*.

4.6 Prostate Volume Estimation

Volume estimation of the CG and PZ are carried out in this work. Thus, surface points are extracted from the predicted label stack and transformed to three-dimensional grid points considering the xy-spacing and z-spacing. The resulting 3D surface point cloud is triangulated using a Delaunay[14]-Triangulation. In the next step, tetrahedrons are created by connecting the surface triangles with a reference point (for instance the object's mass point). Afterwards, the volume of each tetrahedron is computed and summed up, which yields to the total volume of an object. By considering the triangles' normal direction (either points towards the reference point or not), the tetrahedrons' volume is either taken negative or positive into account. This consideration enables volume estimation of concave three-dimensional objects like the peripheral zone. The proposed volume estimation algorithm was developed during my bachelor thesis [58]. To test the volume estimation algorithm on synthetic data, Figure 4.6 illustrates an example.

Figure 4.6(a) represents a synthetic three dimensional surface point cloud of an unit ball. The volume of an unit ball is $4,1887$ VU. Triangulating the three dimensional surface point cloud in 4.6(a) leads to figure 4.6(b). The subsequent volume estimation from a

[14]Boris Nikolajewitsch Delone, Russian Mathematician in 19th century



(a) (b)

Figure 4.6: Delaunay Triangulation of a synthetic three dimensional point cloud with subsequent volume estimation. (a) synthetic surface point cloud of a sphere with diameter of 1; (b) Delaunay Triangulation of (a) with consequent volume estimation from a reference point $r_p = (2, 2, 2)$. A unit ball has a volume of $3\pi/4 = 4,1887$ VU. The estimated volume of (b) is $4,1497$ VU

reference point $r_p = (2, 2, 2)$ results in $4,1497$ VU. Increasing the surface point density would yield to a approximation to the "ground truth" volume, which is $4,1887$ VU for the unit ball. It is hypothesized that the accuracy of the proposed volume estimation algorithm is precise enough for clinical usage. The proposed volume estimation algorithm is compared to current clinical standard prostate volume estimation techniques, which are namely the Ellipsoid [59], Myschetzky [60] and Prolate spheroid [59] technique.

4.7 Error Metrics

To evaluate the performance of a setup, five error metrics are taken into account. To each of the two segmented prostate zones of a predicted label stack, error metrics calculations are applied. Whereby, error metrics are determined in comparison to ground truth. The utilized five error metrics are as follows:

1. Dice coefficient (DC)
2. Sensitivity
3. Specificity
4. positive predictive value (PPV)
5. Hausdorff Distance of Boundaries (HdB)

Because four out of the five error metrics are statistical parameters for binary classifier evaluation, Figure 4.7 is introduced considering four possible cases for each predicted pixel: false positive (FP), true positive (TP), false negative (FN) and true negative (TN). Hence, the yellow result set in Figure 4.7 represents the predicted binary label of one slice. The green circle in Figure 4.7 represents the corresponding ground truth slice. The red zone indicates the amount of pixels, which are overlapping and thus correctly classified through the FFANN. The surrounding white stands for the correctly classified background pixels. Furthermore, the green pigmented "FN area" in 4.7 determines the amount of pixels which are indeed ground truth pixels, but are not correctly classified as prostate zone pixels through the FFANN. Accordingly, the yellow "FP area" in 4.7 points out the amount of pixels which are classified through the FFANN as prostate zone pixels, but actually are not prostate zone pixels.

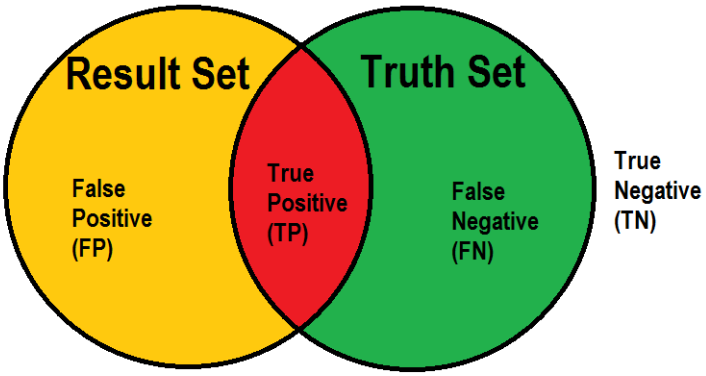


Figure 4.7: Statistical classifier evaluation considering four cases: false positive (FP), true positive (TP), false negative (FN) and true negative (TN)

The following paragraph describes the proposed error metrics calculation algorithm. As shown in the first line in Algorithm 2, the predicted and post-processed output $\hat{\mathbf{Y}}_{2st}$ is separated into label stacks. $q = (1, ..., Q)$ illustrates the total number of prediction studies and $l = (1, ..., l)$ represents the total number of slices of each prediction study.
Afterwards, each study is split into two binary label stacks – one represents the prediction of the CG and the second one represents the prediction of the PZ. The division is performed to evaluate CG and PZ separately. Then error metrics calculation is applied to each slice to each of the two binary label stacks over all cases. Each column in the error matrix $\mathbf{E}_{CG}$ represents one metric. Next, the mean is calculated for each error metric over all slices. Consequently, this procedure results in 5 error scores for the CG ($\mathbf{Score}_{CG}$) and in 5 error scores for the PZ ($\mathbf{Score}_{PZ}$). This means that a DC in the middle of the prostate with e.g. 1000 correctly classified pixels has the same meaning/weight as a DC from a slice at the beginning or the end of the prostate stack with e.g. 10 correctly classified pixels. For instance, consider three slices: first slice contains in total 30 prostate pixels, second slice contains 1000 prostate pixels and third slice contains 30 prostate pixels. Next, suppose the proposed algorithm detects in the first slice 10 prostate pixels, in the second 950 and in the third slice 10. DC scores are then suppositionally 0.3, 0.95 and 0.3 and the overall mean DC score averages to 0.52, despite the fact that 970 out of 1020 prostate pixels are correctly. This example illustrates the DC establishment of one

Algorithm 2 Error Metric Calculation

Require: $\hat{\mathbf{Y}}_{2st}$

separate cases: $\hat{\mathbf{Y}}_{2st} \Rightarrow \sum_{q=1}^{Q} \mathbf{L}(i,j,l)$

split into 2 binary label stacks: $\sum_{q=1}^{Q} \mathbf{L}(i,j,l) \Rightarrow \sum_{q=1}^{Q} \mathbf{L}_{CG}(i,j,l) \bigcup \sum_{q=1}^{Q} \mathbf{L}_{PZ}(i,j,l)$

$cnt = 1$
for q=1 **to** q=Q **do**
  for l=1 **to** l=N **do**
    $\mathbf{E}_{CG}(cnt) \Leftarrow getAllErrorMetrics\left(\sum_{q=1}^{Q} \mathbf{L}_{qCG}(i,j,l)\right)$
    $\mathbf{E}_{PZ}(cnt) \Leftarrow getAllErrorMetrics\left(\sum_{q=1}^{Q} \mathbf{L}_{qPZ}(i,j,l)\right)$
    $cnt++$
  end for
end for
$\mathbf{Score}_{CG} = mean\,(\mathbf{E}_{CG})$
$\mathbf{Score}_{PZ} = mean\,(\mathbf{E}_{PZ})$

study comprised of three slices in order to point out the strict DC calculation.
Finally, a setup is described with 10 error scores. The rest of this chapter describes the calculations of the error metrics itself.

1. Dice coefficient The calculation of the DC is defined in equation 4.5 and can be understood as an area based segmentation metric.

$$\mathrm{DC} = \frac{2 * \mathrm{TP}}{(\mathrm{FP} + \mathrm{TP}) + (\mathrm{FN} + \mathrm{TP})} \tag{4.5}$$

The DC is the proportion of the mutual-overlap – the union of prediction and ground truth. A DC of 1 represents perfect agreement of the prediction slice and ground truth slice. In contrast to that, 0 indicates no overlap between the prediction and ground truth. DC score is in the following considered to be the most important error metric score.

2. Sensitivity is known as the fraction of positives that are correctly assigned. Therefore, sensitivity represents the ability to detect ground truth.

$$S_{ens} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{4.6}$$

Sensitivity is in literature also known as recall and is defined and illustrated in Equation 4.6. As FP and TN are not taking into account, setting for instance the whole binary label stack of the CG to 1, would result in a sensitivity of 1. For that reason specificity is also incorporated, which would be 0 in this example. Thus, a sensitivity of 1 represents

the best achievable score, but simultaneously specificity has also to be taken into account.

3. Specificity is defined as the ratio of TN to TN plus TP and is illustrated in Figure 4.7.

$$S_{peci} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{4.7}$$

Specificity is also known as the ability to detect negative results which means in this work the ability to detect background. Setting for instance the whole binary label stack of the central gland to 0 would result in a specificity of 1 and a sensitivity of 0. Therefore, as previously explained, sensitivity and specificity have to be considered combined.

4. Positive Predictive Value represents in this work how much of the predicted CG pixels are truly CG pixels when considering the binary label stack of the CG. Therefore the positive predictive value (PPV) is defined as follows:

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{4.8}$$

The PPV determines how much from the assigned pixels are truly correct classified. The optimal score would be 1.

5. Hausdorff Distance of Boundaries is defined as the furthest distance between two boundary sets $\mathbf{S}_1$ and $\mathbf{S}_2$. $\mathbf{S}_1$ and $\mathbf{S}_2$ are basically represented by the predicted slice and the ground truth slice. As the Hausdorff distance is the maximum distance between the predicted binary slice to the nearest point in the binary ground truth slice, a lower score is better. The calculation of the HdB is shown in Equation 4.9, whereby sup stands for supremum and P_1 is a point of $\mathbf{S}_1$ and accordingly is P_2 a point of $\mathbf{S}_2$. In this work the HdB is measured in milimeters. Different to literature, this work presents the 100th percentile of the HdB and not the 95th percentile.

$$dist_H(\mathbf{S}_1, \mathbf{S}_2) = \max \left\{ \sup_{P_1 \in S_1} dist(P_1, S_2), \sup_{P_2 \in S_2} dist(P_2, S_1), \right\} \tag{4.9}$$

For model evaluation purposes, statistical analysis utilizing cross-validation is carried out. Therefore, the fraction of training cases is defined to 83%, respectively the fraction of prediction cases is determined to 17%. For instance, a cross-validation on 25 studies leads to a training set of 21 studies and a prediction set of 4 studies. To receive robust results, 10 iterations are performed. In each iteration, studies for training and prediction are picked randomly. Hence, during each iteration a new FFANN model is trained. Error metrics, which are shown later on in Tables, are again one more time averaged over the iterations to receive 5 overall error metric scores for the CG and 5 overall error metric scores for the PZ. All calculations are performed on a machine comprised of an i7 core processor and 16GB of RAM.

Chapter 5

Results

This chapter contain the results of this work structured and presented in three subsections. The first Section 5.1 describes the results of different experiments, which were carried out to determine the optimal parameters of the proposed FFANN model. Error metrics, previously described in Section 4.7 are calculated to enable a comparison of different setups. Subsequently, Section 5.2 is about the evaluation of the proposed algorithm applied on the prostate gland as well as on the prostate zones. In both cases the beforehand determined optimal parameters are utilized. Finally, subsection 5.3 shows an evaluation of the algorithm based on 100 *MD Anderson Cancer Center* studies as well as an evaluation of the proposed volume estimation (see Section 4.6).

5.1 Model Parameter Estimation

Optimal neural network model parameters are understood as parameters which achieve the best error metrics scores in reasonable time. Thus, these parameters provide the best pixel classification result and accordingly the best segmentation result. Because of the involved generalization which is caused by the utilized neural network model, the DC is considered to be the most important error metric. Furthermore, results state that the sensitivity correlates with the DC and the average specificity achieves steadily scores of >0.8. Consequently, the DC is used for illustration purposes further on in this result chapter in order to present results clearly. This subchapter is about establishing the optimal neural network parameters. Hence, the following parameters are considered to be important to be determined:

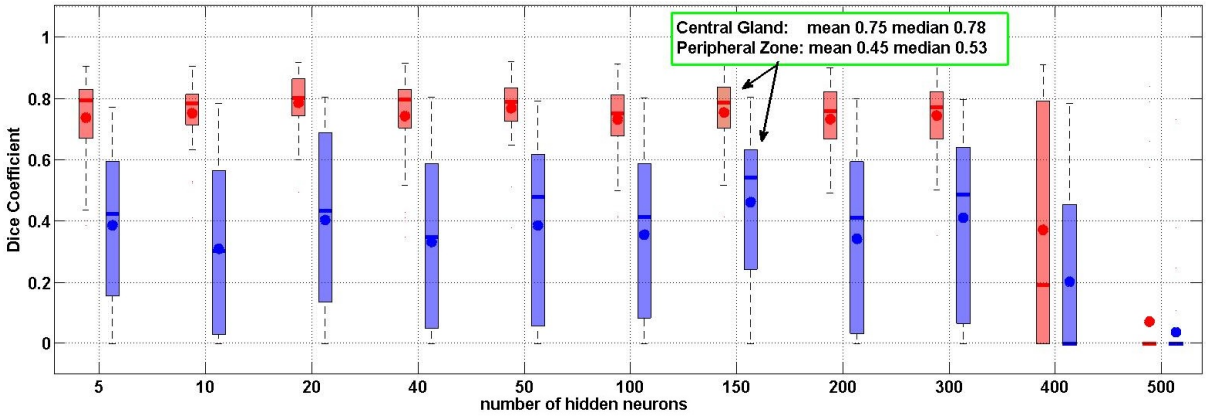
- optimal number of hidden neurons and hidden layers (see 5.1.1)
- optimal number of training studies - best training performance (BTP) (see 5.1.2)
- optimal batchsize and epochs (see 5.1.3)
- optimal weight weight initialization method (WIM) (see 5.1.4)
- optimal neighborhood size (NHS) (see 5.1.5)

Each item of the above illustrated list forms in the following chapter its own subsection. Each subsection explains the corresponding experiment and presents results in order to find incremental the optimal neural network parameters.

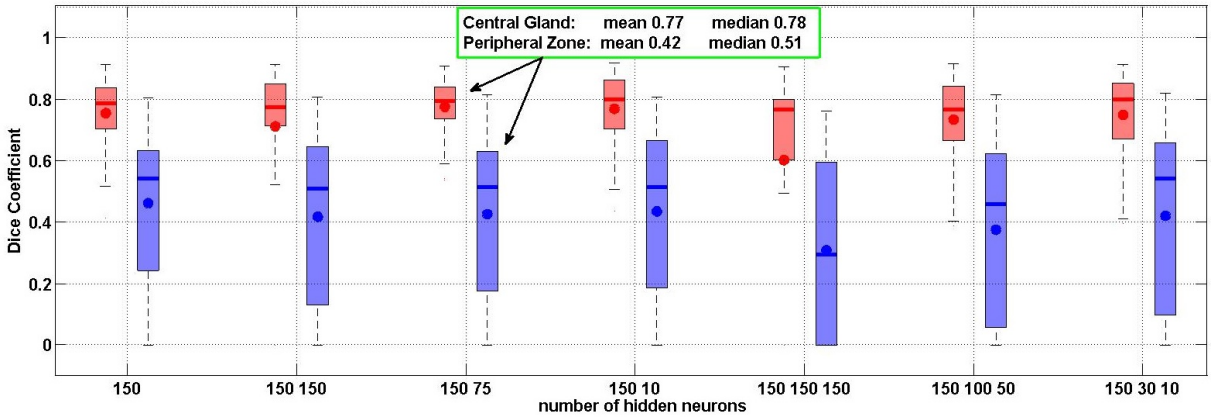
5.1.1 Neural Network Structure

As already described in section 4.4.1, a FFANN can basically consist of three different network shapes (straight, linear and exponential). The experiment in this section focuses on finding which of these shapes fits best for this purpose. The proposed experiment is performed on 25 3-Tesla studies using the fully-automated *A2-Mode*. For each neural network setup a cross-validation is carried out. One cross-validation consists of 10 cross-validation steps. In each step 21 studies are randomly picked for training. Respectively, the remaining 4 studies are used for prediction. In addition, in each step error metrics are calculated in accordance to algorithm 2, except of the two last code lines. The two last two code lines are just responsible to calculate the mean and median in order to receive 10 error metrics for all prediction studies.
Considering now only the DC score and 10 cross-validation steps yield in total to 40 DC scores for the CG and 40 DC scores for the PZ. Thus, one FFANN setup provides 80 DC scores. All DC scores of several FFANN setups are visualized in boxplots in Figure 5.1. In Figure 5.1 each red box represent 40 CG DC scores of one FFANN setup. Respectively, each blue box indicates 40 DC scores of the PZ of one FFANN setup, whereby, the edges of the boxes present the 25th and the 75th percentiles. Furthermore, the mean of 40 DC scores is characterized by a dot and the median by a slash. Whiskers of each box extend to the most extreme data points. The behaviour of the first FFANN is considered to be similar to the second FFANN. Hence, it is believed that finding the optimal parameters of the first FFANN corresponds to the optimal parameters of the second FFANN. In addition, batchsize is set default to 20 utilizing 2 epochs.

Figure 5.1(a) shows the results of 11 different FFANN setups, whereby each is comprised of one hidden layer which is in turn made up of an increasing number of hidden neurons. The 11 setup configurations are as follows: 5, 10, 20, 40, 50, 100, 150, 200, 300, 400, 500. For instance, 100 corresponds to 100 hidden neurons utilizing one hidden layer and yield to neural network shape of $28\times100\times3$, whereupon 28 is the number of features and 3 the number of classes. The results show that the DC scores for the CG fluctuate steadily around 0.78 using 5 to 300 hidden neurons. The same behaviour is captured for the PZ. The DC score for the PZ shakily stays around 0.4. The best setup is indicated with arrows and the corresponding mean and median DC scores are illustrated in Figure 5.1 in green boxes. Utilizing 400 and 500 hidden neurons results in lower to zero DC scores for the CG as well as for the PZ. Because there is firstly no specific optimal setup in Figure 5.1(a) identifiable and secondly the behaviour of different shapes using multiple hidden layers should be captured as well, four depth-experiments are carried out next.
Figure 5.1(b) illustrates one out of four depth-experiments from base 150 hidden neurons. This means that 150 hidden neurons form the basis of a straight- linear-and exponentially shaped FFANN model. The remaining three depth-experiments from bases 100, 200 and 400 are presented in the Appendix B.2. The x-values 150 150 and 150 150 150 in Figure 5.1(b) indicate a straight shaped neural network comprised of two and three hidden layers. Next, boxes located at x-values 150 75 and 150 100 50 characterize a linear shaped FFANN with two and three hidden layers. Lastly, exponentially shaped FFANN models with two and three hidden layers are represented in Figure 5.1(b) by boxes around the x-values



(a) A2-Mode with increasing number of hidden neurons in one hidden layer



(b) A2-Mode with various number of hidden neurons and hidden layers from base 150 hidden neurons

Figure 5.1: DC scores in boxplots of cross-validation using various numbers of hidden neurons and hidden layers. The x-axis shows the number of hidden neurons and hidden layers. For instance, 150 150 stands for 150 hidden neurons in the first hidden layer and 150 hidden neurons in the second hidden layer. The y-axis represents the DC score. Red boxes indicate scores for the CG and blue boxes scores for the PZ. The edges of the boxes present the 25th and 75th percentiles of each score. Slashes within the box represent the median and points represent the mean. Whiskers extend to the most extreme data points.

150 10 and 150 30 10. Considering all four depth-experiments, straight shaped networks provide the lowest DC scores. Furthermore, increasing the depth (adding hidden layers) as well as increasing the number of neurons results in decreased DC scores. Depth-experiments illustrated in the Appendix B.2 show their best setup, which is indicated through green boxes, when utilizing an exponentially shaped FFANN. Regarding Figure 5.1(b) the best two setups are 150 75 and 150 10, whereby 150 75 is considered as best setup of all experiments. Therefore, the linear shaped setup 150 75 provides to the most accurate results utilizing the *A2-Mode*. Finally, the optimal number of hidden neurons and hidden layers for further investigations are 28×150×75×3 for the first FFANN model and 28×150×75×3 for the second FFANN model.

5.1.2 Learning Curve Experiment

This experiment focuses on capturing the learning performance of the proposed algorithm on 25 3-Tesla studies. Thus, the algorithm is set to the *A2-Mode* and is trained several times, whereby each time an increasing number of training studies is utilized. The goal of this experiment is to discover,if the algorithm performs more accurately when the FFANN model is trained with more studies. The resulting learning curve is illustrated in Figure 5.2. The x-axis represents the increasing number of utilized training studies. As the fraction of training is fixed to 0.83, the first x-value (5) indicate that the algorithm is trained on 4 studies and tested on 1 study. Cross-validation is carried out, which yields to 10 iterations in total for each setup. The training as well as the prediction studies are randomly picked. The last x-value (25) results in a training set of 21 studies and a prediction set of 4 studies. The number of hidden neurons and hidden layer is set to their optimal value of 28×150×75×3 for both FFANN models. Batchsize is set default to 20 utilizing 2 epochs.

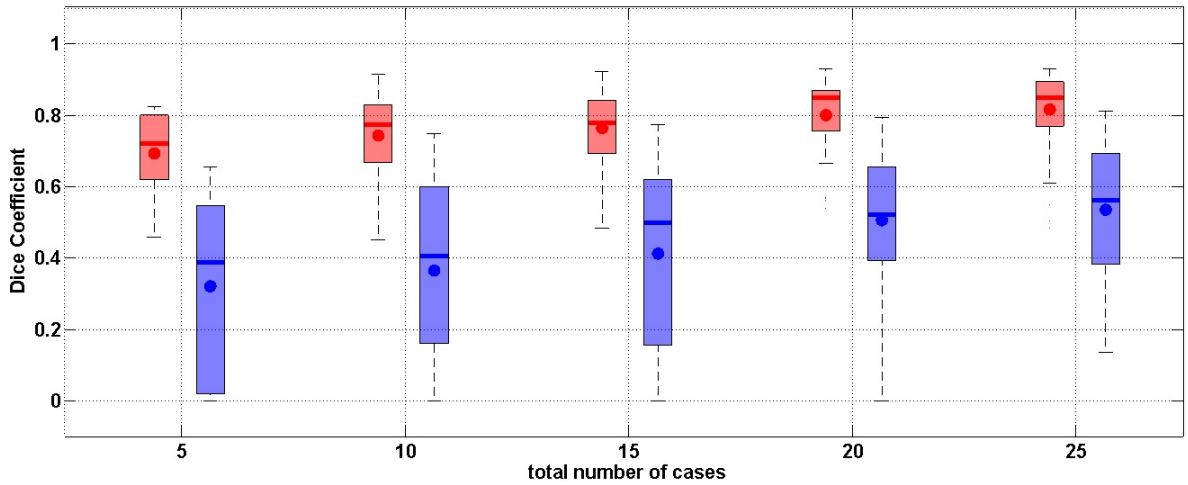


Figure 5.2: DC scores of cross-validation utilizing the A2 Mode in dependency of a increasing number of training studies. The fraction of training studies is set to 0.83. The x-axis represents the number of utilized studies from 5 to 25. The y-axis presents DC score. Red boxes indicate scores for the CG and blue boxes indicate scores for the PZ. The edges of the box present the 25th and 75th percentiles of each score. Slashes within the box represent the median and dots represent the mean. Whiskers extend to the most extreme data.

The y-axis in Figure 5.2 indicates DC scores. Red boxes indicate DC scores for the CG and blue boxes characterize DC scores for the PZ. Edges outline the 25th and the 75th percentile. Red dots symbolize mean DC scores for the CG and the red slashes mark the median DC scores for the CG. Analogous, DC for the PZ are represented with blue dots and blue slashes. Whiskers of each box extend to the most extreme data points.

Referring to Figure 5.2, one of the most obvious consequences of utilizing an increasing number of training studies is that the the proposed algorithm performs more precisely. The most precise results are captured using 21 studies for training and 4 studies for

prediction. Hence, the mean DC score for the CG utilizing 25 studies in total is 0.81. Accordingly, the median DC score is 0.84 for the CG. In addition, the mean and median DC scores for the PZ using 25 studies are 0.47 and 0.52.
Derived from Figure 5.2 the learning behaviour of the proposed algorithm has not reached its saturation. So far, the optimal number of training studies respectively, the best training performance (BTP) is reached utilizing 21 training studies. Moreover, the learning characteristics is captured again later on in section 5.3 on a larger dataset with 100 studies.

5.1.3 Batchsize and Epochs Experiment

The batchsize parameter as well as the epochs parameter are defined as the tradeoff between mean squared error and computational costs. Because weights of the FFANN are updated after each batch, a lower batchsize is associated with high computational costs and a higher batchsize corresponds to low computational costs. The goal of the batchsize experiment is to establish, if the proposed algorithm performs more precisely utilizing a decreased batchsize. An epoch is defined as one sweep through of the training set throughout the FFANN model. Hence, the two FFANN models should perform more precisely utilizing an increasing number of epochs. The aim is to discover the behaviour of FFANN models, when each of the models "sees" the training set multiple times.

Figure 5.3 represents the result of the batchsize experiment based on 25 3-Tesla studies. Therefore, cross-validation utilizing the *A2-Mode* is carried out. In total five different batchsizes (5000, 1000, 500, 100, 10) were tested. In Figure 5.3 the vertical axis represents five different batchsizes in decreasing order. The horizontal axis shows the DC. Red boxes in Figure 5.3 reveal scores for the CG and respectively, blue boxes specify scores for the PZ. Edges of the boxes present the 25th and 75th percentiles of each score. Slashes within the box indicate the median and dots characterize the mean. Whiskers extend to the most extreme data points not considering outliers.
The declared batchsizes (5000, 1000, 500, 100, 10) are maximum batchsize values. The actual batchsize size is less or equal the maximum batchsize because the number of pixels in the training set divided by the maximum batchsize must be an integer value. Starting from the maximum batchsize and decrement each time by one, the actual batchsize is the first number, which divides the total number of pixels in the training set without remainder. Figure 5.3 confirms the assumption that the algorithm is performing more precisely using a decreased batchsize. Lowering the batchsize can be seen as increasing the computational time. A batchsize of one would result in a weight adaption after each feature vector, which is called online-learning. Figure 5.3 leads to the presumption that the DC scores for both prostate zones would probable increase further, if the batchsize would be set to a value smaller than 10. But this experiment would disrupt the computation time of the utilized resources, which are a notebook comprised of an i7 core processor and 16GB of RAM.

The next experiment is to determine how many epochs provide the most precise epochs setting. Hence 1, 2, 3, and 4 epochs are tested and presented in Figure 5.4. Again cross-validation utilizing 10 iteration is carried out. In each iteration 21 studies are randomly

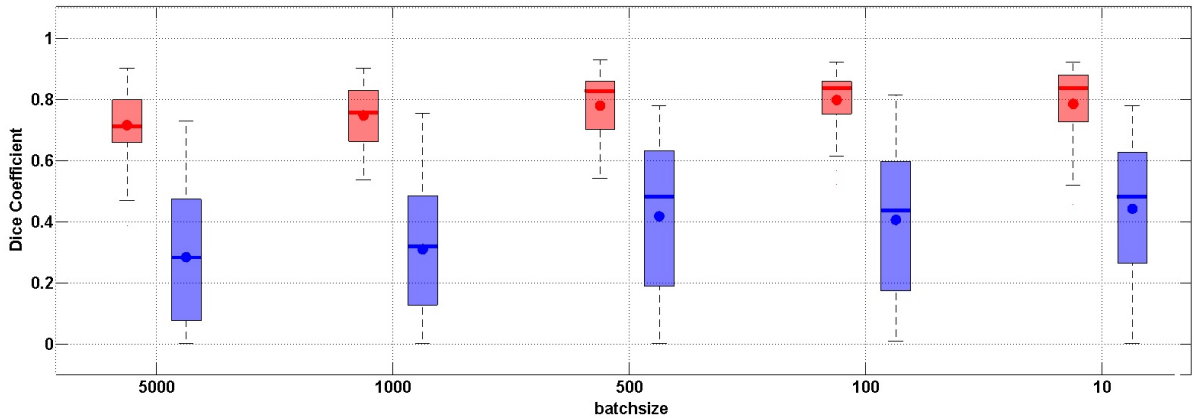


Figure 5.3: DC scores of cross-validation utilizing the *A2-Mode* in dependency of different batchsizes. The fraction of training studies is set to 0.83 and 25 3-Tesla studies have been used in total. The x-axis represents the decreasing batchsize from 5000 to 10. The y-axis presents the DC. Red boxes illustrate scores for the CG and blue boxes symbolize scores for the PZ. Edges of the boxes present the 25th and 75th percentiles of each score. Slashes within the box represent the median and points represent the mean. Whiskers extend to the most extreme data points.

picked for training and the remaining four studies are selected for prediction. The cross-validation results using 1, 2, 3 and 4 epochs are shown in boxplots in Figure 5.4. The x-axis indicates the number of epochs and the y-axis illustrates the DC. Red boxes identify the DC scores for the CG. According to this, blue boxes highlight the DC scores for the PZ. Edges of the boxes depict the 25th and 75th percentiles of each score. Slashes within the box represent the median and dots represent the mean. Whiskers extend to the most extreme data points.

Figure 5.4 points out that the overall algorithm preciseness shrinks using a increasing number of epochs despite the fact that the mean squared error is dropping in each epoch. Certainly, the reason for that phenomena could have its origin in the relatively small batchsize, which yields in combination with multiple epochs as a consequence to overfitting. Utilizing 21 randomly picked training studies results approximately in a training set (design matrix) $\mathbf{X}$ of $3.000.000 \times number\ of\ features$. Thus, a batchsize of 10 is relatively small and leads probably to overfitting when using multiple epochs. Increasing the batchsize and number of epochs simultaneously could solve the problem of overfitting. In conclusion, referring to Figure 5.4 using a batchsize of 10 yields to an optimal epochs number of 1 or 2.

5.1.4 Weight Initialisation Experiment

This section is concerned with establishing the best weight initialization method in order to avoid local minima on the error surface during the backpropagation algorithm. Therefore, three different weight initialization method (WIM)s are carried out. The first method is called zero weight initialization and as the name suggests all weights are strictly initialized with zeros. The second weight initialization method is named random weight

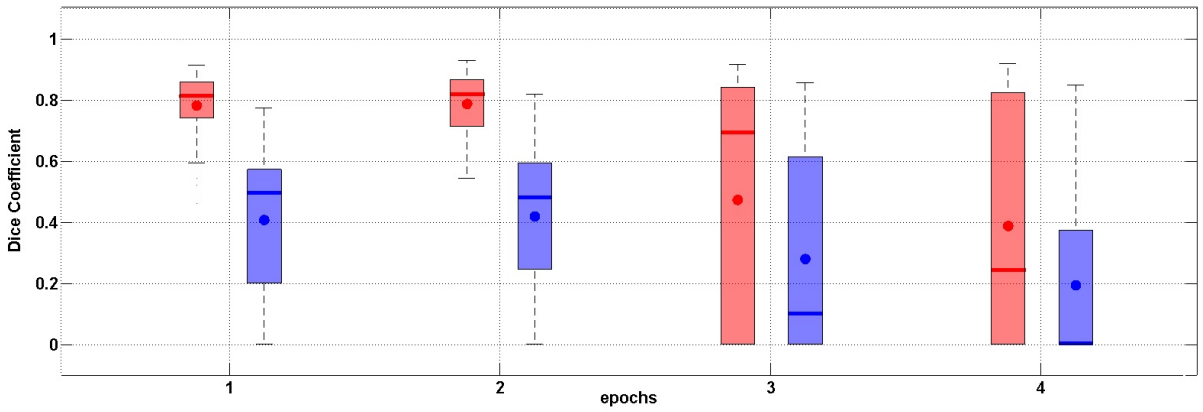


Figure 5.4: DC scores of cross-validation utilizing the *A2 Mode* in dependency of different epochs. The fraction of training studies is set to 0.83 and 25 3-Tesla studies are used in total. The x-axis represents the increasing number of epochs from 1 to 4. The y-axis presents the DC. Red boxes indicate scores for the CG and blue boxes represent scores for the PZ. Edges of the boxes present the 25th and 75th percentiles of each score. Slashes within the box represent the median and dots represent the mean. Whiskers extend to the most extreme data points.

initialization. Y. Lecun et al [61] recommended to draw the weights from a uniform random distribution limited by the upper and lower bounds $\pm\sqrt{6/(f_{in}+f_{out})}$. In particular, considering the weights between the input layer and the first hidden layer, f_{in} is the number of input neurons and f_{out} is represented by the number of hidden neurons of the first hidden layer. Hence, using the optimal FFANN structure $28 \times 150 \times \times 75 \times 3$ results in $f_{in} = 28$ and $f_{out} = 150$.

The third weight initialization method is truly the most complex computational one and is entitled as weight initialization utilizing DBNs. Thus, a DBN is firstly trained based on the training set $\mathbf{X}$ in an unsupervised manner. Subsequently, the learned weights of the DBN are used as initial weights of a FFANN model. Lastly, the FFANN is trained or in other words fine-tuned using backpropagation. However, for all three WIMs biases are constantly initialized with zeros.

Figure 5.5 illustrates DC scores utilizing three different WIMs. The DC scores of each setup are represented in boxplots. The higher the DC scores, the more precisely the algorithm performs. On the one hand red boxes reveal DC scores for the CG and on the other hand blue boxes show DC scores for the PZ. Edges of the boxes present the 25th and 75th percentiles of each score. Slashes within the box represent the median and points represent the mean. Whiskers extend to the most extreme data points. Figure 5.5 clearly points out that zero weight initialization provides the lowest DC scores and consequently, does not provide practical relevant results. In contrast, the two remaining WIMs show almost equal DC scores. It was believed that learning the weights in a pre-training step utilizing DBN leads to more accurate results than results achieved with random weight initialization. Nevertheless, Figure 5.5 provides the evidence that random weight initialization and weight initializing using DBNs leads in the proposed algorithm

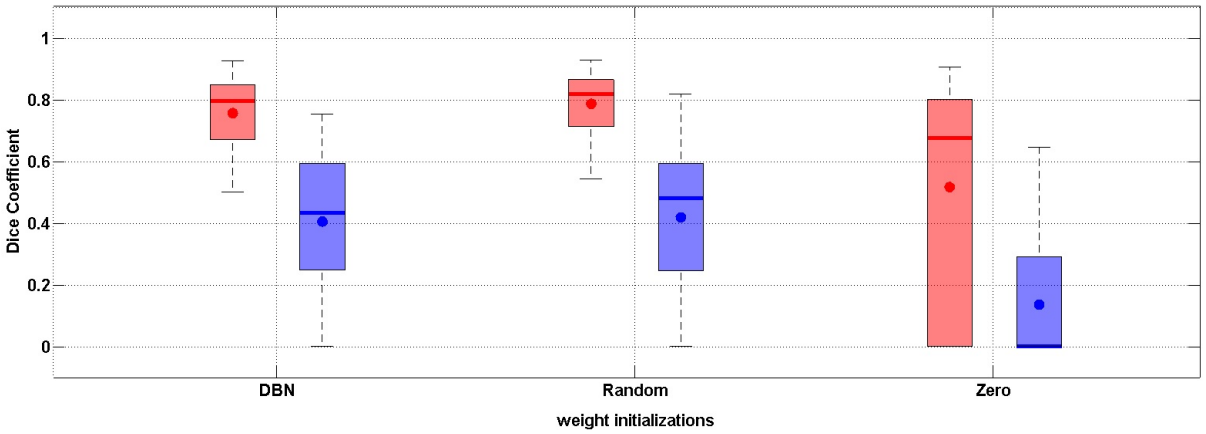


Figure 5.5: DC scores utilizing cross-validation and the *A2-Mode* in dependency of three different weight initialization methods. The fraction of training studies is set to 0.83 and 25 3-Tesla studies are used in total. The x-axis represents three different WIM, namely initialization with deep belief networks, random weight initialization and zero weight initialization. The y-axis presents the DC for the CG in red and for PZ in blue. Edges of the box present the 25th and 75th percentiles of each score. Slashes within the box represent the median and dots represent the mean. Whiskers extend to the most extreme data points.

to correlating results. However, considering the time consuming pre-training step utilizing DBNs, random weight initialization is used superior for further investigations.

5.1.5 Neighborhood Size Experiment

This section is about discovering the optimal neighborhood size (NHS) for the feature extraction process. As described in section 4.3, neighborhood gray values from the current pixel of interest are taken into account as features. Thus, different NHSs yield to distinct feature-vector sizes and consequently to a diverse FFANN classification behaviour. Furthermore, filtered gray value features are part of the feature extraction for which the NHS plays a decisive role in the filter process and effects as a consequence the classification outcome, too.
In total four NHSs (1, 3, 5, 7) are experimentally tested. A NHS of 1 leads in a total feature vector length of 28. Hence, for each pixel 28 features are extracted. The feature vector has 52 dimensions using a NHS of 3 and 152 dimensions using a NHS of 5. Moreover, a NHS of 7 yields to a total feature vector length of 370. Increasing the NHS results in an increased computational time. Cross-validation on 25 3-Tesla studies utilizing the *A2-Mode* and 10 iterations is carried out for each NHS experiment. The fraction of training is steadily set to 0.83. Thus, 21 studies are used for training and 4 for prediction. The result of the NHS experiment is illustrated in boxplots in Figure 5.6. The horizontal axis shows the increasing NHSs and the vertical axis presents the DC scores for the CG in red boxes and for the PZ in blue. Edges of the boxes present the 25th and 75th percentiles of each score. Slashes within the box characterize the median and points within each box indicates the mean. Whiskers extend to the most extreme data points.

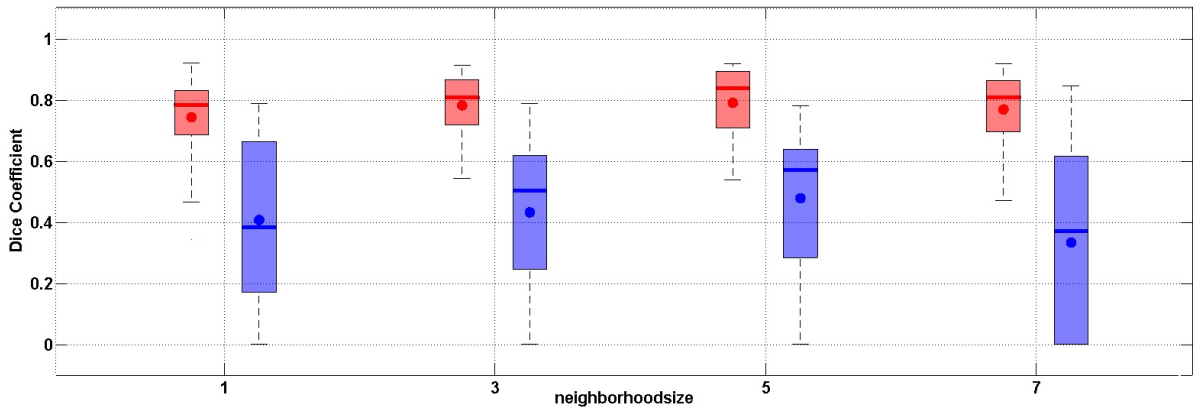


Figure 5.6: DC scores utilizing cross-validation and the *A2-Mode* in dependency of four different NHSs. The NHS plays a key role in the feature extraction process and defines mainly the feature vector dimension. The fraction of training studies is set to 0.83 and 25 3-Tesla studies are used in total. The x-axis represents various NHSs. The y-axis presents the DC. Red boxes highlight scores for the CG and blue boxes mark scores for the PZ. Edges of the boxes present the 25th and 75th percentiles of each score. Slashes within the box outline the median and dots present the mean. Whiskers extend to the most extreme data points.

Figure 5.6 reveals raising DC scores from NHS 1 to 5. Afterwards, the DC scores drop down at NHS 7. Consequently, the most precise algorithm performance emerges at NHS 5. Accordingly, the mean DC scores are 0.78 for the CG and 0.47 for the PZ. In addition, the median DC scores are 0.83 for the CG and 0.56 for the PZ.

The optimal FFANN parameters are presented in Table 5.1, whereby, each parameter is equal for both layers. The best FFANN shape is $152 \times 150 \times 75 \times 3$ for both layers. The optimal batchsize and epochs are considered to be 10 and 1. In addition, the optimal NHS is deemed to be 5. The best WIM is considered to be the random weight initialization. Lastly, the best BTP revealed using 21 training studies. Finally, all optimal FFANN model parameters are established and can be utilized for further investigations on different datasets.

Table 5.1: Optimal FFANN parameters

	FFANN shape	**batchsize**	**epochs**	**NHS**	**WIM**	**BTP**
localization layer	$152 \times 150 \times 75 \times 3$	10	1	5	random	21
labeling layer	$152 \times 150 \times 75 \times 3$	10	1	5	random	21

Optimal FFANN parameters for the proposed algorithm; FFANN stands for feed forward artificial neural network; NHS stands for neighborhoodsize; WIM stands for weigh initialization method and BTP stands for best training performance

5.2 Comparison between Prostate Zone and Prostate Gland Segmentation

This section provides a comparison between prostate zone segmentation and prostate gland segmentation. The prostate gland is comprised of CG and PZ and is subsequently treated as one region. Hence, Subsection 5.2.1 presents error metric scores based on prostate gland segmentation on the 25 3-Tesla studies. Accordingly, outlines Subsection 5.2.2 the error metric scores for prostate zone segmentation based on 25 3-Tesla studies and 25 1.5-Tesla studies. Additionally, the volume estimation algorithm is evaluated on 25 3-Tesla studies. For the established results in this section are the previously determined optimal model parameters, which are shown in Table 5.1, utilized.

5.2.1 Automated Prostate Gland Segmentation

The error metric scores for prostate gland segmentation are illustrated in Figure 5.7 and in Table 5.2. A red box in Figure 5.7 reveals the score for the corresponding error metric. Edges of the boxes extend of the corresponding score to the 25th and 75th percentile. Slashes within the box symbolize the median and dots indicate the mean. Mean and median error metric scores are highlighted in Table 5.2. Whiskers in Figure 5.7 extend to the most extreme data points. As the minimum DC score is 0.715, the algorithm constantly achieves a DC score over 71.5% and half of the predicted prostate glands has a DC score between 0.825 and 0.898.

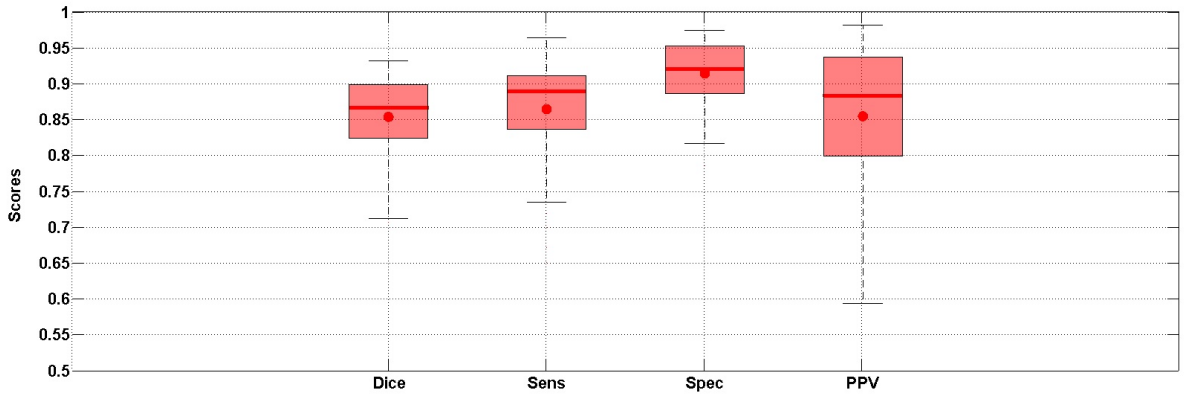


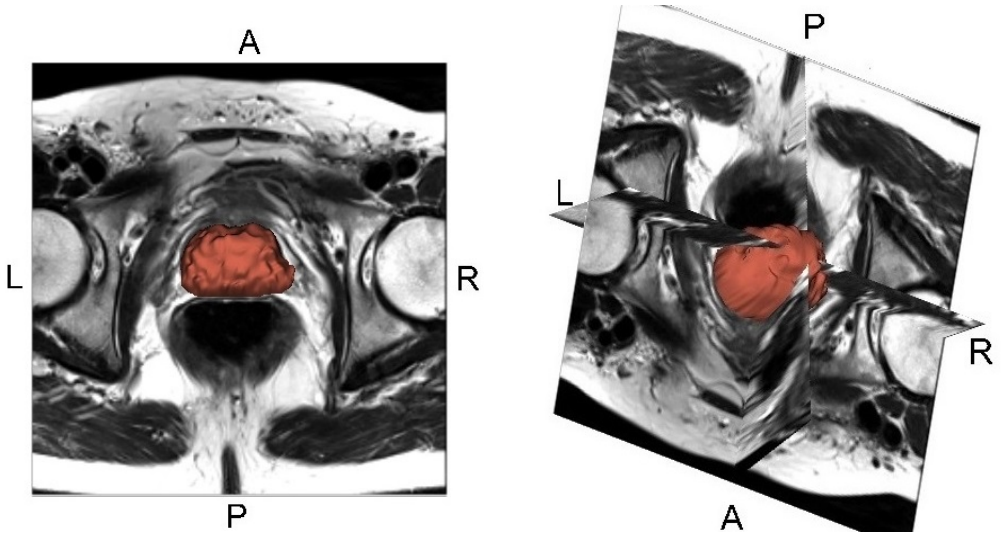
Figure 5.7: The scores of four error metrics in boxplots. The four metrics are: DC, sensitivity, specificity and PPV. Cross-validation on 25 3-Tesla studies utilizing the A2 Mode is carried out. The fraction of training studies is set to 0.83. Red boxes characterize scores for the prostate gland. Edges of the boxes present the 25th and 75th percentiles of each score. Slashes within the box represent the median and dots represent the mean. Whiskers extend to the most extreme data points.

As already mentioned at the beginning of the result chapter, the sensitivity highly correlates with the DC score. This characteristic is proofed in Table 5.2. Furthermore, an

average specificity of 0.919 certifies that 91,9% of all pixels are correctly classified as background. The mean PPV is 0.854. Thus, 85.4% of the as prostate gland assigned pixels are truly prostate gland pixels. The mean HdB amounts to 8.50 millimeters.

Table 5.2: Error metric scores for automated prostate gland segmentation on 25 3-Tesla studies

	DC	**Sens**	**Spec**	**PPV**	**HdB**
mean	0.853	0.863	0.913	0.854	8.50 *mm*
median	0.865	0.888	0.919	0.882	7.81 *mm*

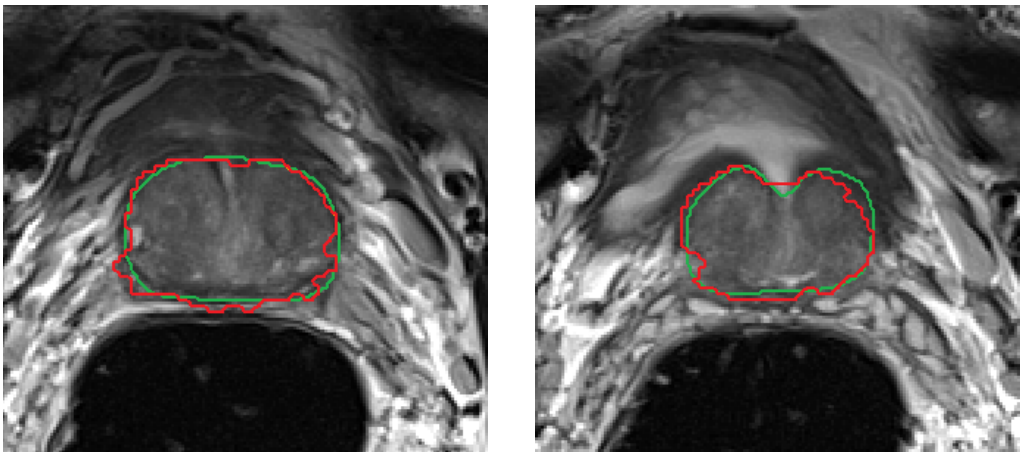


(a) (b)

Figure 5.8: Figures (a) and (b) show segmentation results of the prostate gland of study *Prostate3T-01-0001* visualized in the software application *3D Slicer*. The 3D prostate gland is shown in red. The capitalised letters indicate the coordinate system: anterior (A), posterior (P), left (L) and right (R).

Figure 5.8 represents the prediction result of study *Prostate3T-01-0001*. The algorithm is trained on 21 studies and study *Prostate3T-01-0001* is one of the four prediction studies. The visualization is done in *3D Slicer*. Figure 5.8(a) shows the transversal cut MR image overlaid by the three dimensional segmentation result in red. Figure 5.8(b) illustrates axial, sagittal and coronal MR images planes superimposed by the three dimensional segmentation result of the prostate gland in red. The capitalised letters in figure 5.8 represent direction-abbreviations of the coordinate system: anterior (A), posterior (P), left (L) and right (R).

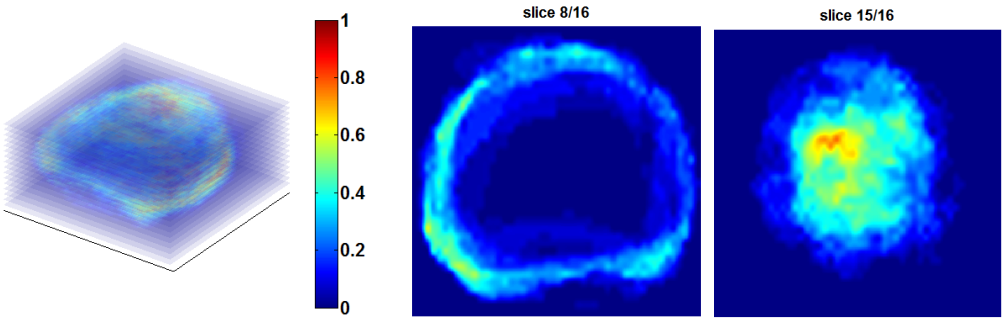
In order to present a segmentation results in detail, Figures 5.9(a) and 5.9(b) show the prostate gland segmentation results of two slices. The boundary of the predicted prostate



(a) (b)

Figure 5.9: Figures (a) and (b) show transversal cuts of the segmented prostate gland of study *Prostate3T-01-0001*. The green boundary characterizes ground truth and the red boundary indicates the segmentation result.

gland is shown in red, whereas the green boundary indicates the ground truth. Because of not utilizing an appearanced based segmentation method and not adding a high level post-processing step, the shape of the predicted prostate gland looks squiggled in comparison to the ground truth prostate gland. However, results are competitive compared to G.Vincent et al [27] and M.Yang et al [33], which achieve mean dice scores of 0.88 and 0.93.



(a) (b) (c)

Figure 5.10: A set of three subfigures: (a) represents the three dimensional averaged prediction error of the prostate gland; (b) shows the eighth slice of the averaged prediction error; (c) illustrates the fifteenth slice of the averaged prediction error

Figure 5.10 represents plots of the averaged prediction error of the prostate gland. Utilizing 10 cross-validation iterations and a prediction fraction of 0.17 leads to 40 predicted studies in total. The false positive (FP) and false negative (FN) prediction areas of these 40 studies are averaged and shown in Figure 5.10(a). In addition, Figure 5.10(b) and Figure 5.10(c) illustrate the eighth and the fifteenth slice of the averaged prediction error. It is important to note that the algorithm is performing well around the prostate center

with error rates close to zero (dark blue). This behaviour is the result of utilizing distance features with respect to the center. Approaching the prostate gland border yields to the fact that the algorithm starts performing poorly and produces an average error around 0.45 (see Figure (c)). The highest error rates with 0.76 occur at the beginning and at the end of the averaged prediction error stack. There is a good match of an plot provided by G.Vincent et al in [27] and Figure 5.10. The illustrated map of the mean distance error plot in [27] reveals the same characteristic as shown in Figure 5.10.

5.2.2 Automated Prostate Zone Segmentation

This subsection describes error metrics and segmentation results of automated prostate zone segmentation utilizing the optimal model parameters from Section 5.1. In comparison to the previous Subsection 5.2.1 the two prostate zones namely CG and PZ are now separately taken into account. As may be seen below, Figure 5.11 illustrates the error metric scores in boxplots of cross-validation on 25 3-Tesla studies. The training fraction is set to 0.83 respectively, the prediction fraction is 0.17. Hence, red boxes mark DC scores for the CG and blue boxes symbolize scores of the PZ. Utilizing 10 cross-validation iterations yield to 40 predicted studies in total. For each region in each study are error metrics calculated. Thus, each error metric comprises 40 scores, which are accordingly represented in boxes. Edges of the boxes present the 25th and 75th percentiles of each score. Slashes within the box illustrate the median and dots indicate the mean. In addition, extend whiskers to the most extreme data points.

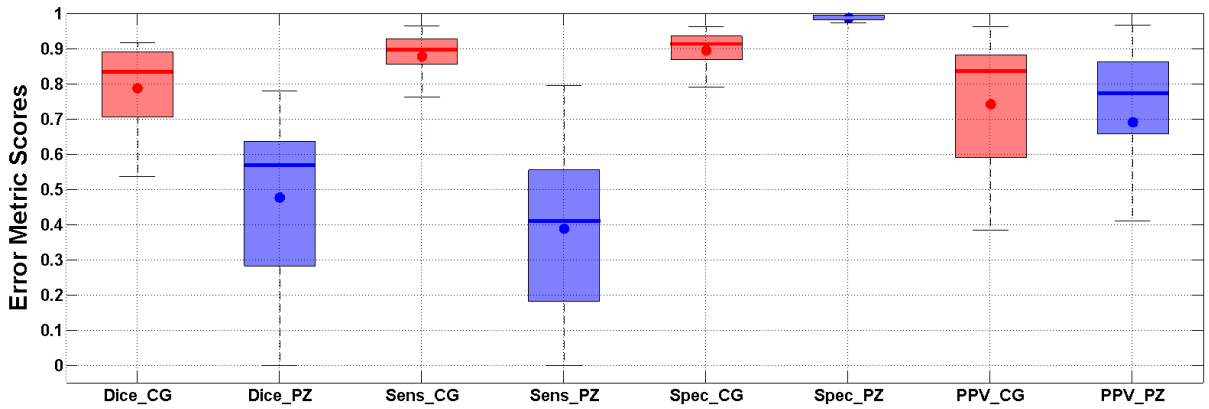


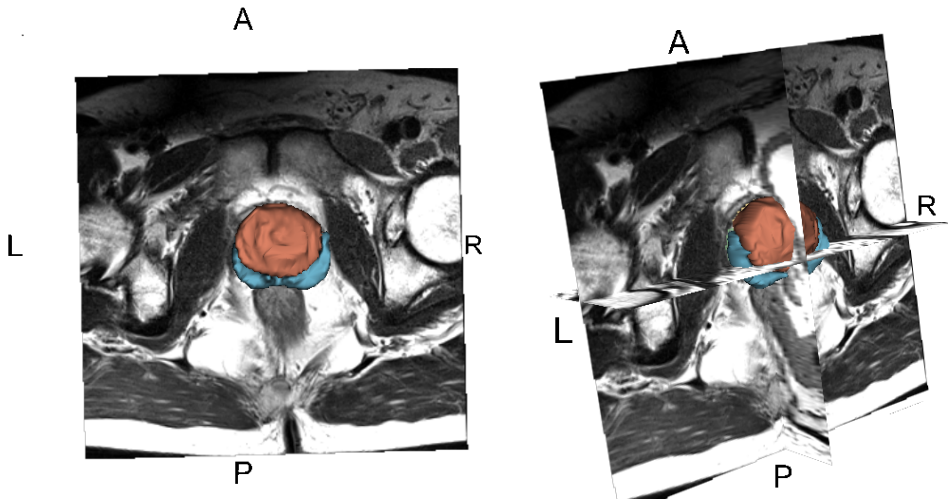
Figure 5.11: The scores of four error metrics in boxplots of prostate zone segmentation. The four metrics are: DC, sensitivity (Sens), specificity (Spec) and PPV. Each error metric is provided twice - once for the CG in red and once for the PZ in blue. Cross-validation on 25 3-Tesla studies utilizing the *A2-Mode* is carried out. The fraction of training studies is set to 0.83. Edges of the box present the 25th and 75th percentiles of each score. Slashes within the box highlight the median and dots indicate the mean. Whiskers extend to the most extreme data points.

Figure 5.11 reveals, that whiskers of the blue DC box and the blue DC sensitivity box extend to zero. This means that the algorithm predicted in some studies no PZ,

despite the fact that there is truly a PZ. In total 4 out of 40 times was no PZ predicted. Furthermore, the mean DC scores and mean sensitivity scores for the PZ are approximately 30% (0.47, 0.38) lower than mean DC scores and mean sensitivity scores for the CG (0.78, 0.87). This characteristic is caused mainly due the variable shape of the PZ, which is because of the generalization property of the FFANN model not captured. This behaviour results in stiffness when it is necessary to predict an highly abnormal or in other words a variable PZ. Another reason for lower DC scores for the PZ could be associated with the different gray value distribution of the PZ over all studies or the fact that distance features are extracted with respect to the prostate's mass point. Summarized, the extracted features describe the CG well but fail partly for the PZ.

Table 5.3: Error metric scores for automated prostate zone segmentation on 25 3-Tesla studies

	DC CG	**DC PZ**	**Sens CG**	**Sens PZ**	**Spec CG**
mean	0.789	0.476	0.877	0.387	0.894
median	0.833	0.569	0.897	0.409	0.913
	Spec PZ	**PPV CG**	**PPV PZ**	**HdB CG**	**HdB PZ**
mean	0.986	0.741	0.690	8.25 *mm*	15.46 *mm*
median	0.986	0.836	0.773	8.13 *mm*	13.68 *mm*

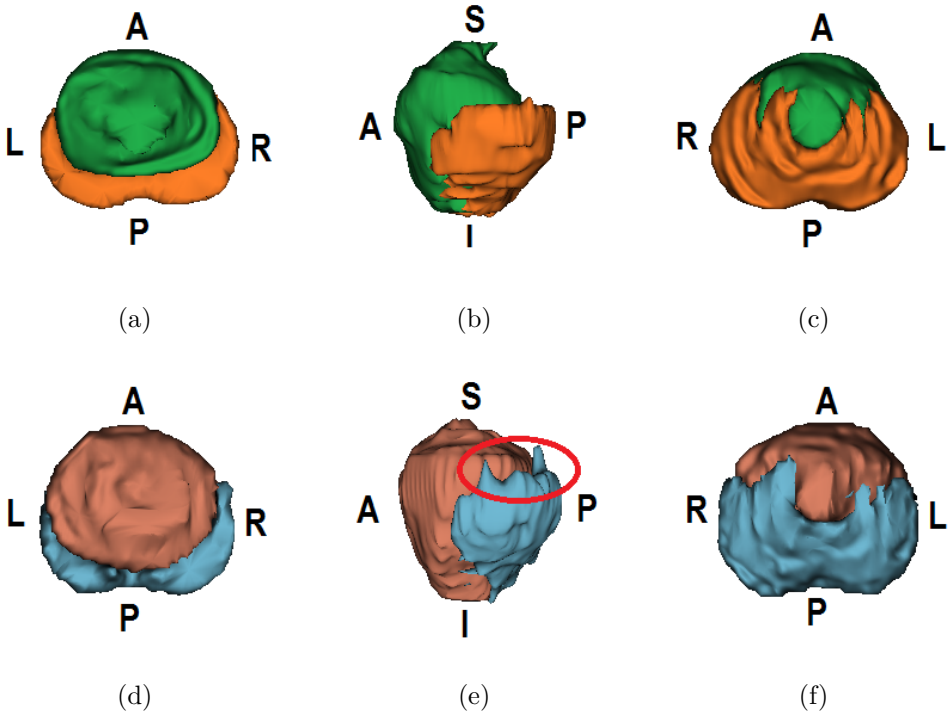


(a) (b)

Figure 5.12: Figures (a) and (b) illustrate the segmentation result of the prostate zones visualized in the software application *3D Slicer*. The 3D CG is shown in red and the PZ is represented in blue. The capitalised letters represent the coordinate system: anterior (A), posterior (P), left (L) and right (R).

All mean and median error metric scores for automated prostate zone segmentation are shown in Table 5.3. It has been found that scores for the CG correlate with scores for prostate gland segmentation. The mean values in Table 5.3 correspond to dots in Figure 5.11 and accordingly, correspond median values in Table 5.3 to slashes in Figure 5.11. There is no corresponding comparably literature for fully-automated prostate zone segmentation. Hence, table 5.3 presents the first fully automated prostate zone segmentation results on open access data.

Figure 5.12 illustrates the segmentation result of study *Prostate3T-01-0005*. The algorithm is trained on 21 studies and study *Prostate3T-01-0005* is one out of four prediction studies. The visualization is done via the software application *3D Slicer*. Figure 5.12(a) shows an adjacent axial cross-section MR image overlaid by the CG segmentation result in red and the PZ segmentation result in blue. Figure 5.12(b) illustrates axial, sagittal and coronal MR image planes combined with the prostate zone segmentation result. The capitalised letters in Figure 5.12 represent direction-abbreviations of the coordinate system: anterior (A), posterior (P), left (L) and right (R).



(a) (b) (c)

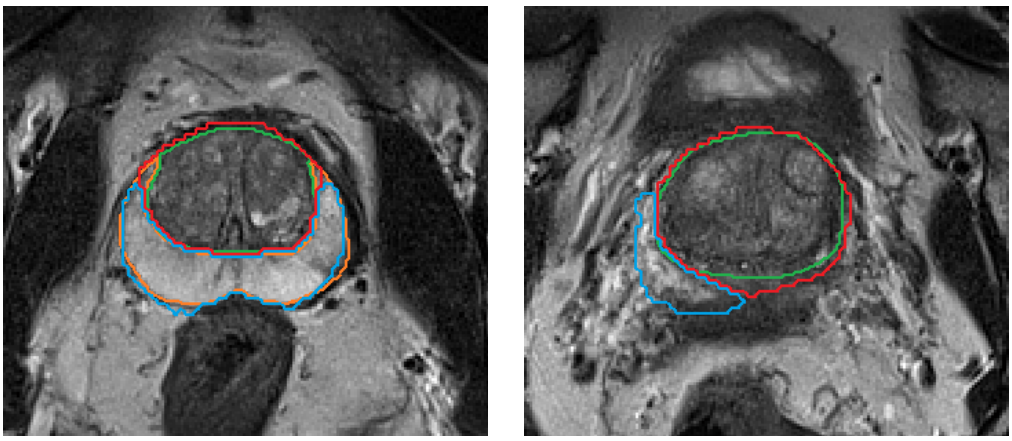
(d) (e) (f)

Figure 5.13: Figures (a), (b) and (c) illustrate the ground truth segmentation of the prostate zones from three different viewpoints. The green region represents the ground truth CG and the orange region indicates the ground truth PZ. Figures (d), (e) and (f) illustrate the predicted segmentation result of the prostate zones from three different viewpoints. The red region shows the predicted CG and the blue region indicates the predicted PZ. The capitalised letters represent direction-abbreviations of the coordinate system. anterior (A), posterior (P), superior (S), inferior (I), left (L) and right (R)

Next, Figure 5.13 shows a direct comparison of a predicted segmentation result to its

ground truth from three different viewpoints. The upper row in Figure 5.13 represents the ground truth and the lower row points out the prediction of the provided automated algorithm. The DC score of study *Prostate3T-01-0005* in Figure 5.13 is 0.80 for the PZ and 0.90 for the CG. Thus, case *Prostate3T-01-0005* is representing a well segmented study. The red circle in 5.13(e) indicates that the automated algorithm predicted PZ in two slices, although there is truly no PZ. An explanation for this behaviour is explained below.

Figure 5.14 presents two adjacent axial cross-section cuts at different position superimposed by the predicted segmentation result and ground truth segmentation. Both prostate zones show different gray value distributions and are therefore by the algorithm well classified. Figure 5.14 is visualized in the same color scheme as 5.13. Hence, the red boundary represents the predicted CG and the blue boundary indicates the predicted PZ. The ground truth is illustrated by the green and orange boundaries. Whereby, the green boundary indicates CG and PZ is marked by the orange boundary. The red circle in Figure 5.13(e) indicates an error in the predicted PZ and outlines the limitations of the proposed algorithm. The algorithm predicted PZ (blue region) despite the fact there is truly no PZ. The reason for that is the light bright structure at the bottom on the right of the CG. Exactly this structure is interpreted through the algorithm as PZ, because in the slices below exists peripheral zone at the same place with almost the same gray value distribution.



(a) (b)

Figure 5.14: Figures (a) and (b) show transversal cuts of the segmented prostate zones of the study *Prostate3T-01-0005*. It is the same study as in figure 5.13 and as in figure 5.12. The green boundary represents the ground truth CG and the red boundary indicates the predicted CG of the proposed automated algorithm utilizing the *A2-Mode*. Accordingly presents the orange boundary the ground truth of the PZ and the blue boundary the corresponding prediction

For the sake of completeness, the proposed algorithm is tested on the 25 1.5-Tesla studies to determine the performance and accuracy utilizing the optimal FFANN parameters. Hence, the error metric scores of cross-validation using 10 iterations are illustrated in

Figure 5.15. Again, the fraction of training was set to 0.83 and the utilized mode is the fully-automated *A2-Mode*. Using 25 1.5-Tesla studies in total leads in each iteration to 21 training studies and 4 prediction studies. Red boxes in Figure 5.15 highlight DC scores for the CG and blue boxes reveal corresponding scores for the PZ. Additionally, the mean and median values of each error metric are shown in Table 5.4.

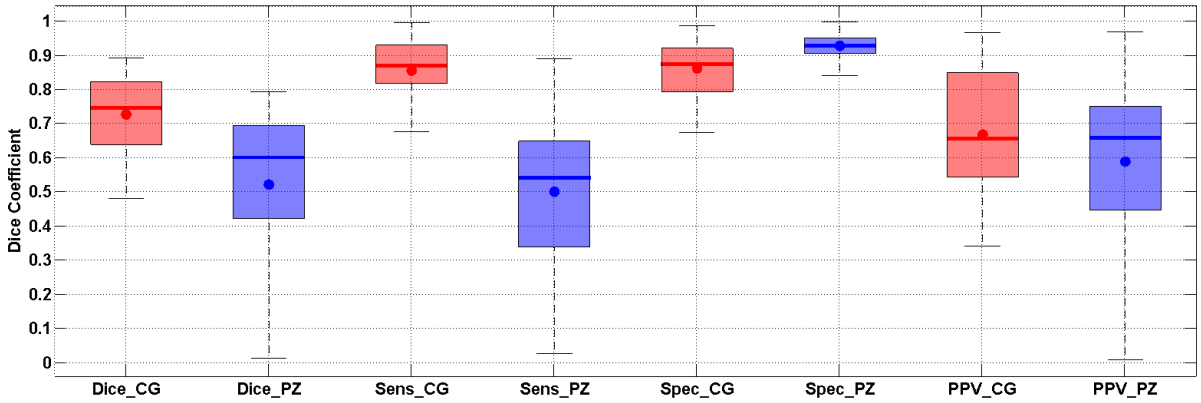


Figure 5.15: represents four error metrics scores in boxplots of prostate zone segmentation utilizing 25 1.5-Tesla studies as well as the *A2-Mode*. Four metrics are: Dice coefficient (DC), sensitivity (Sens), specificity (Spec) and positive predictive value (PPV). Each error metric is provided twice - once for the CG in red and once for the PZ in blue. The fraction of training studies is set to 0.83. Edges of the box present the 25th and 75th percentiles of each score. Slashes within the box characterize the median and dots represent the mean. Whiskers extend to the most extreme data points.

Table 5.4: Error metric scores for automated prostate zone segmentation on 25 1.5-Tesla studies

	DC CG	**DC PZ**	**Sens CG**	**Sens PZ**	**Spec CG**
mean	0.757	0.520	0.853	0.498	0.859
median	0.743	0.599	0.867	0.539	0.871
	Spec PZ	**PPV CG**	**PPV PZ**	**HdB CG**	**HdB PZ**
mean	0.925	0.666	0.587	7.99 *mm*	10.08 *mm*
median	0.926	0.655	0.657	6.34 *mm*	9.11 *mm*

The error metrics scores of Table 5.4 lie in the same range as error metric scores on 25 3-Tesla studies, which are illustrated in Table 5.3. The scores are considered to be in the same range, because the deviation of the mean DC scores for the CG and for the PZ are 0.03 and 0.05. The remaining error metric scores correlate as well. Due the fact, the deviations differ at the second position behind the decimal point, it is considered that

the proposed algorithm is independent to the magnetic field strength. This characteristic is caused by the feature normalization in Section 4.3.

Additionally, a volume estimation evaluation of the 25 3-Tesla studies is employed. Each study is predicted once without being in the training set simultaneously. Firstly, the accuracy of the described volume estimation algorithm in Section 4.6 is established. Therefore, the volumes of the PZs and the CGs are estimated through the proposed algorithm based on the ground truth labels. In addition, volumes of both zones are measured by the software application *3D Slicer* from the ground truth labels. The assessed volume of this measurement is considered to be the ground truth volume. Hence, the fraction of the estimated ground truth volume, assessed through the proposed algorithm and the volume measurement by *3D Slicer* is determined as $F_{GT\ CG}$ for the CG and as $F_{GT\ PZ}$ for the PZ. These fractions determine how accurate the proposed volume estimation algorithm performs on ground truth. Moreover, $F_{GT\ Sum}$ determines the volume fraction of both zones respectively, the prostate gland. This fraction is derived from $V_{GT\ CG\ FFANN} + V_{GT\ PZ\ FFANN}$ and $V_{GT\ CG\ 3DSlicer} + V_{GT\ PZ\ 3DSlicer}$.

Table 5.5: Volume estimation evaluation on 25 3-Tesla studies

Fraction	**Mean**	**Median**	**SD**	**STE**	r^2
$F_{GT\ CG} = V_{GT\ CG\ FFANN}/V_{GT\ CG\ 3DSlicer}$	0.940	0.943	0.027	0.005	0.999
$F_{GT\ PZ} = V_{GT\ PZ\ FFANN}/V_{GT\ PZ\ 3DSlicer}$	1.321	1.314	0.246	0.048	0.910
$F_{GT\ Sum}$	1,06	1.05	0.09	0.02	0,982
$F_{P\ CG} = V_{P\ CG\ FFANN}/V_{GT\ CG\ 3DSlicer}$	1.05	1.006	0.274	0.054	0.912
$F_{P\ PZ} = V_{P\ PZ\ 3DSlicer}/V_{GT\ PZ\ 3DSlicer}$	0.55	0.670	0.360	0.071	0.451
$F_{P\ Sum}$	0.87	0.895	0.152	0.030	0,844

Volume estimation evaluation results based on 25 3-Tesla studies. Abbreviations are as follows: standard deviation (SD), standard error (STE), Person correlation coefficient $\boldsymbol{r^2}$, ground truth labels (GT), predicted labels (P), central gland (CG), peripheral zone (PZ), FFANN indicates volumes are established by the proposed volume estimation algorithm in 4.6, *3DSlicer* indicates volumes are measured with *3DSlicer*

The mean, median, standard deviation (SD) and standard error (STE) as well as the Pearson correlation coefficient(r^2) are illustrated in the upper half of Table 5.5. Detailed results of the volume measurements as well as for $F_{GT\ CG}$ and $F_{GT\ PZ}$ are attached in the Appendix B.1. A volume fraction of 1.00 indicates the estimated volume is equal to the ground truth volume. Furthermore, indicates a volume fraction <1 an underestimated predicted volume and in contrast means a volume fraction >1, that the predicted volume is overestimated. A r^2 value of 0.999 for $F_{GT\ CG}$ indicates the volume estimation for the CG is almost superposable. Respectively, $F_{GT\ PZ}$ and $F_{GT\ Sum}$ achieve r^2 values of 0.910 and 0,982. The lower have in table 5.5 shows the mean, median, SD, STE and r^2 scores

for $F_{P\ CG}$, $F_{P\ PZ}$ and $F_{P\ Sum}$, whereby, $F_{P\ CG}$ presents the fraction of the estimated CG volume out of the prediction utilizing the proposed algorithm to the measured CG volume by *3D Slicer* based on the ground truth labels. Accordingly, $F_{P\ PZ}$ represents the fraction of the estimated PZ to ground truth and $F_{P\ Sum}$ the fraction of the estimated prostate gland to ground truth. Hence, $F_{P\ CG}$ and $F_{P\ PZ}$ determine for each zone the volume estimation accuracy of the predicted labels in comparison to the volumes based on the ground truth labels. The estimated volume of each zone logically depends strongly on the predicted labels. $F_{P\ CG}$ and $F_{P\ PZ}$ achieve mean volume fraction scores of 1.05 and 0.55 and r^2 values of 0.912 and 0.451. One of the most obvious consequences of a mean fraction of 0.55 for $F_{P\ PZ}$ is that the PZ is strongly underestimated throughout the proposed algorithm. $F_{P\ Sum}$ characterizes the fraction of the predicted prostate gland to the ground truth prostate gland. $F_{P\ Sum}$ achieves a mean volume fraction of 0.87 and a r^2 value of 0.84. Fractions $F_{P\ CG}$ and $F_{P\ PZ}$ of each study are illustrated in detail in the Appendix B.1. Volume estimation of the PZ provides the lowest score, which is caused by the imprecise prediction.

5.3 Evaluation of 100 MD Anderson Cancer Center Studies

This section reviews the performance of the proposed algorithm on a large dataset. Experts from the MD Anderson Cancer Center drew contours on 100 abdominal T2-weighted MRI studies via an implemented DICOM-Viewer in *Matlab*. These contours contain labels for the CG as well as for the PZ. Cross-validation utilizing 10 cross-validation iterations as well as the optimal FFANN parameters shown in Table 5.1 are employed.

In contrast to previous experiments the *I-Mode* is now utilized instead of the *A2-Mode*. The *I-Mode* or in other words the interactive mode requires a manually determined center point of the prostate as well as manually determined transverse(D_1) length, craniocaudal(D_2) length and anteroposterior (D_3) length of the prostate. Manual determinations are carried out through MD Anderson Cancer Center experts as well. Hence, the *I-Mode* presents a semi-automatic version of the proposed algorithm. Furthermore, the fraction of training studies is determined to be 0.83. As the learning curve in Figure 5.2 reveals the learning behaviour did not reach its saturation, an experiment utilizing all 100 studies is carried out. Thus, the number of training studies is 83 and accordingly, the number of prediction studies is 17. The results of the experiment utilizing the *I-Mode* on 100 MD Anderson Cancer Center studies are shown in Figure 5.16 and in table 5.6.

As shown in Table 5.6, all mean and median scores for the PZ increased significantly utilizing the interactive mode. Thus, the mean DC score for the PZ amounts to 0.69 on 100 studies. Furthermore, the remaining mean and median scores for the CG improved slightly, too. For instance, the mean DC scores for the CG are 0.81 on 100 studies. The mean Hausdorff Distance of Boundaries (HdB) achieves a score for the CG of 3.40 *mm* and 4.81 *mm* for the PZ. In summary, all mean and median error metric scores improved compared to all previous experiments. This improvement is caused by the minimal user interaction of about 20 seconds to determine the center point as well as the principle

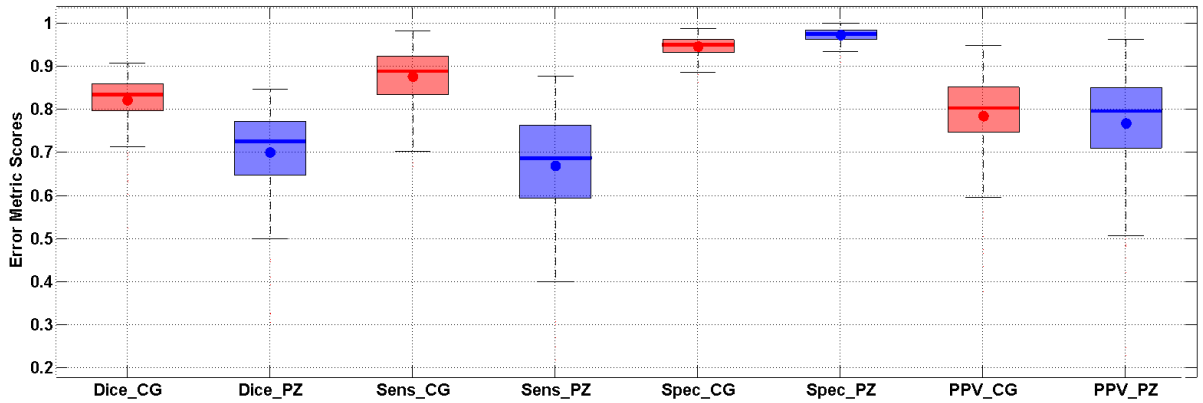


Figure 5.16: Four error metric scores in boxplots of prostate zone segmentation utilizing 100 MD Anderson Cancer Center studies and the *I-mode*. Four metrics are: DC, sensitivity (Sens), specificity (Spec) and PPV. Each error metric is provided twice - once for the CG and once for the PZ. Cross-validation on 100 studies utilizing the *I-Mode* is carried out. The fraction of training studies is set to 0.83. Red boxes highlight DC for the CG and blue boxes mark DC scores for the PZ. Edges of the boxes present the 25th and 75th percentiles of each score. Slashes within the box represent the median and dots represent the mean. Whiskers extend to the most extreme data points.

lengths D_1,D_2 and D_3. Consequently, the proposed algorithm is competitive to the work presented by M.Rusu et.al [32]. The processing time of one study on a machine comprised of an i7 core processor and 16GB of RAM averaged to 15 seconds. The segmentation results of each slice of one MD Anderson Cancer Center study are shown in the Appendix B.1. However, the characteristic of the averaged prediction error of the prostate gland shown in Figure 5.10 is transferable to automated and semi-automated prostate zone segmentation. This means that the proposed algorithm performs well in middle-slices. But the algorithms begins performing poorly approaching the start or end of the utilized image stack.

Because of having manually determined lengths D_1, D_2 and D_3 for each study available, volume estimation evaluation is carried out as well. The fraction of training studies is set to 0.80 and randomly picking training studies and prediction studies are disabled. This yields to 80 training and 20 prediction studies. In order to predict each study once without being in the training set leads to five experiments in total. Each time training and prediction studies are substituted in order to avoid using a study for training and prediction simultaneously. This procedure is repeated five times until all studies are predicted once.
Five prostate volume estimation techniques are proposed in total and shown in Table 5.7. The Ellipsoid, Myschetzky and the Prolate spheroid techniques require D_1, D_2 and D_3. The proposed volume estimation algorithm does not need D_1, D_2 and D_3 - it only requires the predicted prostate zones in combination with the corresponding DICOM-header in order to assess the xy-spacing as well as the z-spacing. The ground truth volume is

Table 5.6: Error metric scores for semi-automated prostate zone segmentation on 100 MD Anderson studies utilizing the *I-mode*

	DC CG	**DC PZ**	**Sens CG**	**Sens PZ**	**Spec CG**
mean	0.81	0.69	0.87	0.66	0.94
median	0.83	0.72	0.88	0.68	0.93
	Spec PZ	**PPV CG**	**PPV PZ**	**HdB CG**	**HdB PZ**
mean	0.96	0.78	0.76	3.40 mm	4.81 mm
median	0.97	0.80	0.79	2.99 mm	4.44 mm

established by measuring the volume in *3DSlicer* based on the ground truth labels.

Each volume estimation technique is compared to the ground truth by building the fraction of the estimated volume to the ground truth volume. This results for each study to volume fractions as follows: F_{Cube}, F_{Ell}, F_{Mys}, F_{Sph}, $V_{P\ CG}$, $V_{P\ PZ}$ and F_{Sum}, where F_{Sum} is derived from $(V_{CG} + V_{PZ})/V_{ground\ truth}$. All D_1, D_2 and D_3 measurements, estimated volumes and volume fractions of all 100 MD Anderson studies are attached in the Appendix in Table B.

Table 5.7: Enumeration of prostate volume estimation techniques with corresponding formulas

model	**description**	**volume estimation**
Cube	V_{Cube}	$D_1 * D_2 * D_3$
Ellipsoid	V_{Ell}	$D_1 * D_2 * D_3 * \pi/6$
Myschetzky	V_{Mys}	$D_1 * D_2 * D_3 * 0.7$
Prolate spheroid	V_{Sph}	$(D_1)^2 * D_2 * \pi/6$
FFANN	$V_{P\ CG}$, $V_{P\ PZ}$, $V_{P\ Sum}$	see section 4.6
Expert	$V_{ground\ truth}$	via 3DSlicer

For each volume fraction the mean, median, standard deviation (SD), standard error (STE) and the Pearson correlation coefficient r^2 over 100 studies are assessed and presented in Table 5.3. For better comparison, F_{Cube} is presented as well. Each following fraction (F_{Ell}, F_{Mys}, F_{Sph}, F_{Sum}) should have at least lower volume fraction scores than F_{Cube}. Because V_{Cube} basically multiplies the three principle prostate length, which results in the prostate's bounding box volume.

The proposed algorithm achieve mean volume fractions of 1.192 for the CG and 0.904 for the PZ. Associated standard deviations are 0.38 and 0.38. This implies that the CG is on average over-estimated and the PZ is on average under-estimated. Combing both regions leads to a compensation and results as a consequence in a mean volume fraction of 1.019 with a standard deviation of 0.159.
In contrast, F_{Ell}, F_{Mys} and F_{Sph} achieve mean volume fractions of 1.315, 1.758 and 1.744. Respectively, standard deviations are 0.032, 0.043 and 0.077. This means that each of these volume estimation technique over-estimates the prostate gland strongly. Which is basically caused by the corresponding formulas of each technique. According to the mean volume fraction of F_{Sum}, are the highest r^2 values (0.912, 0.649, 0.922) obtained using the proposed algorithm. Hence, clinical standard prostate volume estimation techniques, which achieve r^2 values of 0.761, 0.761 and 0.54 are far exceeded. Furthermore, the achieved r^2 value of 0.82, presented by R.Toth et al [29] is outperformed.

Table 5.8: Volume estimation evaluation on 100 MD Anderson studies

Fraction	**Mean**	**Median**	**SD**	**STE**	$\boldsymbol{r^2}$
F_{Cube}	2.511	2.412	0.623	0.062	0.761
F_{Ell}	1.315	1.263	0.326	0.032	0.761
F_{Mys}	1.758	1.689	0.436	0.043	0.761
F_{Sph}	1.744	1.655	0.771	0.077	0.540
F_{CG}	1.192	1.121	0.380	0.0038	0.912
F_{PZ}	0.904	0.898	0.388	0.039	0.649
F_{Sum}	1.019	1.028	0.159	0.016	0.922

Volume evaluation results of 100 MD Anderson Cancer Center studies. Abbreviations are as follows: standard deviation (SD), standard error (STE), Person correlation coefficient r^2, ground truth labels (GT), predicted labels (P), central gland (CG), peripheral zone (PZ)

Chapter 6

Discussion

Automated prostate structure segmentation in MR images provides information about the size, shape, position and volume of the prostate gland and prostate zones. Thus, the knowledge about the prostate increases, which can affect and improve as a consequence multiple fields such as prostate cancer staging, treatment selection as well as prostate biopsies. A fully-automated prostate zone and prostate gland segmentation algorithm from in vivo T2-weighted MRI studies is presented. In addition, a semi automated version of the proposed algorithm is carried out and tested on a large scale dataset. Subsequently, volume estimation based on the segmentation results has been proposed. The invented supervised machine learning algorithm constitutes multi layer feed forward artificial neural networks (FFANN). Hence, the proposed multi layer FFANN is trained by means of features in order to solve a multi-class classification problem. To achieve input data clustering, several hand engineered low-level features are extracted for each pixel from a pre-processed MR image stack. The proposed feature extraction incorporates multiple texture, distance, statistical, probabilistic as well as local neighborhood features. The fully-automated algorithm is comprised of two FFANN models, which are ordered consecutively. In contrast, the semi-automatic version contains only one FFANN. Basic postprocessing steps as morphological operations and Gaussian smoothing results in the algorithm's output. Building a three-dimensional surface point cloud from the algorithm's output in order to assess a closed surface model through triangulation represents the core of the proposed volume estimation algorithm. Consequently, volumes are estimated based on the closed surface models.

Multiple experiments have been proposed to establish the optimal FFANN parameters, which are as follows: $152 \times 150 \times 75 \times 3$ hidden neurons, batchsize of 10, one epoch, neighborhood size of 5, weight initialization method: random weight initialization, best training performance obtained on 21 training studies. The reason for achieving equal Dice coefficient (DC) scores with random weight initialization and deep belief network weight initialization is unknown, but it has been suggested by Hinton and Salakhutdinov [57] as a possible improvement.

Utilizing the optimal FFANN parameters as well as the strict error metric calculations, especially the strict DC calculation leads to a mean DC score for the prostate gland on

25-3 Tesla studies of 0.85. Hence, the presented results are competitive to others [33],[27]. Considering prostate zone segmentation on 25 3-Tesla and on 25 1.5-Tesla studies the mean DC scores are 0.78/0.75 for the central gland (CG) and 0.47/0.52 for the peripheral zone (PZ) without requesting any user interaction, whereby, the prediction time amounts to 30 seconds per study. Based on the fully-automated segmentation yields a subsequent volume estimation to mean volume fractions of 1.05 for the CG, 0.55 for the PZ and 0.87 for the prostate gland. Respectively, the Pearson correlation coefficients (r^2) values are 0.91, 0.45 and 0.84. These results present the first complete fully-automated prostate zone segmentation results in the literature based on in vivo T2-weighted MRI studies. Hence, fully-automated segmentation with subsequent volume estimation for the central gland as well as for the prostate gland achieves clinical relevant results. In contrast, scores for the peripheral zone are too imprecise and needs further investigations.

Enabling minimal user interaction in order to determine the principle lengths as well as the mass point of the prostate leads to a mean DC scores on 100 in vivo T2-weighted MRI studies of 0.81 and 0.69. It is the first time that semi-automated prostate zone segmentation is carried out on a large scale datasets like this. These scores are similar to current results presented in the literature, which have utilized smaller datasets [32]. The processing time amounts to 15 seconds per study. Volume estimation achieved mean volume fraction scores of 1.19 for the CG, 0.90 for the PZ and 1.01 for the prostate gland. Respectively, the r^2 values are 0.91, 0.64 and 0.92. This means that the proposed semi-automated algorithm provides segmentation results which correlate higher with the ground truth than traditional techniques, which are namely the Myschetzky (r^2=0.761), the Ellipsoid (r^2=0.761) and the Prolate spheroid (r^2=0.541) technique. Furthermore, r^2 values for the CG and prostate gland outperform current volume estimation methods presented the literature [29]. While the initial findings are promising, this study has highlighted existing problems for automated as well as semi-automated peripheral zone segmentation. By reason of the natural variability of the peripheral zone, further research is necessary to increase the segmentation for the peripheral zone in terms of accuracy.

In addition, this study raises a number of questions for future research by means of applying the existing algorithm to other organs. The proposed algorithm is not limited to learn just the prostate gland/zones from labels, it is able to learn any arbitrarily shape. Moreover, the algorithm's output can be utilized as basis for further high-level postprocessing or could be used in combination with an appearance based segmentation technique to achieve higher dice scores. Limitations of the proposed supervised algorithm include the fact that it has to be trained on manual expert segmentations in order to use it for fully-automated or semi-automated prostate zone segmentation. Concluding this study, the proposed prostate zone segmentation algorithm combined with the proposed volume estimation can save valuable time for clinicians by providing in realtime accurate prostate zone segmentations and accurate prostate volume estimations.

Acknowledgements

I would like to thank my main supervisor Marvin D. Hoffland for insightful suggestions and optimal guidance. In addition, I would like to thank my secondary supervisor Naveen Garg for his incessant support and trust as well as for his meaningful assistance. Moreover, this work was made possible by grants from the Austrian Marshall Plan Foundation[15] and the Industriellenvereinigung Kaernten[16].

[15] Austrian Marshall Plan Foundation, (accessed October 2013), http://www.marshallplan.at/
[16] IV Kaernten, (accessed October 2013), http://iv.ifit-e.uni-klu.ac.at/~iv/frontend/

Bibliography

[1] J. M. Harris and J. Scott, *Visual cortex: Anatomy, functions, and injuries*. Neurology - laboratory and clinical research developments, New York: Nova Science Publishers, 2012. 1

[2] E. Kandel, et al., *Principles of Neural Science*. McGraw Hill Medical, 5 ed., July 2012. 1

[3] V. Ladurantaye, et al., *Visual Cortex - Current Status and Perspectives*, ch. Chapter 10 - Models of Information Processing in the Visual Cortex, pp. 227–246. InTech, September 2012. 1

[4] S. Thorpe, et al., "Speed of processing in the human visual system," *Nature*, vol. 381, pp. 520–522, June 1996. 1

[5] G. Wallis and H. H. Bülthoff, "Learning to recognize objects," in *Trends in Cognitive Science*, pp. 22–31, 2000. 1

[6] I. N. Bankman, ed., *Handbook of Medical Image Processing and Analysis*. Academic Press, second ed., Dez 2008. 1, 3.1, 3.1, 3.2, 3.2

[7] T. Serre, L. Wolf and T. Poggio, "Object recognition with features inspired by visual cortex," *CVPR-Volume*, pp. 994–1000, 2005. 1

[8] C. Farabet, et al., "Learning hierarchical features for scene labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013. 1

[9] R. B. Palm, "Prediction as a candidate for learning deep hierarchical models of data," Master's thesis, 2012. 1, 4.4

[10] H. Lee, et al., "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *in International Conference on Machine Learning*, 2009. 1

[11] Geoffrey E. Hinton and S. Osindero, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, p. 2006, 2006. 1

[12] Geoffrey E. Hinton, "Learning multiple layers of representation," *Trends in Cognitive Sciences*, vol. 11, pp. 428–434, October 2007. 1

[13] M.Jabarouti, et al, "Medical image segmentation using artificial neural networks," *Artificial Neural Networks - Methodological Advances and Biomedical Applications*, April 2011. 1, 1

[14] OECD, "Health care resources - medical technology." online, June 2013. 1

[15] "2012 World Population sheet," anual report, Population Reference Bureau, 2012. 1

[16] American Cancer Society, "Cancer facts & figures 2013," tech. rep., American Cancer Society, Atlanta, GA 30303, 2013. 1, 1, 2.1, 2.1

[17] M. Mengel, W. Holleman and S. Fields, *Fundamentals of clinical practice*. New York: Kluwer Academic/Plenum Publishers, 2nd ed., 2002. 1, 1

[18] E. R. Davies, *Computer and Machine Vision: Theory, Algorithms, Practicalities*. Academic Press is an imprint of Elsevier 225 Wyman Street, Waltham, 02451, USA The Boulevard, Langford Lane, Kidlington, Oxford OX5 1GB, UK: Elsevier, 4th ed., 2012. 1, 3.1

[19] N. Sharma and L. Aggarwal, "Automated medical image segmentation techniques," *Journal of Medical Physics*, vol. 35, no. 1, pp. 3–14, 2010. 1

[20] J. Jiang, P. Trundle and J. Ren, "Medical image analysis with artificial neural networks," *Computerized Medical Imaging and Graphics*, vol. 34, pp. 617–631, December 2010. 1

[21] C. Tempany and F. Franco, "Prostate mri: Update and current roles," *Applied Radiology*, vol. 41, no. 3, pp. 17–22, 2012. 1, 2.1, 2.2, 2.2, 2.3

[22] G. Yan and B. Wang, "An automatic kidney segmentation from abdominal ct images," in *Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference on*, vol. 1, pp. 280–284, 2010. 1

[23] S. Sun, C. Bauer, and R. Beichel, "Automated 3-d segmentation of lungs with lung cancer in ct data using a novel robust active shape model approach," *Medical Imaging, IEEE Transactions on*, vol. 31, no. 2, pp. 449–460, 2012. 1

[24] J. Haaga and C. Lanzieri, *CT and MR Imaging of the Whole Body*. 11830 Westline Industrial Drive, St. Louis, Missouri 63146: Mosby, 4th ed., 2003. 1

[25] A. D'Amico, et al., "Calculated prostate cancer volume: The optimal predictor of actual cancer volume and pathologic stage," *Urology*, vol. 49, pp. 385–391, May 1997. 1, 2.1

[26] A. Heidenreich, et al., *Guidelines on Prostate Cancer*. European Association of Urology, 2011. 1, 2.1

[27] G. Vincent, et al., "Fully automatic segmentation of the prostate using active appearance models," 2012. 1.1, 5.2.1, 5.2.1, 6

[28] R. Toth, R. Sparks, and A. Madabhushi, "Medial axis based statistical shape model (massm): Applications to 3d prostate segmentation on mri.," in *ISBI*, pp. 1463–1466, IEEE, 2011. 1.1

[29] R. Toth, et al., "Accurate prostate volume estimation using multifeature active shape models on t2-weighted mri," *Academic Radiology*, vol. 18, pp. 745–754, June 2011. 1.1, 5.3, 6

[30] S. Ghose, A. Oliver, R. Marti´, X. Llado, J. Freixenet, J. Mitra, J. Vilanova, J. Comet, and F. Meriaudeau, "Statistical shape and probability prior model for automatic prostate segmentation," in *Digital Image Computing Techniques and Applications (DICTA), 2011 International Conference on*, pp. 340–345, 2011. 1.1

[31] G.Litjens, N.Karssemeijer and H.Huisman, "A multi-atlas approach for prostate segmentation in mr images," 2012. 1.1

[32] M.Rusu, et al., "Statistical 3d prostate imaging atlas construction via anatomically constrained registration," vol. 2013. 1.1, 5.3, 6

[33] e. a. M. Yang, "Prostate segmentation in mr images using discriminant boundary features," *Transactions on Biomedical Engineering*, vol. 60, pp. 479–488, February 2013. 1.1, 5.2.1, 6

[34] R. Shah and M. Zhou, *Prostate Biopsy Interpretation: An Illustrated Guide*. Springer, 2012. 2.1

[35] American Cancer Society, "Prostate cancer," tech. rep., American Cancer Society, Atlanta, GA 30303, September 2012. 2.1, 2.2, 2.3

[36] H. Nguyen, et al., "Normal human ejaculatory duct anatomy: A study of cadaveric and surgical specimens," *The Journal of Urology*, vol. 155, pp. 1639–1642, May 1996. 2.1

[37] A.Hou, D.Swanson and A. Barqawi, "Modalities for imaging of prostate cancer," *Advances in Urology*, December 2009. 2.2, 2.3

[38] S. Vema and A.Rajesh, "A clinically relevant approach to imaging prostate cancer:review," *American Journal of Roentgenology*, vol. 196, pp. 11–14, December 2011. 2.2

[39] S. Tabatabaei, et al., "Prostate cancer imaging: What surgeons, radiation oncologists, and medical oncologists want to know," *American Journal of Roentgenology.*, vol. 196, pp. 1263–1266, June 2011. 2.2

[40] A.HU, et al., "Is it time to consider a role for mri before prostate biopsy?," *Clinical Oncology*, vol. 6, pp. 197–206, May 2009. 2.2

[41] M.Roethke, et al. , "Mri-guided prostate biopsy detects clinically significant cancer: analysis of a cohort of 100 patients after previous negative trus biopsy," *World Journal of Urology*, vol. 30, pp. 213–218, April 2012. 2.2, 2.3

[42] D. Engehausen, et al., "Magnetic resonance image-guided biopsies with a high detection rate of prostate cancer," *The Scientific World Journal*, vol. 2012, no. Article ID 975971, p. 6 pages, 2012. 2.2, 2.3

[43] H. Hedvig, et al., "Imaging prostate cancer: A multidisciplinary perspective1," *Radiology*, vol. 243, no. 1, pp. 28–53, 2007. 2.2

[44] H. Jadvar, "Prostate cancer: Pet with 18f-fdg, 18f- or 11c-acetate, and 18f- or 11c-choline.," *Journal of Nuclear Medicine*, vol. 52, pp. 81–89, January 2011. 2.2

[45] W. Akhter, "Role of mri in the diagnosis of prostate cancer," *Clinical and Experimental Medical Sciences*, vol. 1, no. 3, pp. 111–116, 2013. 2.3

[46] CM.Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., 2006. 3, 3.2

[47] K. P. Murphy, *Machine learning: a probabilistic perspective*. 2012. 3

[48] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*. Thomson-Engineering, 2007. 3.1, 3.1

[49] I. Levner, *Data Driven Object Segmentation*. dissertation, Department of Computer Science,University of Alberta, Edmonton, Spring 2009. 3.1

[50] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998. 3.2

[51] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, 2013. 3.3

[52] J. Flores, *Focus on Artificial Neural Networks*. Engineering tools, techniques and tables, Nova Science Publishers, Incorporated, 2011. 3.3.1

[53] D. Kriesel, *A Brief Introduction to Neural Networks*. 2007. available at http://www.dkriesel.com. 3.3.1

[54] G. E. Hinton and S. Osindero, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, p. 2006, 2006. 3.3.1, 3.3.2

[55] G. Hinton, "A Practical Guide to Training Restricted Boltzmann Machines," tech. rep., 2010. 3.3.2, 3.3.2

[56] A. Fischer and C. Igel, "An introduction to restricted boltzmann machines," in *CIARP*, pp. 14–36, 2012. 3.3.2, 3.3.2

[57] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, pp. 504–507, July 2006. 3.3.2, 6

[58] H.-P. Wieser, "Robust volume calculation of closed surface models." Bachelor Thesis, July 2011. 4.6

[59] G. M. e. a. Hoffelt S, Marshall L, "Transrectal ultrasonic volumetry of the prostate: in vivo comparison of different methods," *Prostate*, vol. 57, pp. 29–32, 1996. 4.6

[60] P. Myschetzky, R. Suburu, B. Kelly, M. Wilson, S. Chen, and F. Lee, "Determination of prostate gland volume by transrectal ultrasound," *Scandinavian Journal of Urology and Nephrology. Supplement*, vol. 137, pp. 107–11, 1991. 4.6

[61] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Effiicient backprop," in *Neural Networks: Tricks of the Trade, this book is an outgrowth of a 1996 NIPS workshop*, (London, UK, UK), pp. 9–50, Springer-Verlag, 1998. 5.1.4

Table of Abbreviations

VC visual cortex

LGN lateral geniculate nucleus

OECD Organisation for Economic Co-operation and Development

MRI magnetic resonance imaging

MR magnetic resonance

DC Dice coefficient

TRUS transrectal ultrasonography

PSA prostate specific antigen

CG central gland

PZ peripheral zone

US ultrasonography

MRS magnetic resonance spectroscopy

CT computed tomography

PET positron emission tomography

DRE digital rectal exam

MAP maximum a posteriori estimation

MNIST Mixed National Institute of Standards and Technology database

ANN artificial neural network

FFANN Feed Forward Artificial Neural Network

DBFFANN deep belief feed forward artificial neural network

DBN deep belief network

RBM Restricted Boltzmann Machine

BMC Boston Medical Center

RUNMC Radboud University Nijmegen Medical Centre

DICOM Digital Imaging and Communications in Medicine

NRRD nearly raw raster data

SV stored values

RWV real world values

DISP display values

NHS neighborhood size

VU volume units

HdB Hausdorff Distance of Boundaries

PPV positive predictive value

FP false positive

TP true positive

FN false negative

TN true negative

BTP best training performance

WIM weight initialization method

A anterior

P posterior

L left

R right

I inferior

S superior

SD standard deviation

STE standard error

Appendix A

Used Tools

A.1 Hardware

All calculations were performed on a machine comprised of a i7 core processor and 16GB of RAM.

A.2 Software

Matlab: a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numerical computation. MATLAB, allows us to solve technical computing problems faster than with traditional programming languages, such as C, C++, and Fortran. (see http://www.matlab.com)

3D Slicer: a free, open source software package for image analysis and scientific visualization. Slicer is used in a variety of medical applications, including autism, multiple sclerosis, systemic lupus erythematosus, prostate cancer, schizophrenia, orthopedic biomechanics, COPD, cardiovascular disease and neurosurgery. (see http://www.slicer.org/ and http://en.wikipedia.org/wiki/3DSlicer)

Appendix B

Additional Figures and Tables

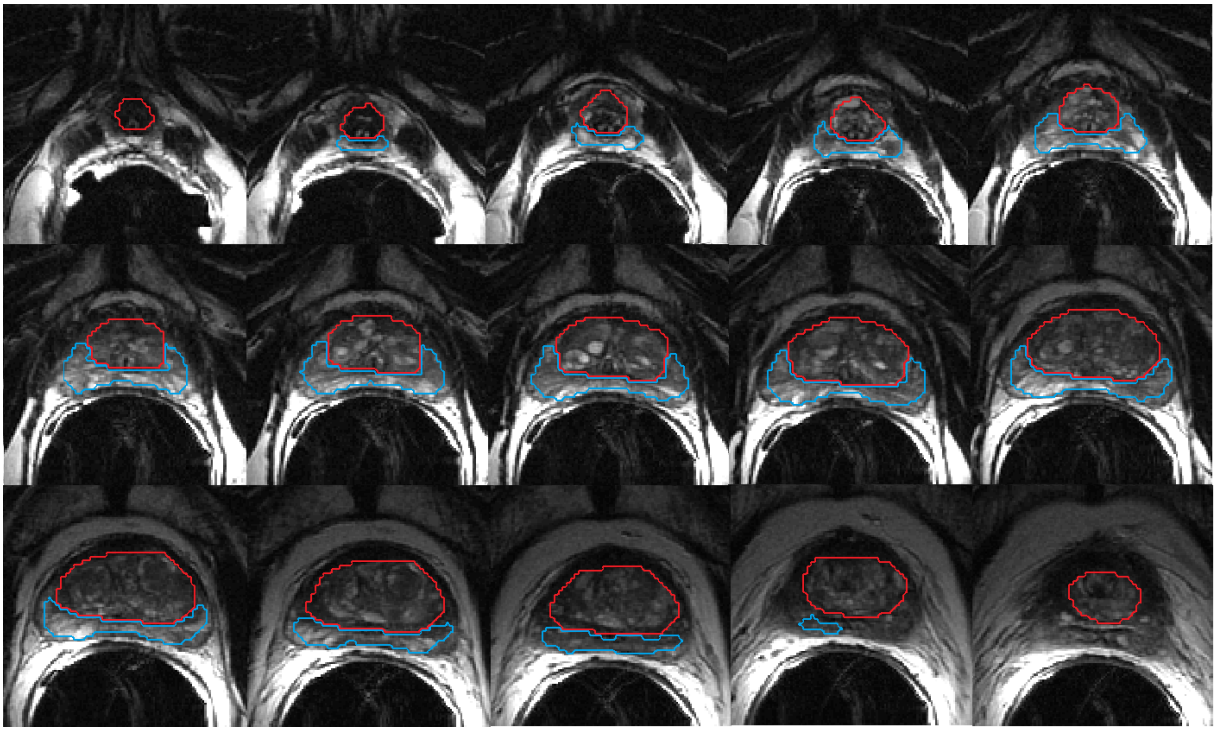


Figure B.1: Prostate zone segmentation result of one MD Anderson Cancer Center study. The red boundary indicates the predicted central gland and the blue boundary represents the predicted peripheral zone

Table B.1: Volume estimations and volume fractions of 25 3-Tesla studies

FFANN		FFANN		3D Slicer		3D Slicer		Frac GT to GT		Frac Pre to GT	
$V_{P\ CG}$	$V_{P\ PZ}$	$V_{GT\ CG}$	$V_{GT\ PZ}$	$V_{P\ CG}$	$V_{P\ PZ}$	$V_{GT\ CG}$	$V_{GT\ PZ}$	$F_{GT\ CG}$	$F_{GT\ PZ}$	$F_{P\ CG}$	$F_{P\ PZ}$
cm^3	cm^3	cm^3	cm^3	cm^3	cm^3	cm^3	cm^3	cm^3	cm^3	cm^3	cm^3
35,81	5,48	37,68	6,21	37,74	4,39	37,94	8,18	0,993	0,759	0,944	0,670
53,86	9,94	50,29	28,17	56,61	8,6	53,55	19,14	0,939	1,472	1,006	0,519
52,54	17,68	46,38	29,39	55,51	13,12	49,33	20,935	0,940	1,404	1,065	0,845
60,39	5,75	59,78	22,19	63,32	5,89	62,26	18,09	0,960	1,227	0,970	0,318
32,81	13,77	28,09	19,31	34,66	12,34	29,94	14,72	0,938	1,312	1,096	0,935
42,62	10,2	41,16	16,16	44,12	6,15	44,91	12,07	0,916	1,339	0,949	0,845
31,95	5,82	22,9	33,18	34,11	6,67	23,81	23,24	0,962	1,428	1,342	0,250
39,5	4,01	36,63	6,59	41,84	0,16	39,29	5,015	0,932	1,314	1,005	0,801
53,64	3,93	45,01	33,75	56,29	49,57	47,71	20,05	0,943	1,683	1,124	0,196
68,07	0,17	87,1	32,17	41,78	0,822	91,77	19,49	0,949	1,651	0,415	0,009
20,59	10,13	14,12	13,58	22,28	6,94	16,74	12,52	0,843	1,085	1,230	0,809
48,09	0,35	27,62	33,28	50,99	6,94	29,86	18,86	0,925	1,765	1,611	0,019
47,27	13,95	43,84	20,72	50,55	10,69	46,31	12,66	0,947	1,637	1,021	1,102
22,08	19,01	12,77	26,07	23,59	15,33	13,76	21,51	0,928	1,212	1,605	0,884
20,73	0,35	11,19	16,54	23,13	0,819	12,56	11,15	0,891	1,483	1,650	0,031
47,88	1,14	76,47	22,82	50,61	4,12	79,52	14,29	0,962	1,597	0,602	0,080
32,91	14,41	31,36	20,96	34,92	12,05	33,55	16,98	0,935	1,234	0,981	0,849
42,83	2,018	55,7	17,01	44	5,6	59,1	13,66	0,942	1,245	0,725	0,148
82,14	4,22	96,33	5,24	86,09	5,87	101	5,03	0,954	1,042	0,813	0,839
47,64	0,16	53,65	8,42	50,17	1,35	56,47	8,8	0,950	0,957	0,844	0,018
72,35	11,34	75,94	14,36	75,42	4,5	80,04	12,92	0,949	1,111	0,904	0,878
50,37	20,33	46,74	25,57	53,24	9,265	49,56	25,28	0,943	1,011	1,016	0,804
54,88	18,16	71,63	29,25	54,44	18,03	74,83	22,86	0,957	1,280	0,733	0,794
51,64	10,15	47,54	20,74	54,09	13,1	50,15	15,19	0,948	1,365	1,030	0,668
20,74	8,09	15,91	30,21	22,19	6,79	16,89	21,5	0,942	1,405	1,228	0,376

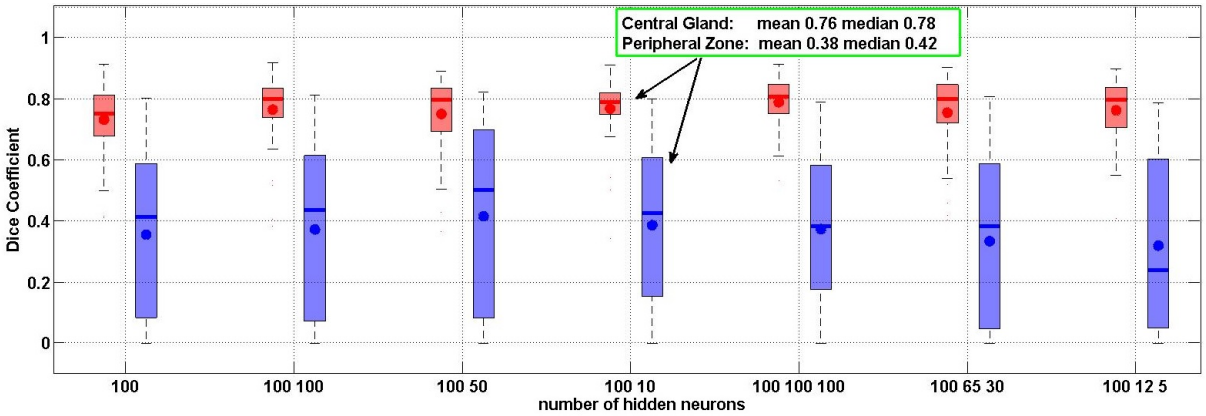
represents the volume measurements of 25 3-Tesla cases. The columns are defined as follows: *fist column* - predicted CG volume $V_{P\ CG}$ based on the proposed algorithm; *second column* - predicted PZ volume $V_{P\ PZ}$ based on the proposed algorithm; *third column* - ground truth CG volume $V_{GT\ CG}$ based on the proposed algorithm; *fourth column* - ground truth PZ volume $V_{GT\ PZ}$ based on the proposed algorithm; *fifth column* - predicted CG volume $V_{P\ CG}$ measured with 3D Slicer; *sixth column* - predicted PZ volume $V_{P\ PZ}$ measured with 3D Slicer; *seventh column* - ground truth CG volume $V_{GT\ CG}$ measured with 3D Slicer; *eights column* - ground truth PZ volume $V_{GT\ PZ}$ measured with 3D Slicer; *ninth column* - fraction $F_{GT\ CG}$ of ground truth CG volume based on the proposed algorithm and the ground truth volume of the CG measured with 3D Slicer $F_{GT\ CG} = V_{GT\ CG\ FFANN}/V_{GT\ CG\ 3DSlicer}$; *tenth column* - fraction $F_{GT\ PZ}$ of the ground truth volume of the PZ based on the proposed algorithm and the ground truth volume of the PZ measured with 3D Slicer $F_{GT\ PZ} = V_{GT\ PZ\ FFANN}/V_{GT\ PZ\ 3DSlicer}$; *eleventh column* - fraction $F_{P\ CG}$ of the predicted CG volume based on the proposed algorithm and the predicted CG volume measured with 3D Slicer $F_{P\ CG} = V_{P\ CG\ FFANN}/V_{GT\ CG\ 3DSlicer}$; *twelfth column* - fraction $F_{P\ PZ}$ of the predicted PZ volume based on the proposed algorithm and the predicted PZ volume measured with 3D Slicer $F_{P\ PZ} = V_{P\ PZ\ FFANN}/V_{GT\ PZ\ 3DSlicer}$

Table B.2: Volume estimations and volume fractions of MD Anderson studies 1-50

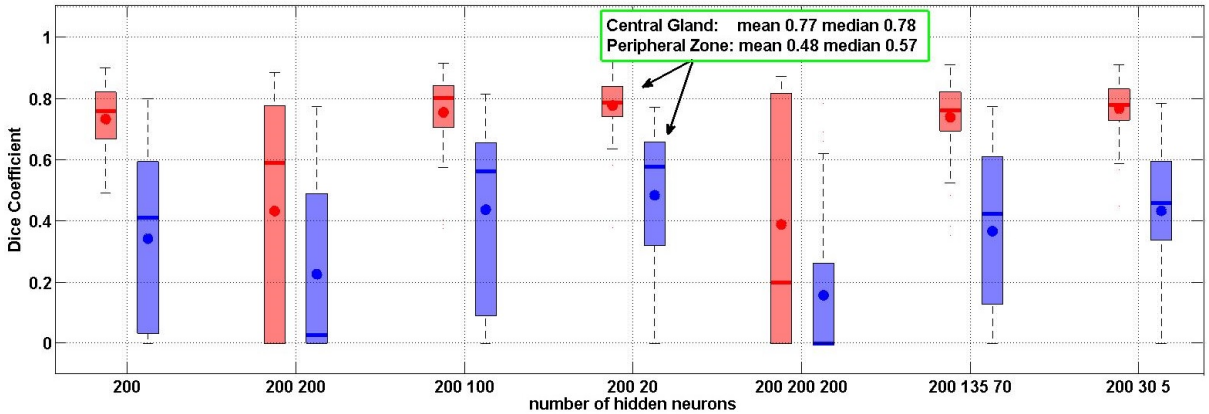
Case	D1 cm	D2 cm	D3 cm	V_{Cube} cm^3	V_{Ell} cm^3	V_{Mys} cm^3	V_{Sph} cm^3	F_{Cube} 1	F_{Ell} 1	F_{Mys} 1	F_{Sph} 1	F_{CG} 1	F_{PZ} 1	F_{Sum} 1
1	2,6	4,6	2,4	28,704	15,03	20,09	16,14	1,048	0,549	0,734	0,589	1,122	0,969	1,028
2	4,8	2,3	4,8	52,992	27,75	37,09	27,50	2,391	1,252	1,674	1,241	1,101	1,058	1,080
3	3,1	5,2	6,1	98,332	51,48	68,83	25,94	3,165	1,657	2,215	0,835	0,996	0,557	0,800
4	5,6	3,7	5,5	113,96	59,67	79,77	60,22	4,149	2,172	2,904	2,192	0,952	1,112	1,030
5	4,3	2,5	4,6	49,45	25,89	34,62	23,99	3,150	1,649	2,205	1,528	1,052	0,766	0,928
6	3	2,4	4,4	31,68	16,59	22,18	11,21	1,829	0,958	1,280	0,647	0,978	2,137	1,187
7	5,3	5	2,8	74,2	38,85	51,94	72,90	2,833	1,483	1,983	2,783	0,940	0,862	0,905
8	1,8	4,4	4,3	34,056	17,83	23,84	7,40	2,151	1,126	1,506	0,467	0,975	0,957	0,968
9	4,2	2,2	3,6	33,264	17,42	23,28	20,14	2,089	1,094	1,463	1,265	1,039	1,277	1,119
10	2,6	4,5	4,4	51,48	26,95	36,04	15,79	3,172	1,661	2,220	0,973	1,224	0,929	1,063
11	5,4	4,1	3,3	73,062	38,25	51,14	62,05	2,189	1,146	1,533	1,860	1,077	0,923	1,013
12	2,7	4,7	4,2	53,298	27,91	37,31	17,78	2,776	1,453	1,943	0,926	1,267	1,755	1,453
13	5,9	4,5	3,1	82,305	43,09	57,61	81,30	2,621	1,372	1,835	2,589	1,080	1,297	1,155
14	5	4,7	3,5	82,25	43,06	57,58	60,99	2,644	1,384	1,851	1,960	1,055	1,105	1,073
15	2,7	4,5	4,3	52,245	27,35	36,57	17,03	2,297	1,203	1,608	0,749	0,923	0,967	0,946
16	5,4	4,6	3,4	84,456	44,22	59,12	69,62	4,060	2,126	2,842	3,347	1,170	0,331	0,815
17	5,3	3,2	4,9	83,104	43,51	58,17	46,65	1,869	0,978	1,308	1,049	1,571	0,701	0,983
18	5,3	3,1	4,8	78,864	41,29	55,20	45,20	2,719	1,423	1,903	1,558	0,955	0,880	0,914
19	5	1,9	3,8	36,1	18,90	25,27	24,65	2,405	1,259	1,684	1,642	0,802	0,925	0,860
20	5,9	4,4	3,1	80,476	42,14	56,33	79,50	2,508	1,313	1,755	2,477	1,124	0,880	1,019
21	4,7	5	3,3	77,55	40,60	54,29	57,33	2,608	1,366	1,826	1,928	0,873	1,393	1,029
22	3,6	5,6	4,5	90,72	47,50	63,50	37,67	4,014	2,102	2,810	1,667	0,921	0,866	0,892
23	4,5	2,3	3,1	32,085	16,80	22,46	24,17	1,664	0,871	1,165	1,254	1,005	1,068	1,037
24	4,8	4,3	2,6	53,664	28,10	37,56	51,42	1,849	0,968	1,294	1,771	1,128	1,108	1,121
25	3,8	2,9	3,5	38,57	20,19	27,00	21,73	1,516	0,794	1,061	0,854	1,123	1,635	3,209
26	5,4	4,5	3,7	89,91	47,08	62,94	68,11	2,336	1,223	1,635	1,769	0,790	1,635	1,007
27	4,5	4,9	2,3	50,715	26,55	35,50	51,50	2,796	1,464	1,957	2,839	1,079	1,320	1,189
28	2,6	4,6	3,3	39,468	20,66	27,63	16,14	2,355	1,233	1,648	0,963	1,018	0,695	0,900
29	2,4	3	3,5	25,2	13,19	17,64	8,97	2,794	1,463	1,956	0,994	1,014	1,470	1,196
30	3,7	2,2	4,7	38,258	20,03	26,78	15,63	2,333	1,221	1,633	0,953	0,865	1,265	1,016
31	4,5	3,7	3	49,95	26,15	34,97	38,89	2,235	1,170	1,564	1,740	1,244	0,726	0,993
32	4,6	2,7	3,5	43,47	22,76	30,43	29,65	2,387	1,250	1,671	1,628	1,246	0,898	1,049
33	5,1	4,7	3,1	74,307	38,91	52,01	63,45	2,616	1,369	1,831	2,233	1,124	1,020	1,084
34	4,7	4,1	2,9	55,883	29,26	39,12	47,01	2,824	1,478	1,977	2,375	1,249	0,860	1,077
35	5,2	4,9	3,2	81,536	42,69	57,08	68,77	3,111	1,629	2,178	2,624	0,949	1,154	1,036
36	3,7	4,5	3	49,95	26,15	34,97	31,97	2,422	1,268	1,696	1,551	1,161	1,120	1,143
37	4,7	3,1	3,4	49,538	25,94	34,68	35,54	2,264	1,185	1,585	1,624	1,038	1,362	1,154
38	3,5	2,1	4,7	34,545	18,09	24,18	13,35	2,236	1,171	1,565	0,864	1,476	1,272	1,368
39	5	3,3	3	49,5	25,92	34,65	42,82	1,983	1,038	1,388	1,716	1,118	0,625	0,946
40	4,4	4,2	2,1	38,808	20,32	27,17	42,20	3,239	1,696	2,268	3,523	1,041	1,293	1,161
41	4,4	3,5	3	46,2	24,19	32,34	35,17	2,729	1,429	1,910	2,077	1,052	0,875	0,952
42	5,1	4,1	3	62,73	32,84	43,91	55,35	2,691	1,409	1,884	2,374	1,151	1,283	1,213
43	3,1	5,4	4,5	75,33	39,44	52,73	26,93	2,670	1,398	1,869	0,955	1,220	0,570	0,937
44	5,6	3,5	5,4	105,84	55,42	74,09	56,97	2,808	1,470	1,966	1,511	1,154	0,360	0,886
45	4,4	4,1	2,3	41,492	21,72	29,04	41,20	2,794	1,463	1,956	2,774	2,391	0,958	1,358
46	6	6	4,4	158,4	82,94	110,88	112,11	3,571	1,870	2,500	2,527	1,130	0,388	0,765
47	4,6	2,6	5,7	68,172	35,69	47,72	28,55	2,941	1,540	2,059	1,232	1,264	0,101	0,759
48	4,9	3,2	4,2	65,856	34,48	46,10	39,88	2,505	1,312	1,753	1,517	1,066	1,002	1,038
49	5,5	3	3,5	57,75	30,24	40,43	47,10	3,894	2,039	2,726	3,176	1,208	0,615	0,895

Table B.3: Volume estimations and volume fractions of MD Anderson studies 51-100

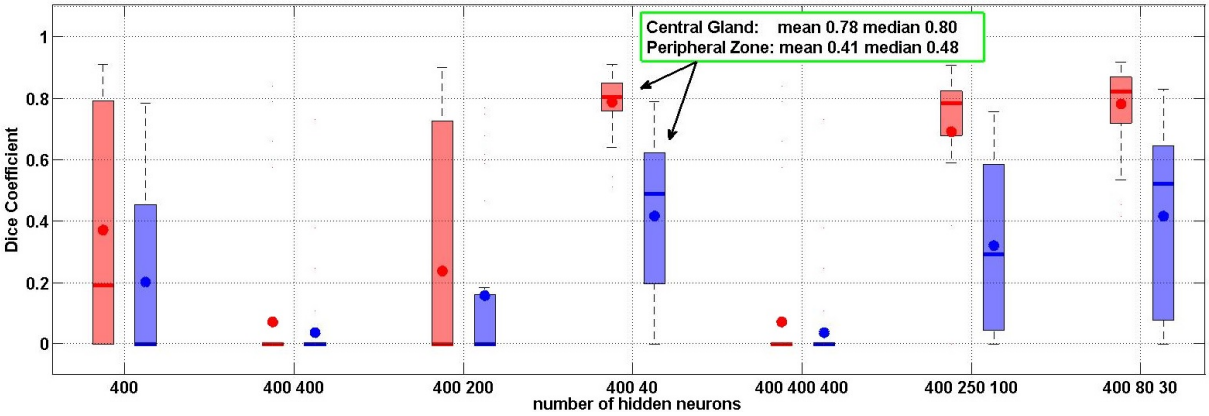
Case	D1 cm	D2 cm	D3 cm	V_{Cube} cm^3	V_{Ell} cm^3	V_{Mys} cm^3	V_{Sph} cm^3	F_{Cube} 1	F_{Ell} 1	F_{Mys} 1	F_{Sph} 1	F_{CG} 1	F_{PZ} 1	F_{Sum} 1
51	5	4,5	2,6	58,5	30,63	40,95	58,39	3,351	1,754	2,345	3,344	1,161	1,300	1,229
52	4,7	4,5	2,5	52,875	27,68	37,01	51,59	3,862	2,022	2,704	3,769	1,096	0,037	0,754
53	5,4	5,5	3,3	98,01	51,32	68,61	83,24	3,390	1,775	2,373	2,879	1,038	1,417	1,157
54	4,2	5,5	2,2	50,82	26,61	35,57	50,36	1,875	0,982	1,313	1,858	1,118	1,341	1,209
55	4,6	2,5	4,2	48,3	25,29	33,81	27,46	2,360	1,235	1,652	1,341	1,055	1,829	1,286
56	3,5	6,3	3	66,15	34,64	46,31	40,06	1,488	0,779	1,041	0,901	1,224	1,498	1,342
57	4,4	5	3,1	68,2	35,71	47,74	50,24	2,313	1,211	1,619	1,704	1,054	0,428	0,925
58	4,9	4,9	3	72,03	37,71	50,42	61,06	2,289	1,198	1,602	1,940	1,122	1,070	1,096
59	4,5	3	4,7	63,45	33,22	44,42	31,53	2,880	1,508	2,016	1,431	0,902	1,427	1,137
60	5,5	2,8	3,4	52,36	27,41	36,65	43,96	1,870	0,979	1,309	1,570	1,362	0,796	1,184
61	5,5	2,5	4,6	63,25	33,12	44,28	39,25	2,191	1,147	1,534	1,360	1,562	0,812	1,088
62	4,6	2,9	2,7	36,018	18,86	25,21	31,85	2,569	1,345	1,798	2,272	1,057	1,052	1,055
63	3,2	4,9	4,7	73,696	38,59	51,59	26,04	2,191	1,147	1,534	0,774	0,980	0,870	0,950
64	4,5	2,5	3,9	43,875	22,97	30,71	26,28	2,339	1,225	1,637	1,401	1,626	0,835	1,131
65	4,6	3,1	4,8	68,448	35,84	47,91	34,05	2,798	1,465	1,959	1,392	1,027	1,384	1,145
66	4	2,2	4,3	37,84	19,81	26,49	18,27	2,015	1,055	1,410	0,973	1,222	0,898	1,055
67	2	3,3	2,8	18,48	9,68	12,94	6,85	2,293	1,200	1,605	0,850	1,175	0,297	0,822
68	5	4	2,6	52	27,23	36,40	51,90	2,231	1,168	1,562	2,227	1,185	0,547	0,904
69	4,1	5,2	2,8	59,696	31,26	41,79	45,37	2,618	1,371	1,833	1,990	1,568	0,494	1,107
70	5,6	5,5	3,1	95,48	49,99	66,84	89,52	2,784	1,457	1,949	2,610	1,343	0,912	1,119
71	5	2,8	4,1	57,4	30,05	40,18	36,33	2,678	1,402	1,875	1,695	1,183	0,043	0,665
72	4,4	3,6	2,4	38,016	19,90	26,61	36,17	2,733	1,431	1,913	2,601	1,121	0,446	0,790
73	5,1	4,2	3,4	72,828	38,13	50,98	56,70	2,982	1,561	2,088	2,322	1,541	0,924	1,162
74	5	4,3	3	64,5	33,77	45,15	55,80	2,978	1,559	2,084	2,576	1,524	0,409	0,915
75	2,7	5,1	4,4	60,588	31,72	42,41	19,30	2,420	1,267	1,694	0,771	1,419	0,987	1,174
76	2,8	3,2	2,5	22,4	11,73	15,68	13,02	3,829	2,005	2,680	2,226	1,201	0,174	0,763
77	4,5	3,1	4,9	68,355	35,79	47,85	32,58	2,043	1,070	1,430	0,974	1,003	1,068	1,027
78	4,7	3	3,2	45,12	23,62	31,58	34,40	1,704	0,892	1,193	1,299	1,658	0,783	1,078
79	5	4,8	2,5	60	31,42	42,00	62,28	3,023	1,583	2,116	3,138	1,414	0,473	0,971
80	4,4	1,4	3,5	21,56	11,29	15,09	14,07	1,642	0,860	1,149	1,071	1,414	0,421	0,713
81	4,9	5	3,5	85,75	44,90	60,03	62,31	3,105	1,626	2,173	2,256	1,207	1,038	2,294
82	5,5	3,3	5,1	92,565	48,47	64,80	51,81	1,651	0,865	1,156	0,924	0,928	0,490	0,720
83	5,5	5,6	3,5	107,8	56,44	75,46	87,92	2,051	1,074	1,436	1,673	0,743	1,749	0,911
84	4	4,3	2,8	48,16	25,22	33,71	35,71	2,362	1,237	1,653	1,751	1,066	0,745	1,034
85	5,4	3,9	2,4	50,544	26,46	35,38	59,03	2,224	1,164	1,557	2,597	1,157	1,019	1,092
86	4,1	5,5	3,2	72,16	37,78	50,51	47,99	2,259	1,183	1,581	1,502	0,965	0,530	0,797
87	4,7	2,2	4,2	43,428	22,74	30,40	25,22	2,363	1,237	1,654	1,372	0,847	0,657	0,750
88	4,7	3,3	3,9	60,489	31,67	42,34	37,84	1,951	1,021	1,365	1,220	1,248	0,517	0,805
89	5,5	4,5	2,8	69,3	36,28	48,51	70,65	1,763	0,923	1,234	1,798	1,096	0,668	0,847
90	3,7	4,5	3	49,95	26,15	34,97	31,97	1,595	0,835	1,117	1,021	1,075	0,865	0,968
91	5,4	4,8	3,4	88,128	46,14	61,69	72,65	2,716	1,422	1,901	2,239	0,922	0,814	0,877
92	5,1	4,5	6,3	144,585	75,70	101,21	60,75	2,253	1,180	1,577	0,947	1,392	0,714	1,072
93	5,9	3,5	4,3	88,795	46,49	62,16	63,24	2,464	1,290	1,725	1,755	0,983	0,681	1,460
94	3,8	2,3	3,8	33,212	17,39	23,25	17,24	1,733	0,908	1,213	0,900	1,140	0,923	1,018
95	5,2	3,6	2,3	43,056	22,54	30,14	50,52	2,089	1,094	1,462	2,451	1,106	0,704	0,855
96	4,5	2,6	4,5	52,65	27,57	36,86	27,33	3,459	1,811	2,421	1,795	1,099	0,681	0,930
97	4,7	4	2	37,6	19,69	26,32	45,86	2,684	1,405	1,879	3,273	1,562	0,640	0,947
98	5,7	4,3	2,3	56,373	29,52	39,46	72,51	2,397	1,255	1,678	3,083	1,648	0,672	1,027
99	2,9	4,8	3,9	54,288	28,42	38,00	20,95	2,381	1,247	1,667	0,919	1,507	0,731	1,031
100	3,2	2,4	3,6	27,648	14,48	19,35	12,76	0,748	0,392	0,523	0,345	4,133	0,747	1,419



(a) A2-Mode with various number of hidden neurons and hidden layers from base 100 hidden neurons



(b) A2-Mode with various number of hidden neurons and hidden layers from base 200 hidden neurons



(c) A2-Mode with various number of hidden neurons and hidden layers from base 400 hidden neurons

Figure B.2: Cross-validation results using different number of hidden neurons and hidden layers represented in boxplots. The x-axis represents the number of hidden neurons and hidden layers. For instance 150 150 stands for 150 hidden neurons in the first hidden layer and 150 hidden neurons in the second hidden layer. The y-axis represents the dice score. This results have been produced by utilizing the *A2-Mode*. Red colored boxed indicate scores for the central gland and respectively represent blue colored boxes scores for the peripheral zone. The edges of the boxes present the 25th and 75th percentiles of each score. Slashes within the box represent the median and dots represent the mean. Whiskers extend to the most extreme data points.

List of Figures

List of Tables

DECLARATION of Sources and Access

I, the undersigned, the author of this thesis, declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is provided.

I declare and understand that the Carinthia University of Applied Sciences will make my thesis available for use within the university library. I have been informed that the Carinthia Tech Institute is not liable for the misuse of its contents by third parties resulting from its reading. In particular, I have been informed that I am responsible for the registration of patents, trademarks or registered designs as well as any resulting claims.

Ich erkläre hiermit:

- dass ich die vorliegende Diplom/Masterarbeit selbstständig und ohne fremde Hilfe verfasst und noch nicht anderweitig zu Prüfungszwecken vorgelegt habe.
- dass ich keine anderen als die angegebenen Hilfsmittel benutzt, die den verwendeten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht und mich auch sonst keiner unerlaubten Hilfe bedient habe.
- dass die elektronisch abgegebene Arbeit mit der eingereichten Hardcopy übereinstimmt.
- dass ich einwillige, dass ein Belegexemplar der von mir erstellten Diplom/Masterarbeit in den Bestand der Fachhochschulbibliothek aufgenommen und benutzbar gemacht wird (= Veröffentlichung gem.§ 8 UrhG).

(Bad Reichenhall, October 21, 2013) (Hans-Peter Wieser)