1290

UNIVERSIDADE Đ
COIMBRA

Leonie Apitz

Identifying Prostate Cancer using Neural Network Image Segmentation

Thesis submitted to the Faculty of Science and Technology of the University of Coimbra for the degree of Master in Biomedical Engineering with specialization in Biomedical Instrumentation, supervised by Professor Paulo Menezes, Professor Cristina Tarín-Sauer, Peter Somers and Simon Holdenried-Krafft

February 2022

1 2 9 0

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Identifying Prostate Cancer using Neural Network Image Segmentation

Leonie Apitz

Thesis submitted to the Faculty of Sciences and Technology of the University of Coimbra
for the degree of Master in Biomedical Engineering with specialization in Biomedical Instrumentation

Supervisors:
Prof. Dr. Cristina Tarín-Sauer (ISYS)
Prof. Dr. Paulo Jorge Carvalho Menezes (ISR)

Coordinators:
Peter Somers, M.Sc.
Simon Holdenried-Krafft, M.Sc.

ISR - Institute of Systems and Robotics, University of Coimbra, Coimbra, Portugal
ISYS - Institute for System Dynamics, University of Stuttgart, Stuttgart, Germany

Coimbra, 2022

This work was developed in collaboration with:

ISR - Institute of Systems and Robotics, University of Coimbra

INSTITUTO DE SISTEMAS E ROBÓTICA
UNIVERSIDADE DE COIMBRA

ISYS - Institute for System Dynamics, University of Stuttgart

University of Stuttgart

University of Tübingen

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Don't let the thesis control your life.

Dedication

I'd want to express my gratitude to my family for their unwavering support and belief in me during this journey. To my mom, who never gave up on me, my grandparents, for all the caress and hour-long phone calls, my sister, for being my role model as I grew up, my brother, for teaching me what it means to be an adult, and Rafa, for the good food and all the dances, both of you have given me a place where I'm happy and that I'm proud to call home.

To my friends: Marisa and Filipe, my best friends, for always being proud of me in good and bad times; Claudia, my oldest friend, for always being there; Leonore, my childhood friend, who has been there over the years and whom I found living right next to me in Germany, a huge thank you for letting me be a ”parasite” in your home for all these months; Cruz, the person I still refer to as my roommate, thank you for your excellent musical taste and for always sharing it with me; Márcio, for all the support and medical insights; my ”university family”, my ”godmother” Catarina for the amazing materials that got me through the course and for all the support that I sometimes wasn't able to appreciate and my ”godchildren” Letícia, Rogério and *Ph* who were the sunshine in my life because they were always proud of my accomplishments, no matter how small; and my friends Bea, Rita, Lúcia, Joana and *Beta* for the adventures and never-ending conversations that helped with the distance.

Thank you to professor Rui Cortesão for opening my mind to subjects I had never considered before and helping me in my search for a place where I would have the unique chance to write my master thesis. Through you, I met my amazing supervisors, professor Cristina Tarín-Sauer, who accepted me in the blink of an eye, always showing kindness, and providing me with this incredible opportunity, and professor Paulo Menezes, who shared his patience, availability, and wisdom with me. I'll be eternally grateful to you. And last, but not least, a really special thanks to my advisors Peter Somers and Simon Krafft, who helped me with this project and were always there when I needed something; I learned a lot from both of you and you were more than just advisors, you were my friends and gave me a lot of strength in the time where most of my support was almost two thousand kilometres away. I couldn't have done it without you. Also, thank you to everyone else at ISYS and IS3L for the friendly conversations.

Thank you so much to everyone who has been a part of this adventure, and I hope to be able to repay you eventually!

Acknowledgments

The author would like to thank the respective database owners of “Prostate MRI and Ultrasound With Pathology and Coordinates of Tracked Biopsy (Prostate-MRI-US-Biopsy)”.

A special thanks also to the University of Tübingen for providing the necessary resources for the experiments.

Resumo

O cancro da próstata é um dos mais diagnosticados mundialmente em homens. O diagnóstico e o estadiamento desempenham um papel essencial para permitir a detecção precoce da doença e aumentar as hipóteses de sobrevivência e a eficácia do tratamento. Para diagnosticar e localizar o cancro, podem ser utilizadas várias técnicas de imagiologia médica, tais como radiografia, tomografia computorizada, ressonância magnética, ultra-som e imagiologia nuclear, cada uma fornecendo diferentes tipos de informação. Entre estas, a ressonância magnética multiparamétrica é a modalidade mais promissora capaz de avaliar a localização, tamanho e fase da lesão dentro da próstata. No entanto, devido à anatomia da próstata, a identificação destas lesões pode ser complicada. Por conseguinte, os modelos de aprendizagem computacional podem ser utilizados para melhorar o diagnóstico. Por esse motivo, o foco principal desta tese é expandir uma segmentação a nível de píxeis da próstata para identificar os tumores cancerosos.

Neste projeto, uma base de dados de 1151 pacientes com cancro da próstata do cancer imaging archive é utilizada para atingir este objectivo, a partir do qual foram seleccionados 707 pacientes. Quatro arquitecturas de redes neurais são propostas para realizar a segmentação em imagens de ressonância magnética. Um U-net simples é utilizado para a segmentação individual da próstata e do tumor. Um U-net cascata é utilizado para fundir as duas tarefas numa única rede. E as duas últimas são um híbrido destes dois modelos que aproveitam a informação do U-net simples numa arquitectura semelhante à do U-net cascata com foco na segmentação do tumor. O U-net simples obteve os melhores resultados para a segmentação da próstata, mas teve um desempenho inferior para a segmentação tumoral. Como resultado, as outras redes foram implementadas para superar esta dificuldade e, em última análise, melhoraram a capacidade de previsão.

Palavras-chave: cancro da próstata, resonância magnética, segmentação semântica, aprendizagem computacional, multiclasse

Abstract

Prostate cancer is one of the most common cancers diagnosed among men. Diagnosing and staging play an essential role in allowing early detection of the disease and increasing the chances of survival and the effectiveness of the treatment. To diagnose and localize cancer, several medical imaging techniques may be used, such as radiography, computed tomography (CT), magnetic resonance imaging (MRI), ultrasound (US) and nuclear imaging, each providing different types of information. Among these, multi-parametric magnetic resonance imaging (mpMRI) is the most promising modality capable of assessing the lesion's localization, size, and stage within the prostate. However, because of the prostates' anatomy, identifying these lesions can be complicated. Therefore, machine learning models can be used to improve the diagnosis. On that account, the main focus of this thesis is to extend a pixel-level segmentation of the prostate to identify cancerous tumors.

A database of 1151 prostate cancer patients from the cancer imaging archive is used to achieve this goal, from which 707 patients were selected. Four alternative state of the art neural network architectures are proposed to perform segmentation on MRI images. A simple U-net is used for individual prostate and tumor segmentation. A cascade U-net is used to fuse the two tasks in one network. And the last two are a hybrid of these two designs that leverages information from the simple U-net in a cascade U-net-like architecture focused on the tumor segmentation. The simple U-net obtained the best results for prostate segmentation but underperformed with tumor segmentation. As a result, the other networks were implemented to overcome this difficulty and ultimately improved the predictive capability.

Keywords: prostate cancer, magnetic resonance imaging, semantic segmentation, machine learning, multi-class

List of Acronyms

1D - One-dimensional

2D - Two-dimensional

3D - Three-dimensional

ANN - Artificial Neural Networks

ADC - Apparent Diffusion Coefficient

BCE - Binary Cross-entropy

BPH - Benign Prostatic Hyperplasia

CT - Computed Tomography

CZ - Central Zone

CNN - Convolutional Neural Networks

DCEI - Dynamic Contrast enhancement

DICOM - Digital Imaging and Communications in Medicine

DL - Deep Learning

DNN - Deep Neural Network

DRE - Digital Rectal Examination

DSC - Dice Similarity Coefficient

DW - Diffusion-weighting

DWI - Diffusion-weighting imaging

FCN - Fully Convolutional Networks

FFS - Feet First-supine

FL - Focal Loss

FTL - Focal Tversky Loss

GAN - Generative Adversarial Networks

GBCA - Gadolinium-based Contrast Agent

IoU - Intersection over Union

MHz - Megahertz

mpMRI - Multiparametric Magnetic Resonance Imaging

MR - Magnetic Resonance

MRI - Magnetic Resonance Imaging

MRSI - Magnetic Resonance Spectroscopy Imaging

PCa - Prostate Cancer

PI-RADS - Prostate Imaging–Reporting and Data System

PSA - Prostate-specific Antigen

PZ - Peripheral Zone

R-CNN - Regional Convolutional Networks

RF - Radio Frequency

ROI - Region of Interest

SAE - Stacked Auto-encoder

SBE - Section-based Evaluation

T - Tesla

T1-w - T1-weighted

T2-w - T2-weighted

TE - Echo Time

TI - Tversky Index

TR - Repetition Time

TRUS - Transrectal Ultrasound

US - Ultrasound

WCE - Weighted Binary Cross-entropy

WG - Whole Gland

List of Figures

2.1 Anatomical depiction of the prostate (Sagittal view). The left side of the image shows the location of the prostate inside the human body. The right side covers the anatomical zones of the prostate itself. 8

2.2 Architecture of an artificial neural network with one input layer, one output layer, and three hidden layers in between. 12

2.3 Architecture of an artificial neuron. 13

2.4 A local and global minimum. 15

2.5 Overfitting behavior. 15

2.6 Example of an architecture of a convolutional neural networks (CNN). 17

2.7 Convolution of an input matrix with the kernel or filter matrix. 17

2.8 Convolution with a 2 × 2 kernel, stride 2, and with and without padding. 18

2.9 Architecture of the U-net model used. Each blue box represents a multi-channel feature map with the number of channels shown on top. On the side of each level of boxes is the x-y size. Copies of feature maps are represented by white boxes. The arrows represent the various operations [45]. 20

2.10 Representation of GELU ($\mu = 0$, $\sigma = 1$), ReLU, and ELU ($\alpha = 1$). 21

2.11 Illustration of dice coefficient. 23

3.1 12-core digital template of the prostate (anterior frontal view). 25

3.2 A succinct example of most aspects of the dataset. A 3D representation of the MRI volume with the prostate volume in gray, the target in green, and the biopsy cores from the digital template and the ones for suspicious areas. Taken from [52]. 26

3.3 Example of an MRI image with the prostate mask on the left and the target mask on the right. 28

3.4 Layout for network training. 29

3.5 Types of data augmentation. 30

3.6 Example of an MRI image. 30

3.7 Example of an MRI image with translation, where translation=[height range, width range]. 31

3.8 Example of an MRI image with rotation (a, b) and with scaling (c, d). 31

3.9 Example of an MRI image with brightness. 31

3.10 Example of an MRI image with the maximum values for Gaussian noise (a), blurriness(b), sharpness (c), and contrast(d). 32

3.11 Architecture of the cascade U-net model. 33

3.12 Architecture of the first adaptation of the cascade U-net model. The red blocks represent the decoding path were the weights were frozen. 34

3.13 Architecture of the second adaptation of the cascade U-net model. The red blocks represent the decoding path were the weights were frozen. 35

4.1 Evolution of the dice coefficient for prostate segmentation using the simple U-net with geometric transformations. 37

4.2 Evolution of the dice coefficient for prostate segmentation using the simple U-net with geometric and intensity transformations. 38

4.3 Example of ground truth prostate masks from each set (a, b, c). Prostate segmentation prediction using a simple U-net and geometric transformations at epoch 63 (d, e, f). Prostate segmentation prediction using a simple U-net and geometric and intensity transformations at epoch 178 (g, h, i). 39

4.4 Evolution of the dice coefficient for target segmentation using the simple U-net with geometric transformations. 40

4.5 Evolution of the dice coefficient for prostate segmentation using the cascade U-net with geometric transformations. 41

4.6 Evolution of the dice coefficient for target segmentation using the cascade U-net with geometric transformations. 41

4.7 Example of ground truth prostate masks from each set (a, b, c). Prostate segmentation prediction using a cascade U-net and geometric transformations at epoch 200 (d, e, f). 42

4.8 Evolution of the dice coefficient for target segmentation using the first adapted cascade U-net with geometric transformations. 43

4.9 Evolution of the dice coefficient for target segmentation using the first adapted cascade U-net with geometric and intensity transformations. 43

4.10 Evolution of the dice coefficient for target segmentation using the second adapted cascade U-net with geometric transformations. 44

4.11 Example of ground truth target masks from each set (a, b, c). Target segmentation prediction using the second adapted cascade U-net and geometric transformations at epoch 185 (d, e, f). 45

C.1 Shrinkwrap of the US prostate volume onto the MRI prostate volume. 61

List of Tables

1.1 Summary of reviewed studies. 3

2.1 Overall PI-RADS scores. 10

2.2 Confusion Matrix. 22

3.1 Distribution of the UCLA score over all the targets. 27

3.2 Summary of used loss functions [42]. 36

4.1 Best results of the dice coefficient, sensitivity, specificity, and accuracy for prostate segmentation using a simple U-net with geometric transformations at epoch 63. . 38

4.2 Best results of the dice coefficient, sensitivity, specificity, and accuracy for prostate segmentation using a simple U-net with geometric and intensity transformations at epoch 178. 38

4.3 Best results of the dice coefficient, sensitivity, specificity, and accuracy for target segmentation using a simple U-net with geometric transformations at epoch 24. . 40

4.4 Results of the dice coefficient, sensitivity, specificity, and accuracy for prostate and target segmentation using a cascade U-net with geometric transformations at the last epoch. 41

4.5 Results of the dice coefficient, sensitivity, specificity, and accuracy for target segmentation using the first adapted cascade U-net with geometric transformations at the last epoch. 43

4.6 Best results of the dice coefficient, sensitivity, specificity, and accuracy for target segmentation using the first adapted cascade U-net with geometric and intensity transformations at epoch 479. 44

4.7 Best results of the dice coefficient, sensitivity, specificity, and accuracy for target segmentation using the second adapted cascade U-net with geometric transformations at epoch 185. 45

B.1 Results using different loss functions. 59

Contents

List of Acronyms **xvii**

List of Figures **xix**

List of Tables **xxiii**

1 Introduction **1**

1.1 Contextualization 1

1.2 Objectives 2

1.3 Current Machine Learning in Medical Imaging 2

1.3.1 Segmentation and detection of prostate cancer 3

1.4 Thesis Structure 6

2 Background Knowledge **7**

2.1 Prostate 7

2.2 Medical Imaging Methods 9

2.2.1 Ultrasound 9

2.2.2 Magnetic Resonance Imaging 9

2.3 Machine Learning and Neural Networks 11

2.3.1 Artificial Neural Networks 12

2.3.2 Convolutional Neural Networks 16

2.3.3 U-NET 19

2.3.4 Evaluation Metrics 22

3 Material and Methods 25

3.1 Dataset . . . 25

3.2 Pre-processing . . . 27

3.3 Image Generation . . . 27

3.4 Training . . . 28

3.4.1 Simple U-NET . . . 32

3.4.2 Cascade U-NET . . . 32

3.4.3 First adapted cascade U-NET . . . 33

3.4.4 Second adapted cascade U-NET . . . 34

4 Results 37

4.1 Simple U-NET . . . 37

4.1.1 Prostate segmentation . . . 37

4.1.2 Target segmentation . . . 39

4.2 Cascade U-NET . . . 40

4.3 First adapted cascade U-NET . . . 42

4.4 Second adapted cascade U-NET . . . 44

4.5 Final discussion . . . 45

5 Conclusions and Future Work 47

Bibliography 49

A Magnetic Resonance Imaging 55

B Loss functions 57

B.1 Results using different loss functions . . . 59

C Ultrasound data inclusion 61

1

Introduction

One of the leading causes of death in the world is cancer. In 2020, prostate cancer (PCa) was the second most diagnosed cancer and the fifth leading cause of mortality from cancer among men, being the first in 48 countries. It was also estimated that there were 1.4 million new cases and 375.000 deaths worldwide. However, there has been a rise in PCa incidence and a decline in mortality rates associated with treatment advancements and earlier diagnosis through screening [1].

Medical imaging techniques are used to diagnose and determine the stage of cancer. To help improve the diagnosis, machine learning models may be used. Therefore, this thesis aims implement a machine learning model to perform segmentation classification of magnetic resonance (MR) images of the prostate and thus identify prostate cancer.

1.1 Contextualization

Not much is known about the etiology of PCa, but several risk factors like advanced age, lifestyle, genetics, and ethnicity are associated with the development of this disease [2]. In the absence of symptoms or indications of disease, screening for PCa is accomplished by performing diagnostic tests to detect cancers at an early and treatable stage. However, these tests are not always reliable and lead to misdiagnoses, where the PCa was not detected or was incorrectly graded (overgrading and undergrading). Therefore, imaging techniques are used as they allow noninvasive detection, localization, grading, and staging of prostate carcinoma. They are also helpful for performing biopsies for histopathological tumor examination [3]. Furthermore, different imaging modalities provide different types of information [3]:

- **Radiography** - Used to evaluate metastatic disease but cannot be used to show localized disease in the prostate.
- **Computed tomography (CT)** - Also useful to detect metastatic disease but low value for local staging and showing intraprostatic pathology.
- **Magnetic resonance imaging (MRI)** - Combination of morphological images and functional techniques being therefore called multiparametric MRI (mpMRI) and is used in guidance for prostate biopsy, the local staging of biopsy-proven cancers, treatment planning, and posttreatment surveillance.

- **Ultrasound (US)** - Transrectal ultrasound (TRUS) is the most commonly used imaging modality of the prostate gland, measures the prostate volume, which is correlated with the estimation of PSA density and provides guidance for prostate biopsy.
- **Nuclear Imaging** - Used to assess the potential bone metastases from prostate cancer in patients with a higher value of prostate-specific antigen (PSA) or symptoms that may be associated with possible bone metastasis. Has a better specificity for cancer but a lower resolution.

Among these imaging modalities, as it is more critical to detect aggressive PCa rather than to detect all of the PCas, mpMRI is the most promising modality capable of assessing the localization, size, and stage of the lesion within the prostate when used in image-guided biopsies [4]. It is necessary to rely on spatial visualization of internal organs achieved by this high-resolution imaging method to evaluate cancerous regions and make prompt treatment choices. Even though radiological evaluation of imaging studies is still primarily visual and relies on domain knowledge and skill, there is a growing trend toward quantitative and volumetric disease evaluation for precision medicine. The pelvis remains one of the most challenging anatomical locations where recent auto-segmentation algorithms have been applied due to substantial intra-, and inter-patient soft-tissue variabilities [5].

1.2 Objectives

With the need of improving prostate cancer diagnosis it is necessary to rely on deep learning architectures. Therefore, this thesis aims to implement neural network image segmentation models to identify cancerous regions by extending initial segmentation of the prostate in MRI images.

For this purpose, it is necessary to find a dataset with a reasonable amount of patients with MRI acquisitions, review it, generate the images and corresponding prostate and tumor masks, and then implement a model. Given the available machine learning models, an excellent network for segmenting medical images is the U-net architecture. This type of architecture is an effective approach for segmenting different organs. However, to identify prostate cancer it is necessary to implement adaptations of this network to achieve better results.

1.3 Current Machine Learning in Medical Imaging

In the realm of computer vision, a rising number of deep learning (DL) methods have been presented in recent years. These techniques have proven effective and may be applied to organ pixel-level segmentation tasks in medical imaging since they evaluate in-depth features from these images, representing the image's structural information in depth. Convolutional neural networks (described in further detail in the next chapter) are a crucial component that results in a wide range of DL network topologies with a varying network structure, complexity, and implementation, resulting in task-dependent performance [6].

1.3.1 Segmentation and detection of prostate cancer

Over the years, several studies for pelvic organs segmentation, prostate gland segmentation, and tumor segmentation have been published. Kalantar et al. (2021) [5] made an overview of the DL-based pelvic segmentation algorithms currently used in pelvic cancers and concluded that there were 52 relevant studies on prostate segmentation. Some of these studies were reviewed to show the different prostate segmentation attempts. To evaluate the segmentation accuracy in these studies the dice similarity coefficient (DSC), intersection over union (IoU), and section-based evaluation (SBE) were used. The reviewed studies in this section are summarized in Table 1.1.

Table 1.1: Summary of reviewed studies.

Image Technique	Deep Learning Strategy	DL Network Dimension	Number of patients	Segmentation Evaluation Metrics	Year	Reference
CT	U-net	2D	85	Prostate (DSC: 0.88)	2018	[7]
T2W MRI	Encoder-Decoder structure with dense DDSPP	2D	150	Prostate (DSC:0.95)	2019	[8]
T2W, DW MRI	Dense multi-path CNN	3D	280	Prostate (DSC: 0.95)	2018	[9]
T2W, ADC MRI	U-net	2D	225	Prostate (median DSC:0.95), Central gland (median DSC:0.93), Peripheral zone (median DSC:0.86)	2019	[10]
T2W, DW MRI	Cascaded U-net	2D	163	Prostate (DSC:0.86), Peripheral Zone (DSC: 0.66)	2019	[11]
T2W, DW MRI	Mask R-CNN	2D	120	Prostate (DSC:0.88), Prostate tumor (DSC: 0.64)	2020	[12]
T2W, DW MRI	Mask R-CNN	2D	36	Prostate (IOU:0.84), Central gland (IOU: 0.51), Prostate tumor (IOU: 0.405	2019	[13]
T2W MRI	Multi-view CNN	2D	19	Prostate Tumor (DSC: 0.92, IoU: 0.67), Central gland (IoU: 0.65), Peripheral Zone (IoU: 0.59)	2019	[14]
T2W, DW MRI	Auto-Encoder	2D	21	Prostate Tumor (SBE: 0.89, sensitivity: 0.91)	2017	[15]
T2W, ADC MRI	Generative adversarial network	2D	152	Prostate Tumor (DSC: 0.41, sensitivity: 0.55)	2017	[16]

From the overview from Kalantar et al. (2021) [5], 12 studies used the CT imaging modality to perform pelvic organ segmentation and prostate segmentation, achieving DSC values for the prostate between 0.82 and 0.90. From these studies, 6 worked with a 2D network dimension, 5 with a 3D network dimension, and one with the combination of 2D and 3D. Also, 6 of them used a type of U-net (simple or with some modifications).

The one study that worked with the same dimensions (2D) and network used in this work (simple U-net) was Kazemifar et al. (2018) [7]. A 2D U-net was used to delineate the prostate, bladder, and rectum in the male pelvic CT images. A total of 85 prostate cancer patients were included in the study. The DSC, sensitivity, and positive predictive value were considered for evaluating the model. The bladder returned the highest average values for these three metrics (0.95; 0.95; 0.95) in comparison to the rectum and the prostate because of its high contrast, regular shape, and larger size. The prostate, as expected, returned the lowest average values for these metrics (0.88; 0.87; 0.92) because it is the most irregular and ambiguous shape to recognize.

From the overview from Kalantar et al. (2021) [5], 40 studies used the MRI imaging modality. 24 working with a 2D network dimension and 16 with a 3D network dimension. From these studies, 4 performed pelvic organ segmentation, 24 segmented the whole prostate gland, 7 segmented diverse areas of the prostate, and 4 completed tumor segmentation. DSC values between 0.64 and 0.95 were achieved for the whole prostate gland segmentation. The best DSC value of 0.95 was achieved in two studies [8, 9].

To accomplish prostate segmentation on T2-weighted (T2-w) MRI, Geng et al. (2019) [8] employed a 2D encoder-decoder architecture combined with dense dilated spatial pyramid pooling. This type of architecture provided a better understanding of the prostate's edge information. The method utilized in this study was tested on 150 patients and found to be more robust and accurate than other methods, indicating excellent performance in prostate segmentation. In the end, it returned a DSC value of 0.954.

On T2-w and diffusion-weighting (DW) MRI, To et al. (2018) [9] used a 3D deep dense multi-path CNN to accomplish prostate segmentation. They employed two datasets, 200 patients and 80 patients. In a quantitative examination, the DSC for these two datasets were 95.11 and 89.01, respectively.

Several studies used a type of U-net to perform segmentation of the prostate gland and its several zones [10, 11]. This is important because cancer is more common in the peripheral zone (PZ), whereas, in the central zone (CZ), it is more common to have benign prostatic hyperplasia (BPH) nodules which might seem like tumors on imaging [10].

Zabihollahy et al. (2019) [10] used a 2D U-net on T2-w and apparent diffusion coefficient (ADC) map MRI images to perform segmentation of zonal prostate anatomy. Their methodology involved the segmentation of the prostate whole gland (WG), central zone (CZ), and peripheral zone (PZ), where $PZ + CZ = WG$. They used a dataset of 225 patients, and on the T2-w images they achieved a median of DSC on test dataset for prostate WG, CZ, and PZ of $95.33 \pm 7.77\%$, $93.75 \pm 8.91\%$, and $86.78 \pm 3.72\%$, respectively.

Zhu et al. (2019) [11] used a cascaded 2D U-net to segment the prostate gland and peripheral zone. This strategy involved two steps. First is training one network to segment the prostate WG and then another network to segment the PZ. An image of the entire prostate gland area is formed and fed into the next U-Net in the second step, which further segments the PZ area based on the segmented output from the first step. A total of 163 patients were used in this study. The proposed method achieved the mean DSC of $92.7 \pm 4.2\%$ for the WG and $79.3 \pm 10.4\%$ for the PZ. The segmentation precision of this method is significantly higher for both WG and PZ segmentation than the classical U-net, which returned the mean DSC of 86.7 ± 9.9 for the WG and 66.8 ± 18.1 for PZ.

For tumor segmentation, there are only a few reported studies. However, from the overview from Kalantar et al. (2021) [5], the following studies were analyzed [12, 13, 14, 15]. Dai et al. (2020) [12] and Liu et al. (2019) [13] both used a a mask region-based CNN approach to segment prostate lesions. Alkadi et al. (2019)[14] used the multi-view CNN approach, and Zhu et al. (2017) [15] worked with hierarchical classification. Another analyzed study was from Kohl et al.

(2017) [16] where generative adversarial networks (GAN) were used.

Dai et al. (2020) [12] used a Mask R-CNN model to segment the prostate gland and intraprostatic lesions on mpMRI on a dataset of 120 patients. For the validation, the best DSC value they achieved for the prostate was 0.88, whereas, for the lesion detection, they obtained the best DSC of 0.64.

Liu et al. (2019) [13] used a two-stage approach to segment the prostate and the lesions, using a Mask R-CNN followed by a deep neural network (DNN). They used 36 patients for training. The model with the best DSC for the validation set is picked as the final model. To evaluate their model, they used the mean intersection over union (IoU). The model was tested on three different datasets. For one of the datasets, the results were compared with the ones from Alkadi [14]. For this dataset, the following results were obtained: an IoU of 0.843 for the WG, 0.781 for the CZ, 0.516 for the PZ, and 0.405 for the lesion.

Alkadi et al. (2019) [14] used a Multi-view CNN, a deep convolutional encoder-decoder architecture with a 3D sliding window technique. This technique allows 3D contextual spatial information given by the MRI series, which retains the 2D domains complexity while utilizing 3D information. This model is used to simultaneously segment the prostate gland, prostatic zones (PZ and CZ), and PCa on T2-w images. A total of 19 patients was used for training. This approach resulted in a DSC of 0.92 and IoU of 0.67 for prostate tumor, IoU of 0.65 for the CZ, and IoU of 0.59 for the PZ.

Zhu et al. (2017) [15] used the combination of a stacked auto-encoder (SAE) model with multiple random forest classifiers on T2-w images and some on DWI. In this work, 21 prostate cancer patients were used, and the evaluating metric was section-based evaluation (SBE), which is defined as the ratio of correctly identified sections by an automated procedure to the total number of sections in which the prostate is divided. It returned an SBE of 89.90 % for the high-level features, whereas, for the combination of high and low-level features, it returned 91.04 %.

Kohl et al. (2017) [16] presented an approach that is particularly advantageous for segmentation tasks on complicated structures with small datasets, GAN. A dataset of 152 patients was used, and an identical U-net type architecture was employed. In terms of DSC and sensitivity, 0.41 and 0.55 were returned for tumor segmentation, respectively.

Overall, considering the tumor size compared to the overall image size, the tumor segmentation approaches returned promising results and appeared to be a viable tool for radiologists in their clinical practices. However, these studies have several limitations as they employ small datasets and therefore may not be suitable for more extensive datasets.

As seen by Zhu et al. (2019) [11], using a cascade U-net to perform simultaneous tasks returned better results than performing the tasks individually. This has also been proved by other studies not related to the segmentation of the prostate. Gubins et al. (2020) [17] extended the U-net architecture by adding multiple decoding paths for each task and analyzed the outputs from each path as well as the connectivity between them. Their suggested architecture performed well on two image multi-output processing tasks (joint denoising and semantic segmentation of noisy RGB images). Therefore, showing the effectiveness of this type of architecture for multi-task problems. Thus, this cascade network is an excellent approach to be extended to detecting cancer.

1.4 Thesis Structure

This dissertation is organized into five chapters. In Chapter 1, the primary goals and motivation of the project and the related work are discussed. In Chapter 2, a brief overview of the theory underpinning the dissertation topics. First, a basic introduction to the prostate, followed by the medical imaging methods, with the main focus on MRI, and lastly, machine learning, where the fundamentals of this technique are represented. In Chapter 3, the methods used at various stages of project development are described. The chapter starts with a description of the dataset and then moves on to the various steps taken, from image processing methodology to the implemented machine learning model. In Chapter 4, the results from each implemented model for both the prostate and target segmentation are discussed in depth. In Chapter 5, this work's overall conclusions, findings, and a description of potential future improvements and research directions.

2

Background Knowledge

2.1 Prostate

The prostate is a dense and small fibromuscular gland present in male anatomy that sits beneath the bladder and surrounds the prostatic segment of the urethra. The rectum is behind the prostate, enabling clinicians to feel it by using their finger or to visualize it by using transrectal ultrasound [18].

It regulates urine flow, hormone synthesis, and, most importantly, prostatic fluid production. The prostatic fluid contains enzymes, zinc, and citric acid and contributes a significant fraction of the ejaculate. Prostate-specific antigen (PSA) is a crucial enzyme in the prostatic fluid that thins the ejaculate, preventing it from coagulating and allowing sperm to freely swim and pass through the urethra. During ejaculation, the prostate releases prostatic fluid into the urethra. It gets mixed with sperm cells from the testicles, seminal fluid from the seminal glands, and trace amounts of fluid from the bulbourethral glands. These make up the ejaculate, which is expelled out of the body afterward [18]. At that moment, the prostate also closes the opening between the bladder and urethra, simultaneously pushing the semen out faster and preventing semen from flowing back into the bladder. One of the hormones produced by the prostate, 5-alpha-reductase, converts testosterone, an androgen, into dihydrotestosterone, which is essential for the development of male secondary sexual characteristics. Therefore, when prostatic problems arise, urinary and sexual functions may be both affected [19].

Anatomically, the prostate gland is partitioned into three lobes: two lateral lobes (left and right) and a medial one [20]. Histologically, the gland is split into different zones: anterior, transition, central, and peripheral. The anterior zone also called the fibromuscular zone, comprises muscle and fibrous tissues. The transition zone is a small glandular area that encloses the urethra. The central zone (CZ) surrounds the ejaculatory ducts and sets the base of the gland. Moreover, the peripheral zone (PZ), the main zone, is where most of the glandular tissue is found, enclosing most of the central zone and a fraction of the distal part of the prostatic urethra. The gland is covered by a connective tissue capsule that contains muscle fibers [19]. An anatomical depiction of the prostate is shown in Figure 2.1.

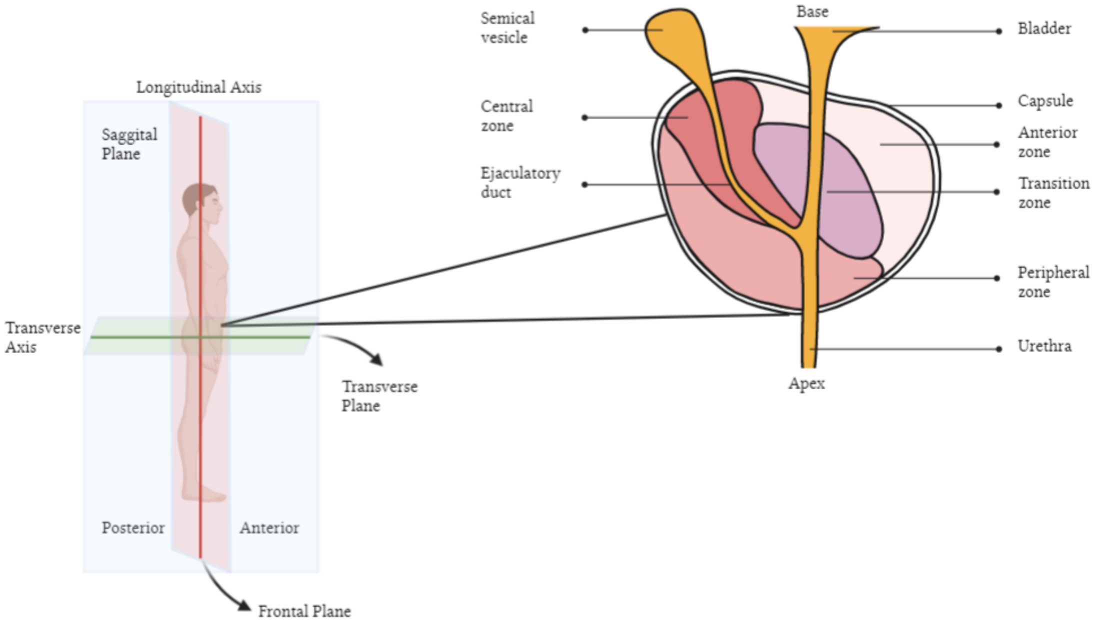


Figure 2.1: Anatomical depiction of the prostate (Sagittal view). The left side of the image shows the location of the prostate inside the human body. The right side covers the anatomical zones of the prostate itself.

Some prostate conditions may lead to urinary obstruction, sexual dysfunction, urinary tract infections, bladder stones, and, in the worst-case scenario, kidney failure [19]. At early stages, PCa may be asymptomatic with a frequently indolent course and may require minimal or even no treatment. However, the most common complaints are trouble with urination, elevated urination frequency, and nocturia that can also occur from prostatic hypertrophies. In addition, urinary retention and back pain may occur at a more advanced stage of the disease [2]. There are several diagnostic tests to detect cancers at an early and treatable stage, like the digital rectal examination (DRE), the PSA blood test, a glycoprotein normally expressed by prostate tissue and a random transrectal ultrasound (TRUS) guided biopsy [21].

If discovered after a biopsy, prostate cancer is graded using the Gleason score. The abnormality of cancer under a microscope determines this score. Less aggressive cancer resembles healthy tissue in appearance. On the other hand, more aggressive cancers have a more abnormal appearance and are more likely to grow and spread rapidly to other parts of the body. If cancer resembles normal prostate tissue, a grade of 1 is given, whereas, when cancer appears to be very abnormal, a grade of 5 is given. From 2 through 4, the grades in between have characteristics that fall somewhere in the middle of these two extremes. Most cancers are classified as grade 3 or higher; therefore, grades 1 and 2 are seldomly used. Depending on the zone of the prostate, this score varies. Therefore, a grade is appointed to the two areas that comprise the majority of cancer. The Gleason score is calculated by adding these two grades [22].

Cancer typically exhibits several structural, physiologic, and molecular changes that affect the prostate's anatomy. Modern medical imaging methods can visualize such changes, especially morphological changes. They are extremely powerful, versatile, and precise tools for evaluating relevant tumor characteristics. Furthermore, there is significant biological heterogeneity both between and within tumors, making them easier to locate [23].

2.2 Medical Imaging Methods

Non-invasive imaging techniques used to visualize the human body for diagnosis and therapeutic purposes are known as medical imaging methods. Of these, ultrasound (US) and magnetic resonance imaging (MRI) are most often used for examining the prostate and are outlined here in more detail.

2.2.1 Ultrasound

Ultrasound (US) is an imaging technique that uses high-frequency sound waves and relies on acoustic physics properties such as reflection, scattering, refraction, attenuation, and absorption to locate and characterize different tissue types. Medical US uses a transducer connected to a display monitor. The transducer sends an ultrasound pulse, sound waves in the range of millions of cycles per second (MHz), through the tissue. Then it receives the echoes back, which contain spatial and contrast information, forming a rapidly moving two-dimensional gray-scale image [24].

2.2.2 Magnetic Resonance Imaging

Magnetic resonance imaging (MRI) is a non-invasive imaging method that uses the properties of the protons inside of the human body and radio frequency (RF) waves to map specific aspects of the internal structure and function of the human body to produce detailed images. The RF wave, which is non-ionizing, is formed from the transition between the nuclear energy levels spin-flip and is the base to create high-quality cross-section images of the body. These are generated in any plane in the presence of a strong external magnetic field. Since magnetic waves in the radio frequency range surround us and do not damage tissues in passing, this imaging modality does not present hazards related to its exposure [25].

By exposing the body to a strong external magnetic field MR signals are collected (the in-depth explanation of this process is described in more detail in Appendix A). From the MR signals, MR images are calculated by applying a specific image sequence, comprising one or more RF pulses and magnetic field gradients. The nature of the acquired MR signal and the image contrast depend on the applied pulse, the gradient, and the relative time of signal acquisition. The specific image sequence is usually designed to assign weights to images based on a given relaxation time (i.e., T1 or T2 weighting) [26].

By using suitable imaging sequences, it is possible to get a comparison between healthy and malignant tissues from the difference in T1 and T2 relaxation times [26]. T1-weighted (T1-w) images are generated by a short repetition time (TR) between two RF pulses and a short echo time (TE) [25], and are useful for good anatomical delineation [26]. T2-weighted (T2-w) images are generated by a long TR between two RF pulses and a long TE [25], and are useful in differentiating pathological from normal tissue [26].

2.2.2.1 Multiparametric Magnetic Resonance Imaging

Multiparametric MRI (mpMRI) is a method that attempts to get an ideal three-dimensional (3D) prostate image [27]. In the mid-1980s, focus was on T1-w and T2-w sequences, but with the development of mpMRI technologies, more sequences were added [28], supplementing the initial ones. Like, dynamic contrast enhancement (DCEI), diffusion-weighting imaging (DWI), which includes the calculation of apparent diffusion coefficient (ADC) maps, and magnetic resonance spectroscopy imaging (MRSI). MRSI can be considered even though it is challenging and may require countless post-processing and input from MR physicists. Thus, the overall benefit becomes relatively small. However, using more sequences for the diagnosis of prostate cancer has shown better results than the T2-w images alone [29].

T1-w and T2-w sequences provide images based on the calculation of the water density in tissues arising from all MRIs. T1-w imaging is mainly used to identify post-biopsy bleeding in the prostate and seminal vesicles. It can also be employed to detect lymph node status and bone metastases, especially after administrating an intravenous gadolinium-based contrast agent (GBCA). T2-w imaging recognizes low signal intensity in tumor tissue. High-resolution imaging showed a clear border in the prostate capsule, especially where fat content exists outside the anatomical pseudo-capsule. Therefore, it becomes helpful to use prostate areas, intra-prostatic lesions, seminal vesicle (SV) infiltration, extra-prostatic expansion, and cancer staging assessment when dealing with this imaging sequence. Nevertheless, the T2-w sequence is not enough to detect cancers in the transition and central zones [27].

This imaging method, when used to evaluate suspected prostate cancer in prostate glands, uses a structured reporting scheme called Prostate Imaging–Reporting and Data System (PI-RADS). Predicting the probability of clinically significant cancer has to have either a Gleason score ≥ 7, a volume >0.5 mL, or an extra-prostatic extension. On DWI and T2-w, each lesion may be graded from 1 to 5 to indicate the possibility of clinically significant cancer, as well as by the presence or absence of dynamic contrast enhancement [30]. These scores are rated as represented in Table 2.1:

Table 2.1: Overall PI-RADS scores.

Overall PI-RADS score	**Probability of the presence of a clinically significant cancer**
1	very low
2	low
3	intermediate
4	high
5	very high
X	technical inadequacy or non-performance of a component of the exam

Depending on the zone of the prostate, transition zone, or peripheral zone, these scores contribute differently to the overall PI-RADS assessment. The T2-w score determines the PI-RADS assessment for the transition zone, occasionally adjusted by the DWI score. The DWI score determines the PI-RADS assessment in the PZ, which is sometimes affected by the presence of dynamic contrast enhancement [30].

2.3 Machine Learning and Neural Networks

In this section, the basics for understanding the implemented networks are described. First, an understanding of machine learning followed by the introduction of artificial neural networks and essential concepts: activation function, back-propagation, optimizer, and loss function. Then, convolutional neural networks are presented as they are imperative to process images. Subsequently, the U-net architecture is introduced. And last, the used evaluation metrics.

Machine learning includes a set of methodologies that allow a computer to learn from data and make predictions [31]. Regarding learning, Mitchell [32] defined it as: "A computer program is said to learn from experience E with respect to some task T and some performance measure P if its performance on T, as measured by P, improves with experience E."

To implement machine learning, three things are required: an input, in this case, images converted to multidimensional arrays, i.e., tensors, a target, in the same format, a mask of the expected output, also known as the ground truth, and a way to determine whether the algorithm is performing well, through the loss function. The loss function computes the distance between the prediction and the true target and is used as a feedback signal to adjust how the algorithm works. This process of adjustment is referred to as learning [33].

Machine learning is grouped into three separate categories [31]:

- **supervised learning**, in which the results of a dataset containing features associated with a label or target are reproduced;
- **unsupervised learning**, in which an unlabeled dataset is organized based on similarities/differences of its structure;
- **reinforcement learning**, in which a dataset interacts with an environment, iteratively learning and enhancing its performance.

A form of supervised learning used in this work is classification, one of the human activities' most common decision-making tasks. Classification problems arise when multiple classes are grouped based on a set of attributes, and an object must be assigned to a predefined class, i.e., it is required to select to which of the several categories a given sample belongs. [34]

2.3.1 Artificial Neural Networks

One of the sub-topics of machine learning are artificial neural networks (ANN), which are modeled after the biological neural networks that make up the brain. Its essential processing elements are called artificial neurons, or simply neurons or nodes [35], which are interconnected and able to perform certain computations on their inputs [34]. The basic architecture of an ANN is comprised of layers, an input, an output layer, and a given number of hidden layers in between [35]. The amount of hidden layers in a network represents the depth of the network and contributes to a data model. There are two kinds of networks: shallow networks and deep networks. The shallow network focuses on learning only one or two layers of data representations. On the other hand, deep networks have numerous layers and can learn all features in a single pass. As a result, machine learning operations are simplified, and performance for a wide range of issues is improved [33].

In classification problems that are not linearly separable, the hidden layers apply non-linear transformations to the input and send its output to the last layer [36], where errors or discrepancies between the predicted and the actual response are found in each node [35]. The node information is ultimately converted to the final classification. This framework is called a feed-forward network and requires continuous outputs rather than binary outputs (0 and 1) [36]. An example of an ANN structure is shown in Figure 2.2.

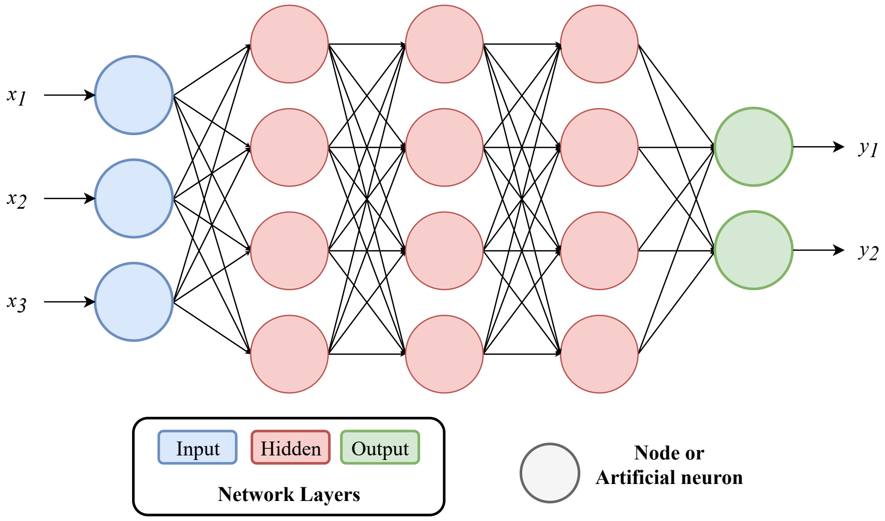


Figure 2.2: Architecture of an artificial neural network with one input layer, one output layer, and three hidden layers in between.

A model neuron receives information from multiple other neurons or external sources, weights each input, and adds them up together [36]. Considering the neuron shown in Figure 2.3, the scalar inputs x_i are multiplied by the scalar weights w_i, with i ranging from 1 to the number of inputs, to get the weighted sum $\sum_{i=1}^{n} x_{\mathrm{i}} w_{\mathrm{i}}$ and then, by adding an offset or bias b to it, the net input z of the activation function ψ is obtained. This function, also known as a transfer function, is primarily responsible for mapping the values it receives from the weighting to a specific range of values, returning the scalar neuron output a [37].

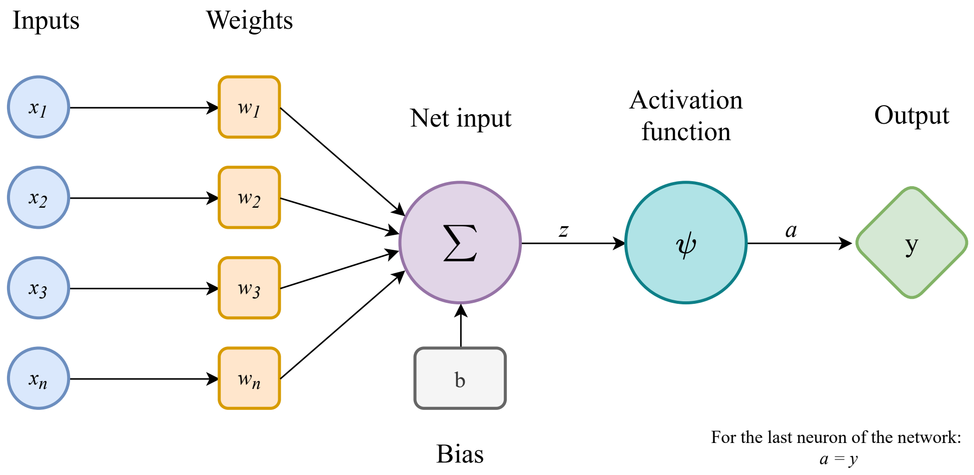


Figure 2.3: Architecture of an artificial neuron.

Thus, the neuron output is calculated as mentioned in Hagan et al. (1997) [37]:

$$a = \psi(b + \sum_{i=1}^{n} x_{\mathrm{i}} w_{\mathrm{i}}) \tag{2.1}$$

The last layer's neurons' outputs are termed the network's output [37], and therefore, $a = y$.

2.3.1.1 Activation function

The output of the neuron depends on the chosen activation function, which can be linear or non-linear [37]. In a linear activation function, the output signal is directly proportional to the input, making it simple to solve but limited in complexity due to the inability to learn and recognize complicated mappings from data. As a result, performance and power become restricted. Therefore, neural networks must learn and estimate linear functions and perform more intricate tasks, such as modeling complex types of data, e.g., images, video, audio, speech, and text. Among the non-linear activation functions, a commonly used one is the sigmoid activation function. This continuously differentiable and smooth function returns values ranging from 0 to 1 [38] and is often used in multi-layer networks which are trained using the back-propagation algorithm. This function is described as mentioned in Hagan et al. (1997) [37]:

$$\psi = \frac{1}{1 + e^{-z}} \tag{2.2}$$

For binary classification problems, as binary outputs are required, one can use the sigmoid function, interpreting output below 0.5 as class 0 and output above 0.5 as class 1 [36].

The learning ability of artificial neurons is achieved by adjusting the weights and biases according to the chosen learning algorithm [35]. This adjustment is based on a feedback signal and is known as training. To train a model, it is necessary to choose an optimizer, a loss function, and metrics to monitor during training, validation, and testing [33].

2.3.1.2 Back-propagation

The core of neural network training is back-propagation [39]. A learning algorithm that is used on a feed-forward network and operates from the output to the input layer [36]. It is the process of fine-tuning the weights of a neural network based on the preceding epoch's loss. Weight tuning ensures lower error rates, boosting the model's generalization and making it more robust [39].

To begin, all of the network's weights are set to small random values, and then the output for each input is returned. Next, the error, defined as loss or cost, is calculated between the obtained and predicted output, also called target. The overall error is estimated by adding all of these numbers and demonstrates the model's generalization ability [36].

To show how the back-propagation algorithm works, the sum of errors for the squared difference between the obtained output $y(x_n,w)$ and the target t_n will be considered for the loss function and is calculated as mentioned in Bishop (2006) [40]:

$$L(w) = \frac{1}{2}\sum_{n=1}^{N} ||y(x_n,w) - t_n||^2 \tag{2.3}$$

where, $y(x_n,w)$ is the output vector (vector of activations from the network), x_n the set of input vectors with $n = 1,...,N$ and w the weights, and t_n is the corresponding set of target vectors.

Back-propagation's purpose is to compute the partial derivatives of the loss function L regarding any weight w or bias b in the network $(\frac{\partial L}{\partial w})$ and $(\frac{\partial L}{\partial b})$ [41]. It attempts to decrease L by using a numerical optimization approach known as gradient descent, which simplifies the math and gives the method its name. Whenever an example is given, back-propagation adjusts the weights and thresholds, minimizing the error over time. This procedure is countlessly repeated until the error converges to a minimum. [36]. The gradient descent is expressed by the following equation [40]:

$$w^{\tau+1} = w^{\tau} - \eta\nabla L(w^{\tau}) \tag{2.4}$$

Where the parameter $\eta > 0$ is the learning rate, that determines the size of the step, τ is the time step, and ∇L is the gradient, derivative of a tensor operation, of L, which is evaluated in each step.

2.3.1.3 Optimizer

Sometimes it is possible to calculate the minimum analytically. However, considering a very high number of parameters involved, it is necessary to use an optimizer, which defines how the gradient of the loss will be utilized to update parameters [33]. To optimize and learn the objective, deep learning algorithms use the stochastic gradient descent approaches and their variants [42]. Considering the mini-batch stochastic gradient, it takes individual batches of training samples and corresponding targets, performs a forward pass, the computation of the loss, a backward pass, and finally, moves the parameters a little in the opposite direction from the gradient using the learning rate. The learning rate acts as a “speed” factor, helps to find the global minimum, and reduces the loss a bit, and therefore it is crucial to choose a reasonable value for it. Looking at Figure 2.4, there is a local minimum at a certain parameter value, and moving in either direction around that point would result in the loss increasing. If the learning rate is too low, it will take a long time to descend along the curve and may get stuck at a local minimum; if it is too high, the updates may result in a completely random point on the curve [33].

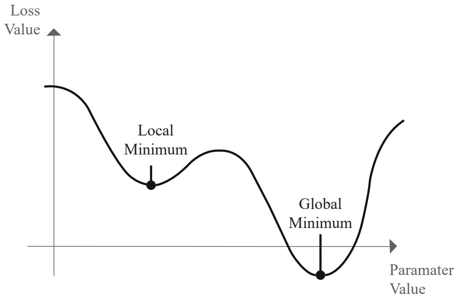


Figure 2.4: A local and global minimum.

However, it does not always find the global minimum of the error for a non-convex function. Therefore the weights’ initial values affect the training outcome. One problem, in particular, sticks out: overfitting [36], represented in Figure 2.5.

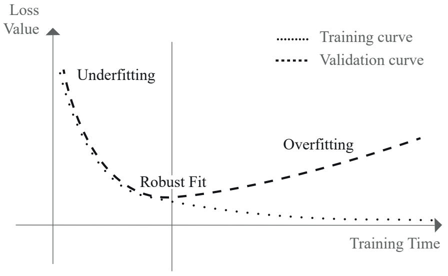


Figure 2.5: Overfitting behavior.

This happens because the weights are adjusted to account for the quirks of the training instances, which are not representative of the entire distribution of cases. When it comes to fitting such quirks, the large number of weight parameters in ANN offer several degrees of freedom [32]. Providing a set of validation data to the algorithm, in addition to the training data, provides an approximation of the network's ability to classify inputs it has not been trained on [36] and allows the algorithm to plot the two curves represented in Figure 2.5. The individual sets must not contain examples that are very similar to provide an unbiased estimate [36].

At the beginning of training, the loss of the training data is correlated with the loss of validation data. During this time, the model is considered to underfit. The network has not yet represented all key patterns in the training data. However, after a certain number of iterations on the training data, generalization ceases to improve, validation metrics halt and ultimately degrade: the model is beginning to overfit [33].

In this work, an Adam based optimizer is used. It is one form of the stochastic gradient descent technique and is a momentum-based optimizer. Individual adaptive learning rates for distinct parameters are computed using estimations of the first and second moments of the gradients, thus the name Adam. This approach only needs a first-order gradient and requires very little memory. It combines the benefits of two techniques, AdaGrad and RMSProp. The stepsize hyperparameter essentially limits its stepsizes. It does not demand a stationary objective, works with sparse gradients, and naturally does step size annealing. As a result, the magnitudes of parameter updates are invariant to gradient rescaling [43].

2.3.1.4 Loss function

The loss function must cover even the most extreme edge situations to learn an objective reliably and faster. As a result, the loss function is crucial for creating sophisticated image segmentation-based deep learning architectures since it influences the algorithm's learning process [42]. For multiclass classification, the most common loss function used is cross-entropy [44]. However, in this work, the loss function used is based on the dice coefficient, described in section 2.3.4 and the loss function itself is defined in Chapter 3.

2.3.2 Convolutional Neural Networks

Convolutional neural networks (CNN) are a type of neural network for processing data with a grid pattern, such as images, which can be thought of as a 2D grid of pixels. The term "convolutional neural network" refers to the network's use of the convolutional mathematical procedure [1], which makes up at least one layer of the network [31]. They are comprised of three types of layers (or building blocks): convolution, pooling, and classification, which are usually fully connected layers [44].

This network's convolutional layer can alternatively be described as a small number of complex layers. And therefore, consists of three phases: various convolutions are executed in parallel to create a group of linear activations, each linear activation is then passed through a non-linear activation function, such as the rectified linear activation function, known as the

[1] This is not the same convolution operation as is used in statistics.

detector stage, and then the pooling function is used to alter the layer's output further [31]. An example of a CNN structure is shown in Figure 2.6.

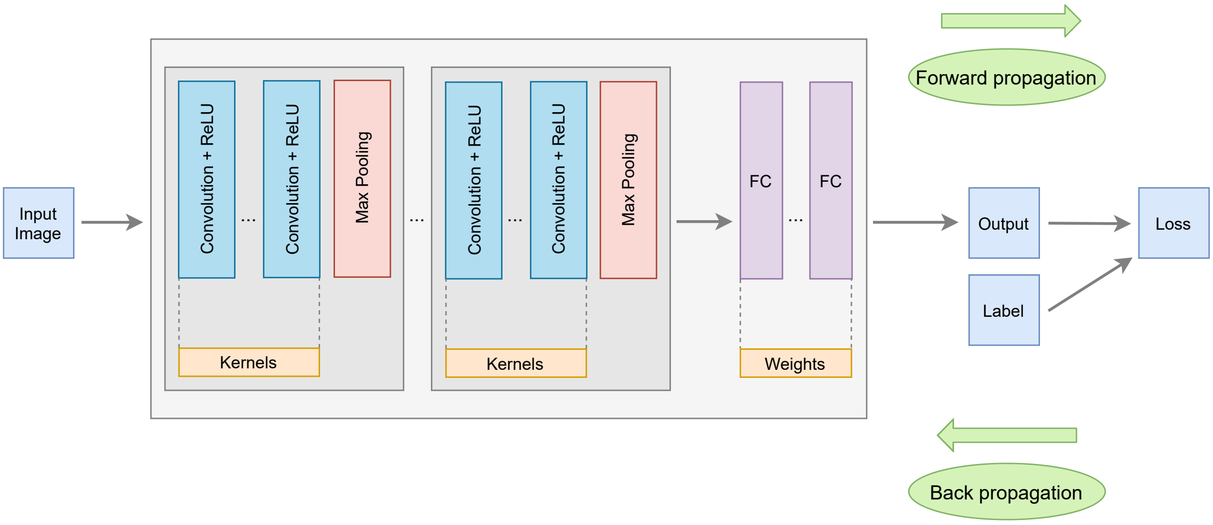


Figure 2.6: Example of an architecture of a convolutional neural networks (CNN).

Convolution is used for feature extraction where a kernel, a multidimensional array of parameters, is applied over the input, a multidimensional array of data, called a tensor, where the output of the operation between these two is referred to as a feature map [31, 44]. This mathematical operation is represented in Figure 2.7, where the convolution between an input matrix with dimensions $I_h \times I_w \times C_{in}$ and a kernel with dimensions $f_h \times f_w \times C_{in}$ returns the feature map with dimensions $(I_h - f_h + 1) \times (I_w - f_w + 1) \times C_{out}$. Here, h stands for height, w stands for width, and C_{in} and C_{out} stand for the number of input and output channels, respectively.

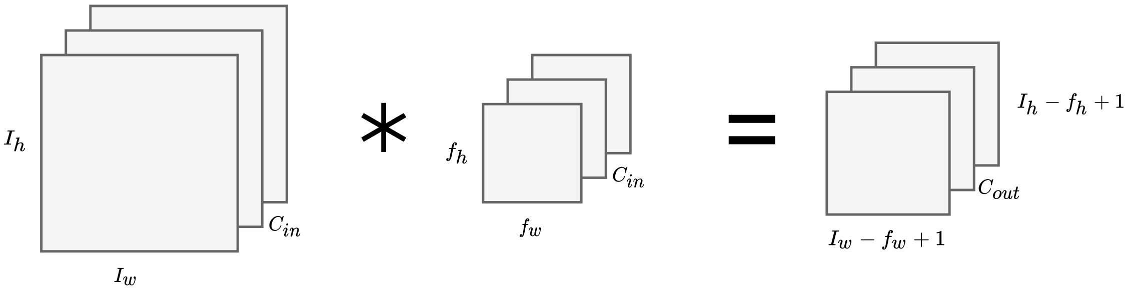


Figure 2.7: Convolution of an input matrix with the kernel or filter matrix.

Considering a 2D convolution, the output activation function from one feature map in the m-th row and n-th column with shape $o_h \times o_w$ is given by the following equation [41]:

$$o_{m,n} = \psi\left(b + \sum_{j=0}^{f_h-1}\sum_{k=0}^{f_w-1} w_{j,k}a_{(m+s_h+j),(n+s_w+k)}\right) \tag{2.5}$$

Where all indices are zero-based, b is the kernel's bias, $a_{x,y}$ is the activation from the associated input neuron, and s is the stride length where the subscript h or w denotes the height and width, respectively.

The size and number of kernels responsible for creating an arbitrary number of feature maps representing various input tensor features are two essential hyperparameters that characterize this operation. Because this operation hinders the center of each kernel from overlapping the input tensor's outermost element, a padding technique, usually zero padding, is applied to the input tensor, which adds rows and columns of zeros to the input tensor to prevent the output feature map from having a smaller width and height than the input tensor. Another component of the convolution operation is the stride, which describes the step width of a kernel sliding over the input. Figure 2.8 illustrates an example of the convolution operation with stride and with and without padding. A stride of 1 is the most usual choice. However, a stride greater than one is occasionally used to accomplish feature map downsampling. An alternative approach for downsampling is a pooling operation [44].

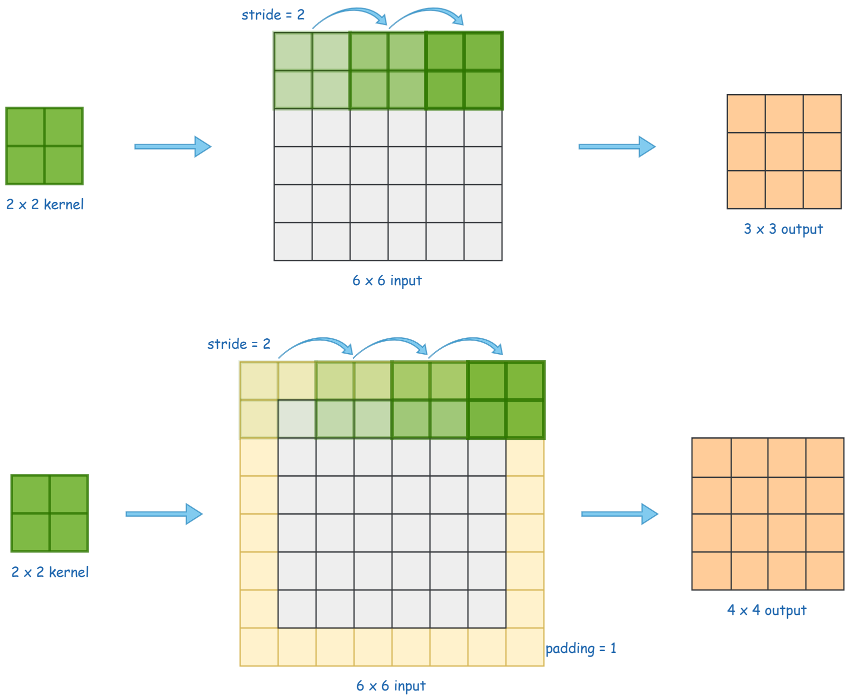


Figure 2.8: Convolution with a 2×2 kernel, stride 2, and with and without padding.

Pooling performs a standard downsampling operation on the feature maps [44] and helps in establishing translation invariance, which means that when a small amount translates the input, the values of most of the pooled outputs remain unchanged while pooling over the outputs of independently parametrized convolutions allows the features to learn which transformations they should be invariant to [31]. Pooling operations are divided into two types: max and average. The most frequent is max pooling, which chooses areas from the input feature maps, outputs the largest value in each region, and discards the remainder. As a result, the pooling layer contains the method's hyperparameter as well as size, stride, and padding, much like the convolution layer [44].

The final convolution or pooling layer is connected to one or more fully connected layers, also known as dense layers. A learnable weight connects each input and output. A subset of fully connected layers map the last layer's output feature mappings to the network's final outputs, such as the probabilities for each class in classification tasks, by flattening them, that is, converting them into a one-dimensional (1D) array of numbers (or vector). The number of output nodes in this subset's final layer is generally the same as the number of classes [44].

Convolutional networks are commonly used for classification tasks, using a single class label as the output to an image [45]. An image comprises multiple pixels, which define different elements in the image when combined. The classification task of these pixels is known as image segmentation [42]. However, in many visual tasks, particularly in biomedical image processing, the intended output should contain localization, which means that each pixel should have a class label assigned to it. Furthermore, thousands of training images in biomedical tasks are generally out of reach. A solution to the amount of images needed with an increased segmentation accuracy is the U-net model [45].

2.3.3 U-NET

The U-net model's architecture is based on a modified and expanded fully convolutional network (FCN), allowing it to work with fewer training images and generate more accurate segmentation [45]. FCN are a type of network used for semantic segmentation that consists of a contracting and an expansive path, also called encoder and decoder. In contrast to standard CNN, it employs locally connected layers such as convolution, pooling, and upsampling rather than fully connected layers (dense layers), carried out using 1x1 convolutions. As a result, the FCN can handle a wide range of input image sizes and produce outputs with the same spatial dimensions thanks to its locally connected layers. Furthermore, as the dense layers are not employed, there are fewer parameters, resulting in faster network training [46]. The basic idea behind this network is to add layers to a regular contracting network by substituting upsampling operators for pooling operators [45]. By adding these downsampling processes to a convolution sequence, the encoder converts a supplied segmentation to a lower-dimensional encoding and classifies each pixel into a particular category. In contrast, the decoder attempts to reproduce the original segmentation from the encoding by upsampling the low-resolution categorized pixels and projecting the features onto a plane, resulting in a high-resolution segmented image [13, 47, 48]. Based on this knowledge, a subsequent convolution layer can learn to build a more precise outcome [45].

For the U-net, the upsampling part of the network contains many feature channels, allowing the network to transfer context information to higher resolution layers. This is a significant change in the architecture in comparison to the FCN. As a result, the expanding and contracting paths are nearly symmetrical, resulting in a u-shaped structure. The network has no fully connected layers and only utilizes the valid component of each convolution, i.e., the segmentation map only contains pixels that have the whole context in the input image. This approach allows for the smooth segmentation of arbitrarily big images by employing an overlap-tile technique [45]. One of the essential characteristics of the U-net is skip connections. Furthermore, these connections allow the decoder's feature maps to integrate more low-level features, boosting the model's segmentation accuracy [13].

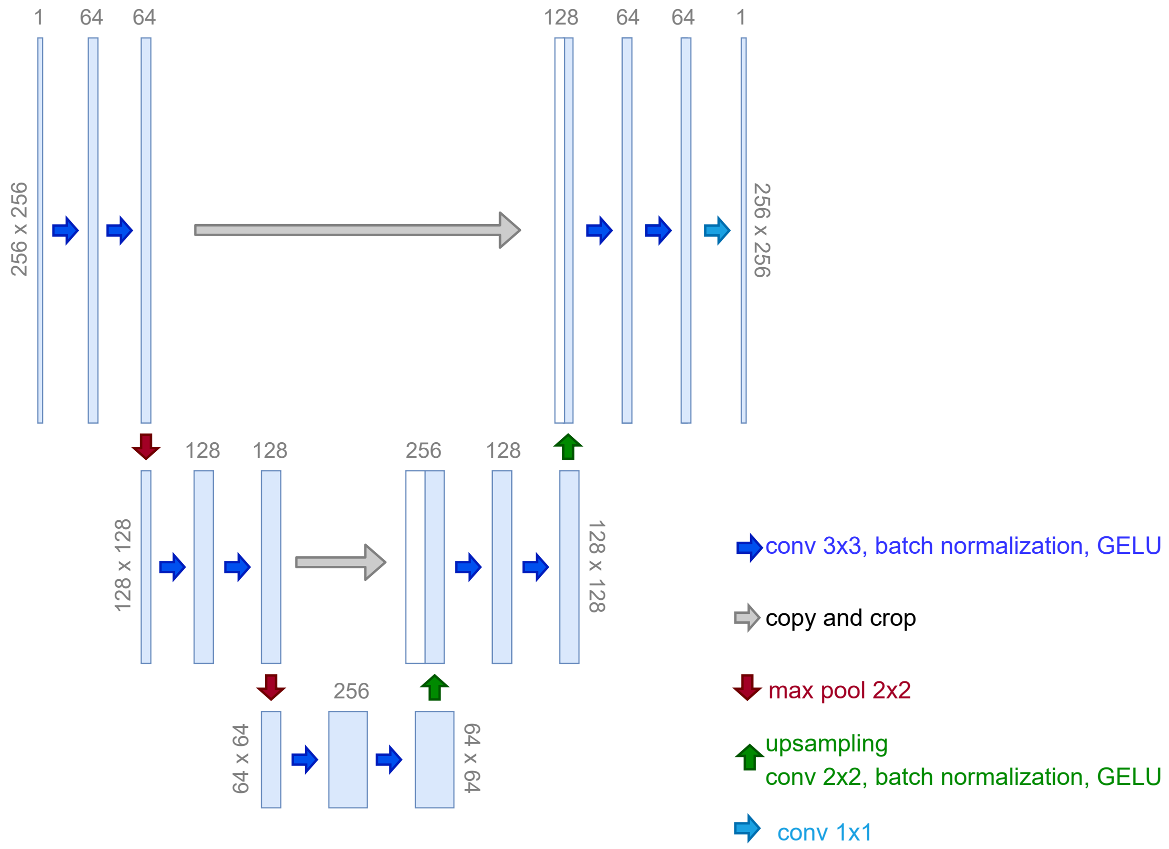


Figure 2.9: Architecture of the U-net model used. Each blue box represents a multi-channel feature map with the number of channels shown on top. On the side of each level of boxes is the x-y size. Copies of feature maps are represented by white boxes. The arrows represent the various operations [45].

The networks' architecture represented in Figure 2.9 comprises a contracting path (on the left), which follows the traditional architecture of a convolutional network, with three downsampling levels, and an expansive path (on the right), with three upsampling levels. The contracting path has two 3x3 convolutions. Each followed by a batch normalization and a Gaussian Error Linear Unit (GELU) activation per downsampling level and a 2x2 max pooling operation with stride 2 for downsampling, resulting in a doubled number of feature channels. Next, the feature map is upsampled using reflective padding for the expansive path, followed by a 2x2 convolution, batch normalization, and GELU activation. The feature map from the contracting path is then concatenated with the corresponding cropped feature map, followed by one 1x1 convolution, batch normalization, and GELU activation. Cropping is necessary for every convolution because of the loss of border pixels. A 1x1 convolution transforms each 64-component feature vector into the appropriate number of classes at the final layer. In total, there are 13 convolutional layers in the network [45].

Batch normalization is a simple and elegant technique to re-parameterize nearly any deep network [31]. It decreases the change in internal neuron distributions during training while also significantly speeding it up [49]. By standardizing the mean and variances of each neuron, it re-parameterizes the model to guarantee that some neurons are always standardized while allowing connections between neurons and non-linear statistics of a single node to change [31]. Furthermore, batch normalization improves gradient flow, which reduces gradients' dependency on parameter scale or initial values, allowing for higher learning rates without the risk of divergence. Therefore, batch normalization also enhances model consistency while reducing the requirement for Dropout [49]. Batch normalization, especially for convolutional networks and networks with sigmoidal nonlinearities, can have a significant impact on optimization performance [31].

The Gaussian Error Linear Unit (GELU) is a high-performance neural network activation function inspired by a combination of dropout, zoneout, and ReLU features. While the ReLU and dropout both affect the output of a neuron, the ReLU deterministically multiplies the input by zero or one, and the dropout stochastically multiplies by zero. The zoneout works as a regularizer, multiplying inputs by one stochastically. When these functionalities are combined, the input is multiplied by one or zero, with the values of this zero-one mask chosen stochastically while still depending on the input [50]. The resulting function GELU, as well as ReLU and ELU, a modification of the ReLU function that permits a nonlinearity similar to ReLU to negative output values, are represented in Figure 2.10.

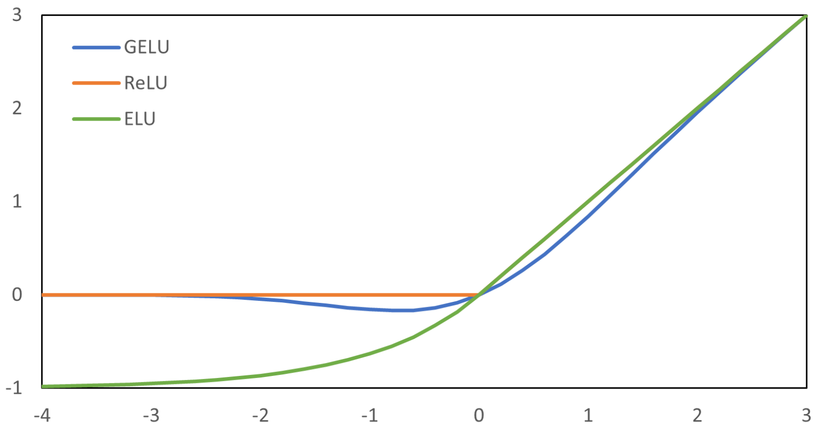


Figure 2.10: Representation of GELU ($\mu = 0$, $\sigma = 1$), ReLU, and ELU ($\alpha = 1$).

Since neuron inputs tend to follow a normal distribution, especially with batch normalization, the input x can be multiplied by a cumulative distribution function of the standard normal distribution, Bernoulli $\Phi(x)$, where $\Phi(x) = P(X < x), X \sim N(\mu,\sigma^2)$, with $\mu = 0$ and $\sigma = 1$. In this case, as x decreases, inputs have a higher probability of being "dropped", therefore the transformation performed to x is stochastic yet dependent on the input. As the cumulative distribution function of a Gaussian is frequently calculated with the error function the Gaussian Error Linear Unit (GELU) is defined as:

$$GELU(x) = xP(X < x) = x\Phi(x) = x \cdot \frac{1}{2}\Big[1 + \text{erf}(x/\sqrt{2})\Big] \tag{2.6}$$

2.3.4 Evaluation Metrics

The evaluation of automatic detection techniques is determined by the type of problem and the data type. In this work's binary classification problem, the samples will fall into two categories: positive (P) or negative (N). The classifier will predict a class for each input sample. Therefore, there are four key terms to remember:

- **True Positives (TP):** Predicted P, and the actual result was likewise P.
- **True Negatives (TN):** Predicted N and got N.
- **False Positives (FP):** Predicted P, but the actual result was N.
- **False Negatives (FN):** Anticipated N but got P instead.

For the topic at hand, binary classification, several essential performance metrics for evaluating the models' performance and competence as class predictors can be generated from these four key terms. Moreover, the most commonly used for this problem are the confusion matrix, accuracy, sensitivity, specificity, precision, recall, and F1-Score. The formulation of these is denoted below:

1. **Confusion Matrix:** summarizes the model's performance by displaying the correct and incorrect classifications for each class and is represented in Table 2.2

Table 2.2: Confusion Matrix.

		Predicted Class	
		Positive	**Negative**
Actual Class	**Positive**	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

2. **Accuracy:** fraction of correctly classified predictions is calculated by averaging the values along the ”main diagonal” of the confusion matrix, i.e

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

3. **Sensitivity:** With regard to all positive data points, the sensitivity (True Positive Rate) is the fraction of positive data points that are accurately considered positive.

$$Sensitivity = \frac{TP}{FN + TP} \tag{2.8}$$

4. **Specificity:** With regard to all negative data points, the specificity (True Negative Rate) is the fraction of negative data points that are accurately considered negative.

$$Specificity = \frac{TN}{TN + FP} \tag{2.9}$$

5. **Precision:** Measures the ability to identify the correct positive predictions of relevant objects.

$$Precision = \frac{TP}{TP + FP} \tag{2.10}$$

6. **Recall:** Measures the ability to identify the correct positive predictions of Ground Truth objects, also known as sensitivity.

7. **F1-Score:** Harmonic Mean between precision and recall

$$F1 = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}} \tag{2.11}$$

One of the most intuitive performance metrics is accuracy. This metric, however, is misleading for this work's binary classification problem because the data is imbalanced, and a poor classifier may return good results by ignoring the minority class. Sensitivity and specificity are standard metrics used to evaluate a model's capacity to predict the presence or absence of cancer. Another handy metric when dealing with imbalanced datasets is the F1-Score. It combines precision and recall to determine how accurate and robust a classifier is. High precision but low recall provides a very accurate result but overlooks several complex incidents to classify. The F1-Score has a range of [0,1], and the higher the score, the better the model's performance [51].

A metric that is equivalent to the F1-Score is the dice coefficient. The dice coefficient is used to measure the similarity between two samples and can be described as two times the area of overlap divided by the total number of pixels in the image [51], represented in Figure 2.11.

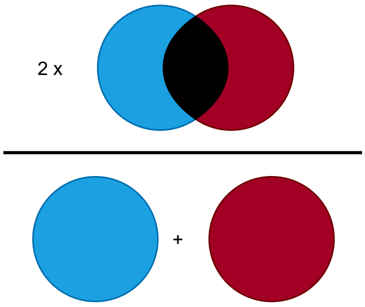


Figure 2.11: Illustration of dice coefficient.

Therefore, the equation that defines this metric is written as:

$$Dice = \frac{2 \times intersection}{union + intersection} \tag{2.12}$$

By substituting the precision and recall in the equation of the F1 score and knowing that $union = TP + FN + FP$ and $intersection = TP$, we get the same equation. Therefore, proving that these two metrics are equivalent. The dice coefficient will be the metric used during the rest of the work.

3

Material and Methods

3.1 Dataset

The neural network was trained using a dataset of US and MRI images from The **Cancer Imaging Archive** [52]. Patients with high PSA and abnormal imaging findings suspected of having prostate cancer were systematically collected. The research included any patient who gave their permission and had a routine, standard-of-care prostate biopsy at the UCLA Clark Urology Center. Therefore, resulting in a dataset of 1151 patients. The Artemis biopsy system was used to track these prostate cancer patients' biopsy procedures, several of which included image fusion with MRI targets.

The non-rigid registration (e.g., "fusion") occurred after a 3D transrectal ultrasound scan. It was done between real-time ultrasound and preoperative MRI, allowing biopsy cores to be sampled from MRI regions of interest. In most cases, systematic biopsy cores were also sampled utilizing a 12-core digital template [52], as shown in Fig 3.1.

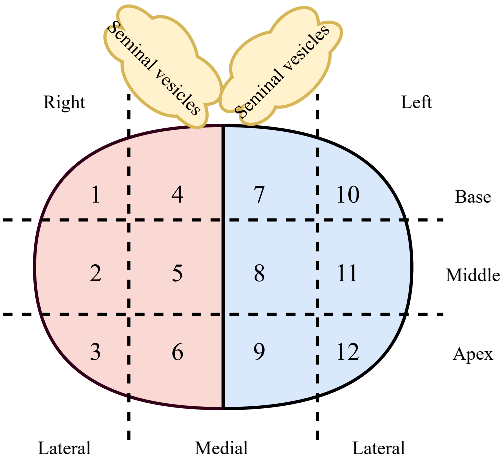


Figure 3.1: 12-core digital template of the prostate (anterior frontal view).

MRI targets were established using mpMRI, which were subsequently scored using the UCLA score, which is similar to PI-RADS. The only sequence included in this dataset is T2-w MRI, which was utilized to trace ROI outlines. A 3 Tesla Trio, Verio, or Skyra scanner (Siemens, Erlangen, Germany) was used for MR imaging. A transabdominal phased array was employed in all cases, and in a subset of patients, an endorectal coil was used. For the US, the end-fire probe was rotated 200 degrees along its axis to get 3D images, which were then interpolated to resample the volume with isotropic resolution [52].

The Artemis system tracked and recorded targeted and systematic core positions relative to the Ultrasound scan using encoder kinematics of a mechanical arm. Primarily, the ultrasound and MRI biopsy data include the coordinates of these locations and STL files. This type of file format is commonly used in three-dimensional modeling software packages. It defines a three-dimensional object's prostate and target surface geometry [52]. A succinct example of most aspects of this dataset are represented in Figure 3.2.

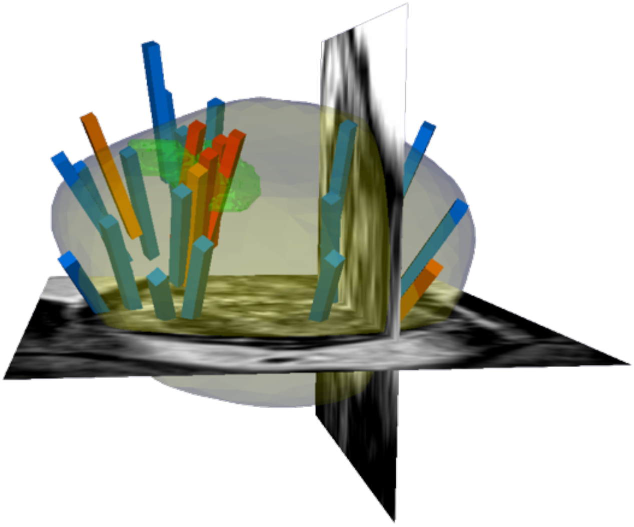


Figure 3.2: A succinct example of most aspects of the dataset. A 3D representation of the MRI volume with the prostate volume in gray, the target in green, and the biopsy cores from the digital template and the ones for suspicious areas. Taken from [52].

The biopsy data and the MRI target information are saved in two different spreadsheets [52]. The first contains coordinates of each biopsy core from the 12-core digital template, coordinates of some targets or prior positives for the MRI and US, the series number from each method, and the patient number. The second, with the target data, contains the series number from each method, the patient number, the UCLA score, the ROI volume, and the number of the target. The target is characterized as a suspicious or tumorous region within the prostate. The UCLA score is similar to the PI-RADS and is used to grade the lesion. In addition, there is a corresponding STL file with the patients' number, target number, and series number for each target number.

Particular examples were revised due to a lack of MRI coordinates or other data issues, such as missing image orientation position, difficulties reading the STL file, or inability to load the data file at all. As a consequence, 707 patients were used to train the network. For the ultrasound, 670 patients fulfilled the mentioned qualitative requirements.

3.2 Pre-processing

The MRI of each patient is divided into multiple slices, which are saved as enumerated Digital Imaging and Communications in Medicine (DICOM) files in the corresponding directory. In contrast, the US of each patient is saved in an individual DICOM file. After the revision of the dataset, the pre-processing stage in MATLAB reads each DICOM file. Then, it creates the volume and DICOM header, which are stored in a structure that includes the necessary information to comprehend how the image is generated, the resolution and size of the image, and how the volume should be created.

For the MRI, to store each slice in the volume the *InstanceNumber* parameter in the DICOM header is utilized. The final volumes for all patients conform to a specific location, which corresponds to feet first-supine (FFS). The dimensions of the images and the space between them are retrieved from the DICOM header so that a function may be used to establish the relationship between the intrinsic coordinates and the world coordinate system, which is then used to produce a grid of coordinates. This grid comprises the sizes of the pixels in each dimension (x, y, and z) and the associated limits. It is essential because it enables the prostate and target STL files to be specified in the correct space once they have been read. Afterward, the generated grid of the prostate STL file and the target are saved. When there are several targets for the same MRI volume, they are merged and stored.

3.3 Image Generation

The individual images and corresponding prostate and target masks are created by reading MATLAB's produced files from the prostate and the targets. For the 707 patients, 706 patients had targets associated. In total, there were 1449 targets, and the distribution of the UCLA score for these targets is represented in Table 3.1.

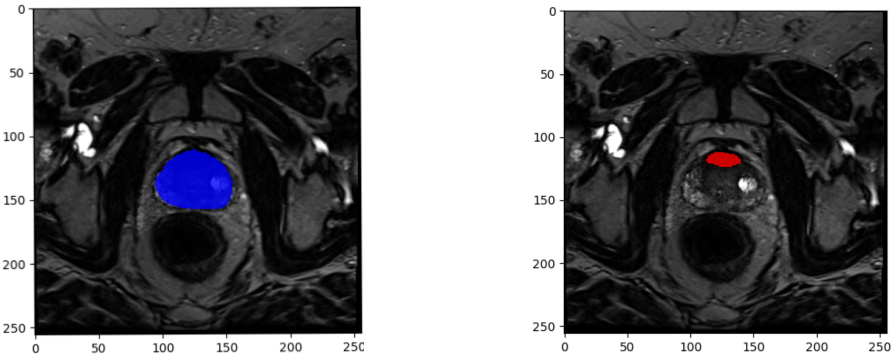
Table 3.1: Distribution of the UCLA score over all the targets.

UCLA Score	Number of targets
0	16
1	114
2	39
3	598
4	430
5	252

Scores of 0 and 1 may represent segmentation of non-malignant morphology, and, therefore, only scores of 2 or higher should be considered [52]. As in this project, the task that the network should accomplish is to recognize what a specialist assumed relevant, all of these targets were considered.

For prostate and target segmentation, two different folders were created with different amounts of data for each imaging method. The first folder contains all the images from the volume and their corresponding prostate and target masks. These are all saved in different folders with the same name. The second folder contains only the images where the prostate is present. This was accomplished by using the prostate STL grid. For this case, the images and corresponding target masks were generated. These are also kept in different folders with the same name. As a result, for the MRI, 51,468 images and corresponding prostate and target masks were generated, with 26,099 images containing the prostate and 6,341 images containing the target.

For training purposes, the folders containing the MRI images were used. The first with 51,468 images and corresponding prostate and target masks. And the second with 26,099 images and corresponding target masks. These are then imported into Python. Figure 3.3 shows an example of the produced images overlaid with the prostate and target masks.



(a) Prostate Mask

(b) Target Mask

Figure 3.3: Example of an MRI image with the prostate mask on the left and the target mask on the right.

3.4 Training

The chosen framework for training is TensorFlow, developed by Google, which provides stable Python as well as C $^{++}$ Application Programming Interfaces (APIs). TensorFlow has an API for automatic differentiation, which is essential for implementing machine learning algorithms like backpropagation for training neural networks. It computes the gradient of computation for specific inputs [53], by using the Data Flow Graph, which comprises several nodes [54]. Each node represents the implementation of a mathematical operation and contains zero or more inputs, and outputs [55].

Additionally, Keras was used. It is a Python-based open-source deep learning library well-known for its clean and simple API. It allows basic deep learning models to be generated, fit, and evaluated in only a few lines of code. Keras also supports using any major deep learning mathematics libraries, such as Tensorflow, as the backend. In addition, the Keras API was included in the Tensorflow version that was released in 2019. This feature simplifies and eases the usage of the TensorFlow project, as using TensorFlow directly may be challenging [56].

The images and corresponding masks must be loaded into Python to train and evaluate the neural network. Nevertheless, the dataset must first be divided into the train, validation, and test sets. The train set is the largest corpus in the dataset and is used for model training; the validation set is used during training to evaluate how well the model is performing, and the test set is used to provide results when training is completed [57].

There must be no patient overlap during splitting. This leads to over-optimistic test-set performance because the model memorizes a characteristic from training and uses it when tested, a phenomenon known as data leakage. As a result, having one patient in the same set, train, validation, or test is critical. Beforehand, the data was split into 60-20-20 for training, validation, and test. This split was done by patient into different folders: train, validation, and test, with each folder containing folders for the images and masks. Because 706 of the 707 patients had a positive target, the patients were divided into sets at random. A layout of the different sets used to train the network is represented in Figure 3.4.

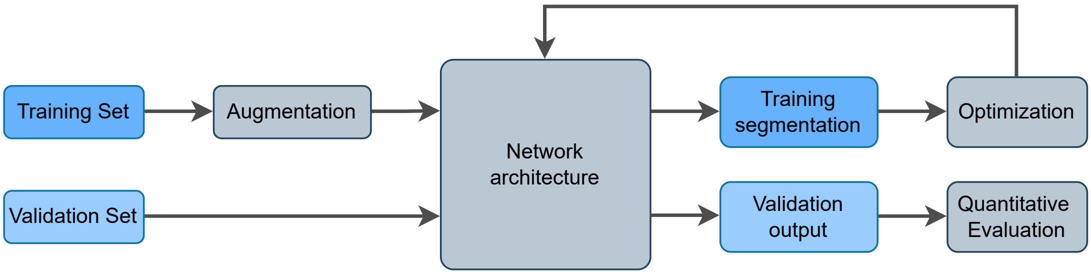


Figure 3.4: Layout for network training.

Next, generators for the images and masks were created and then wrapped together for each set. This was achieved by utilizing the ”ImageDataGenerator” function, which enables the generation of batches of tensor image data with real-time data augmentation.

Because the amount of data available is limited, augmented data can be generated and added to the training set. Therefore, a machine learning model's generalization tends to increase with additional data. This is known as dataset augmentation, and it results in a more robust model [31]. Several types of data augmentation are represented in Figure 3.5.

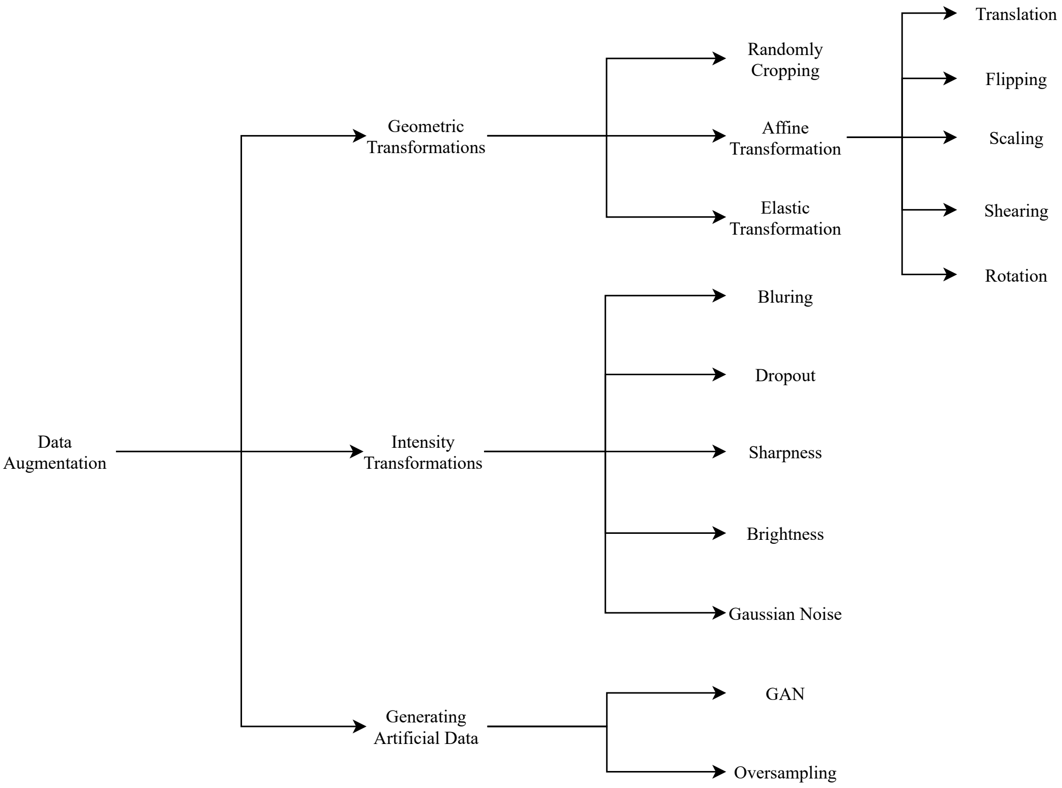


Figure 3.5: Types of data augmentation.

From these types of augmentation, geometric and intensity transformations were applied. The generator function made it possible to select some of the represented augmentation types and select the amount to be applied. To demonstrate the different augmentations applied, the image represented in Figure 3.6 will be considered.

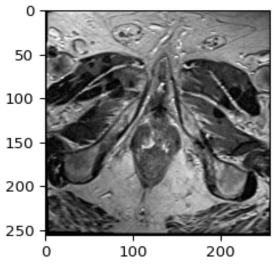
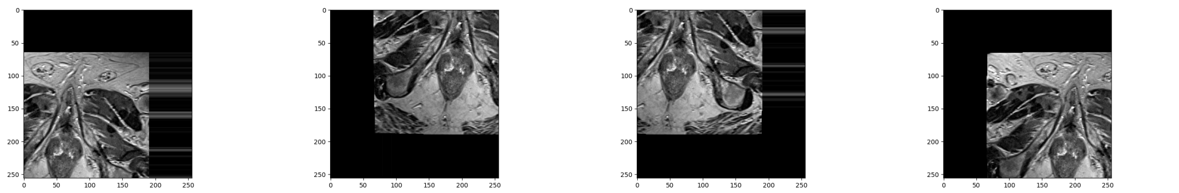


Figure 3.6: Example of an MRI image.

A translation was applied, with a width and height shift in the range of 0.25, corresponding to the proportion of the image size to be shifted. Resulting in a width and height shift in the interval [$-0.25 \times$ image size, $0.25 \times$ image size]. This transformation helps, so the network does not memorize to only look in the center of the images for suspicious areas. Examples of this augmentation are represented in Figure 3.7.



(a) translation= [-0.25, -0.25]

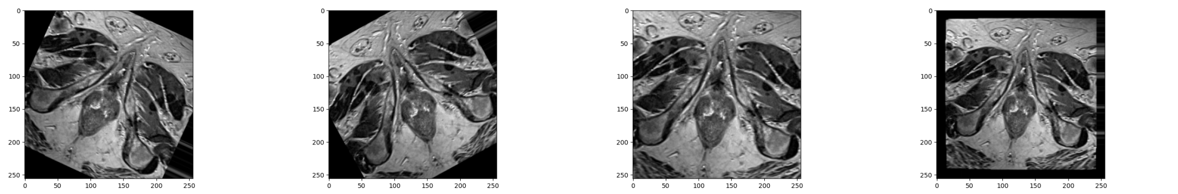
(b) translation= [0.25, -0.25]

(c) translation= [-0.25, 0.25]

(d) translation= [0.25, 0.25]

Figure 3.7: Example of an MRI image with translation, where translation=[height range, width range].

A rotation with a range of 30 degrees was applied. This results in a random rotation of the image in the interval $[-30, 30]$. For scaling, a zoom range of 0.1 was used resulting in an interval of $[0.9, 1.1]$, where values below one zoom in and over one zoom out. Examples of these augmentations are represented in Figure 3.8.



(a) Clockwise rotation

(b) Counterclockwise rotation

(c) Zoom in (zoom factor=0.9)

(d) Zoom out (zoom factor=1.1)

Figure 3.8: Example of an MRI image with rotation (a, b) and with scaling (c, d).

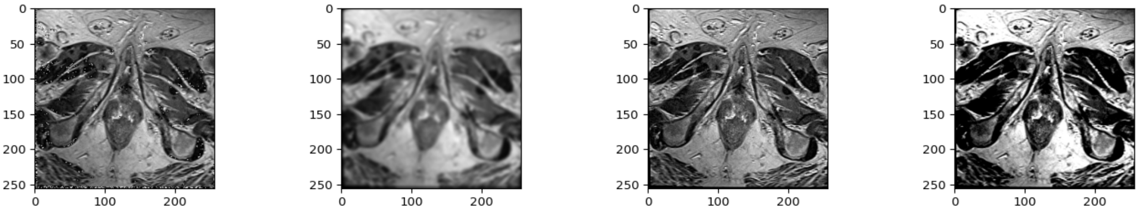
An interval of [0.8, 1.2] was used for brightness, where values under 1 darken the image and over 1 brighten it. Examples of this augmentation are represented in Figure 3.9.

(a) Brightness factor=0.8

(b) Brightness factor=1.2

Figure 3.9: Example of an MRI image with brightness.

Using the pre-processing function from the generator, it was possible to add other types of augmentation. In this case, intensity transformations were applied: Gaussian noise, blurriness, sharpness, and contrast. These transformations were applied randomly, with the possibility of applying none of them, one, several, or even all of them. For Gaussian noise,a zero-mean was used with a variance σ^2 in the range of $[0.1, 2]$. For the blurriness, a filter with size 11×11 was used with σ in the range of $[0.1, 2]$. A factor in the range of $[1, 2]$ was used for sharpness. For contrast, a factor in the range of $[1, 2]$ was used. Examples of these augmentations are represented in Figure 3.10.



(a) Gaussian noise **(b)** Blurriness **(c)** Sharpness **(d)** Contrast

Figure 3.10: Example of an MRI image with the maximum values for Gaussian noise (a), blurriness(b), sharpness (c), and contrast(d).

Now that the data is split into different sets and several types of augmentation have been applied, it is time to create the model to train the network. In this work, four approaches were used.

3.4.1 Simple U-NET

The U-net, discussed in Chapter 2 and represented in Fig 2.9 is used to perform individual prostate and target segmentation. For the target, two scenarios are considered, one for all the images and another one for the set of images that include the prostate. For the input, an input size with dimensions $256 \times 256 \times 1$ was given, with 256×256 the image's dimensions, and 1 representing the number of channels. Also, three downsampling and upsampling levels were used, with two convolutions on each level. Also, several filters are given for each level, for the first level 64, the second 128, and the third 256. The sigmoid activation is used for the output layer, which returns a vector with values between 0 and 1, where each element represents a probability.

3.4.2 Cascade U-NET

To perform prostate and target segmentation simultaneously, a cascade U-net was implemented. This type of architecture is helpful to perform multi-task problems and usually returns better results than doing these tasks individually. It uses the base of the simple U-net, with the same parameters used for this network, the dimensions, upsampling and downsampling levels, amount of convolutions, number of filters and activation function. The difference between these models is that the cascade U-net has an additional decoding path with the same characteristics as the simple U-net decoding path with a sigmoid activation function for the output layer. It receives the input image and corresponding prostate and target masks. The prostate masks are used for the first decoding path to predicting the prostate, and the target masks are used for the second decoding path to predicting the target. Skip connections were added between these two paths, so during training, the first decoding path performs prostate segmentation and passes the information to the second decoding path. The architecture of this model is represented in Figure 3.11.

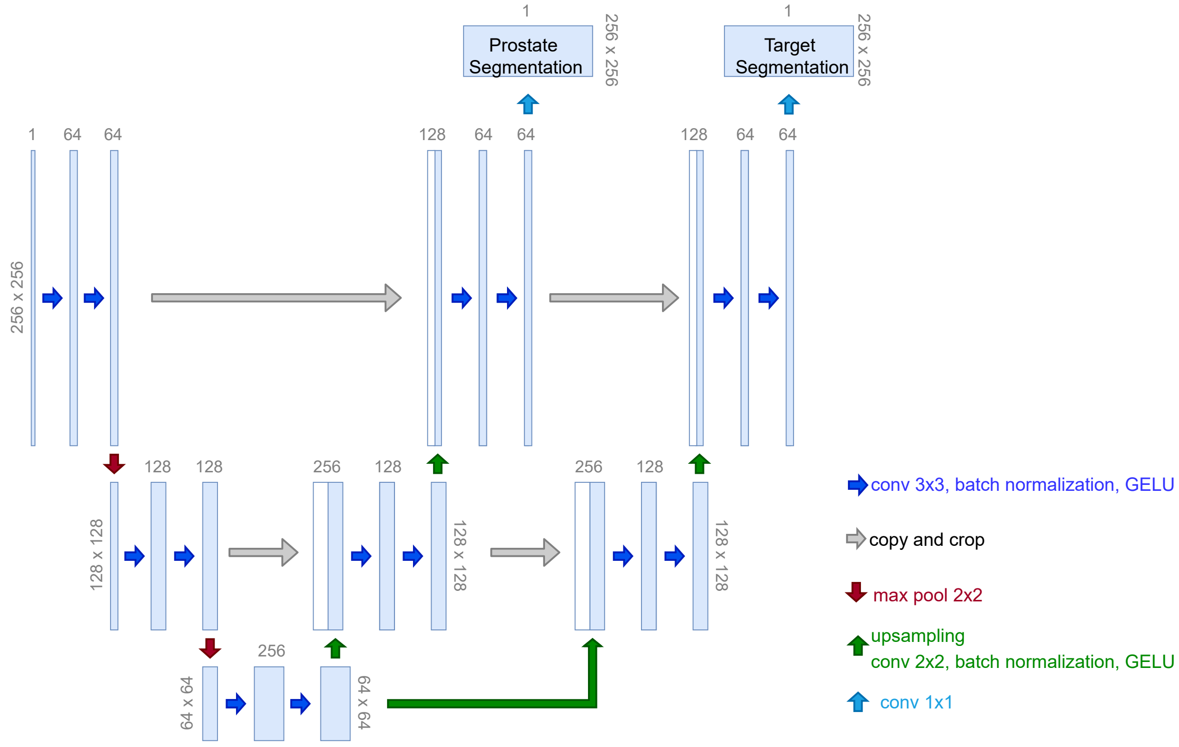


Figure 3.11: Architecture of the cascade U-net model.

However, as this model receives all the images with the corresponding prostate and target masks, the percentage of images containing a positive target is very low. Therefore, it becomes difficult to perform target segmentation using this model as it is.

3.4.3 First adapted cascade U-NET

A third model was implemented to perform target segmentation, consisting of an adaptation between the two previous models. This approach has a similar architecture as the cascade U-net and uses information from the training of the simple U-net for prostate segmentation. As prostate segmentation is a more straightforward task than target segmentation and the simple U-net has shown itself effective in performing this task, the weights from this network are used. These weights are saved for the epoch that achieved the best results during training and loaded into the third model. As the cascade U-net has two decoding paths, the same weights are given to each path. Then, with the weights loaded, as the main interest is to perform target segmentation, the layers in the first decoder are frozen. Freezing these layers prevents the weights from being updated in this path and, therefore, maintain the information acquired while training the simple U-net for prostate segmentation. This is important because the target is always inside the prostate. After this, the images that contain the prostate are used for this network, along with the target masks. The outputs from both paths are summed up and multiplied by 0.5, so it does not return values higher than 1. This adaptation is an approach to get better results for the target segmentation and is represented in Figure 3.12.

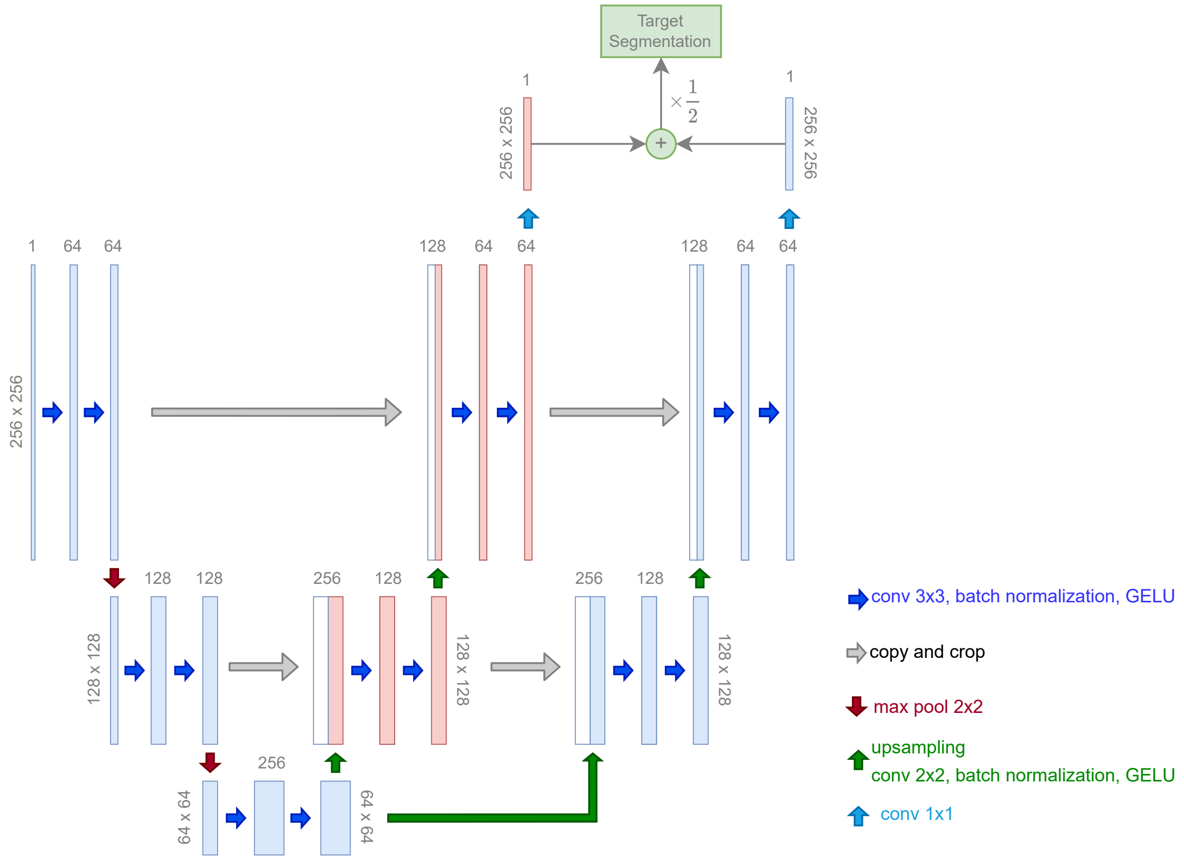


Figure 3.12: Architecture of the first adaptation of the cascade U-net model. The red blocks represent the decoding path were the weights were frozen.

Nevertheless, using just the target masks affects the way the encoder works and, therefore, may lead to the network "forgetting" where the prostate was predicted. Consequently, another adaptation of the cascade U-net is made.

3.4.4 Second adapted cascade U-NET

The second adaptation of the U-net is similar to the previously implemented model and is used to perform target segmentation. In this approach, the weights from the training for the prostate segmentation using the simple U-net are also used, and the first decoder is also frozen. In contrast to the first adaptation of the cascade U-net, this one takes both the prostate and target masks as inputs and returns two outputs. The first output for the prediction of the prostate, which, because of the frozen weights in that path, during training, does not change much, and it helps overcome the "forgetting" of where the prostate was predicted. The second output is obtained the same way as the first adapted cascade U-net output. This adjustment improves the performance of the network and allows a more accurate target segmentation. A representation of this network is given in Figure 3.13.

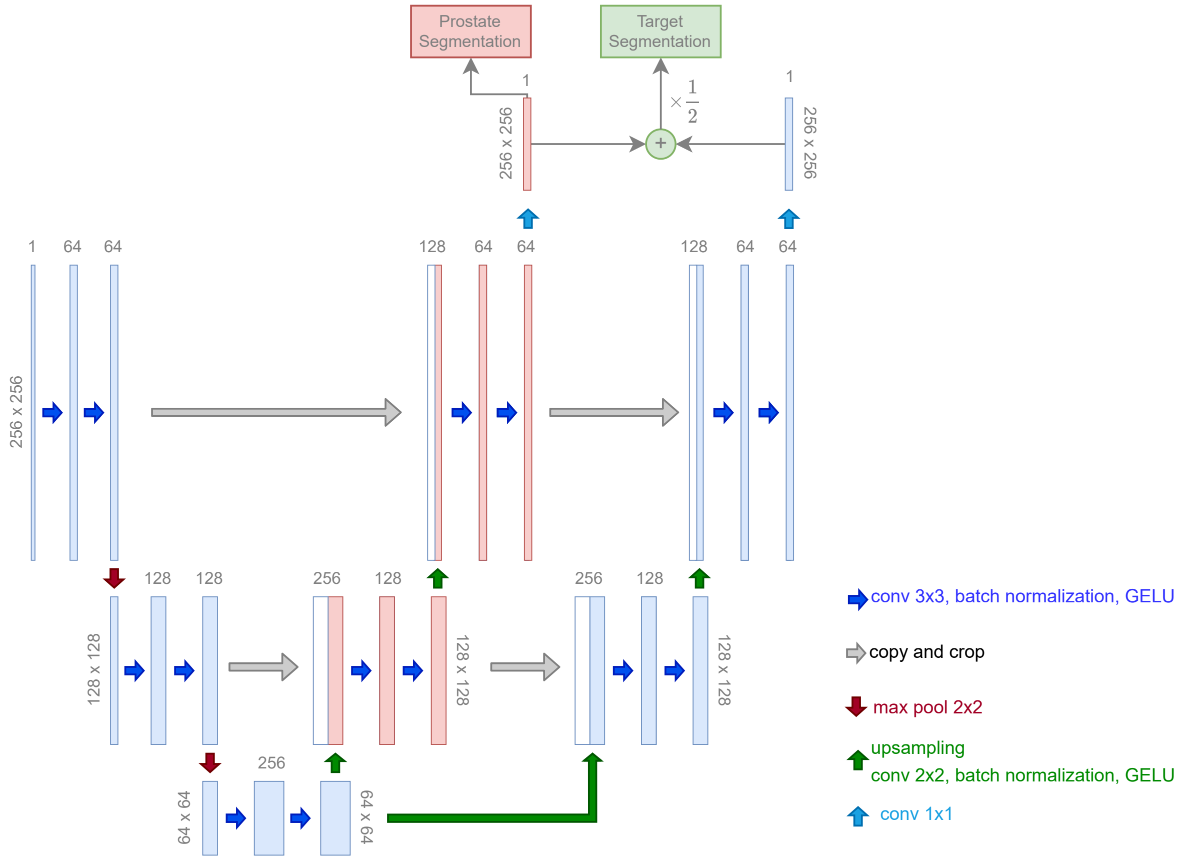


Figure 3.13: Architecture of the second adaptation of the cascade U-net model. The red blocks represent the decoding path were the weights were frozen.

For the optimizer, the **rectified Adam** was used. This optimizer is an adaptation of the Adam optimizer with some differences. The learning rate uses a learning rate warmup strategy, which employs a lower value in the initial few steps instead of a constant learning rate. This optimizer also suppresses the momentum term for the first few input training batches. The Adam optimizer has several issues in terms of generalization: a failure to generalize during the first few batch updates, a high variance, and some convergence concerns. Using the rectified Adam, these issues are overcome and, thus, the name of this optimizer. As it typically results in improved accuracy, a more generalizable network and fewer epochs are necessary for the complete training [58].

As for the loss function, their performance was evaluated to see which one was the most suited for the problem at hand. The loss functions used are represented in Table 3.2.

Table 3.2: Summary of used loss functions [42].

Loss functions	**Use cases**
Binary Cross-Entropy	Works better with well-balanced datasets. A Bernoulli distribution based loss function
Weighted Cross-Entropy	Commonly utilized with skewed dataset The β coefficient is used to weight positive occurrences.
Focal Loss	It works best with highly imbalanced datasets. Reduce the importance of easy examples, allowing the model to learn difficult cases.
Dice Loss	Inspired by the dice coefficient, a metric for evaluating segmentation performance
Tversky Loss	Dice Coefficient Variant False positives and False negatives have added weight
Focal Tversky Loss	Tversky loss variant with an emphasis on difficult cases
Combo Loss	Combination of Dice Loss and Binary Cross-Entropy The benefits of BCE and dice loss are employed in situations where there is a slight class imbalance.
Log Cosh Dice Loss	Dice Loss variant and inspired regression log-cosh method for smoothing For skewed datasets, variations can be employed

From these, the log cosh dice loss, a modification of the dice loss developed by Jadon (2020) [42] performed the best. Therefore, the equations of the dice loss and log cosh dice loss are here represented, whereas the definition of the others can be found in Appendix B.

The **dice loss** is based on the dice similarity coefficient (DSC), defined in equation 2.12. It is defined as the inverse of the DSC, thus 1 – DSC, and is defined as follows [42]:

$$DiceLoss(y, \hat{p}) = 1 - \frac{2y\hat{p} + 1}{y + \hat{p} + 1} \tag{3.1}$$

where y is the expected value, $\hat{p}$ the predicted, and 1 is added in the numerator and denominator to avoid zero division errors when both y and $\hat{p}$ are equal to zero.

Due to the non-convex nature of the dice coefficient, it may not be easy to get optimal outcomes. As a result, log-cosh is a typical strategy for smoothing the curve in regression-based problems. Jadon(2020) [42] shows that it stays continuous and finite after first-order transformation. As a result, this function is distinct from others in that it is tractable while encapsulating the characteristics of the dice coefficient. It is defined as follows [42]:

$$L_{lc-dce} = log(cosh(DiceLoss)) \tag{3.2}$$

For the metrics to be evaluated, from the metrics described in Chapter 2, the values for the accuracy, dice coefficient, sensitivity, specificity, and the output of the loss function were considered.

4

Results

Focusing on the first implemented network, the U-net, several loss functions were tested, with the log cosh dice coefficient performing the best (see appendix B.1). Therefore, this loss function was used to obtain the results for this network and the others.

4.1 Simple U-NET

Considering the simple U-net, three problems were evaluated, prostate segmentation and target segmentation on two different amounts of data. For all of them, a learning rate of 1×10^{-4} was used, and different combinations of augmentation were tested, using geometric transformations (translation, rotation, scaling), with and without intensity transformations (brightness, Gaussian noise, blurriness, sharpness, contrast).

4.1.1 Prostate segmentation

For prostate segmentation the two combinations of augmentations were considered. With the geometric transformations, the network achieved higher values for the DSC in fewer epochs, compared with the combination of geometric and intensity transformations. However, intensity and geometric transformations result in a more robust model, essential for training models. The network ran for 200 epochs with a batch size of 8, and by applying geometric transformations, the dice coefficient evolved as represented in Figure 4.1.

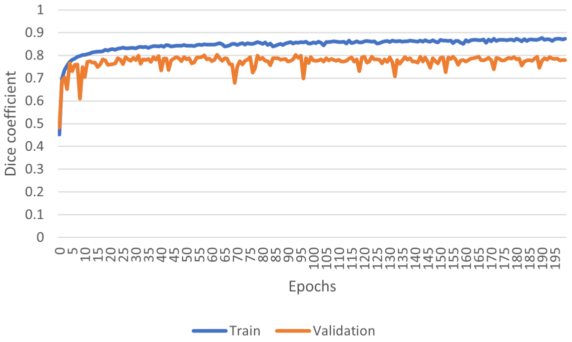


Figure 4.1: Evolution of the dice coefficient for prostate segmentation using the simple U-net with geometric transformations.

For the validation set, the network returned the best results at epoch 63, and the results for the dice coefficient, sensitivity, specificity, and accuracy are represented in Table 4.1.

Table 4.1: Best results of the dice coefficient, sensitivity, specificity, and accuracy for prostate segmentation using a simple U-net with geometric transformations at epoch 63.

Evaluation	DSC	Sensitivity	Specificity	Accuracy
Train	0.8510	0.8551	0.9976	0.9941
Validation	0.8033	0.8039	0.9972	0.9939
Test	0.8188	0.8307	0.9972	0.9947

Considering geometric and intensity transformations, the network also ran for 200 epochs with a batch size of 8, and the dice coefficient evolved as is represented in Figure 4.2.

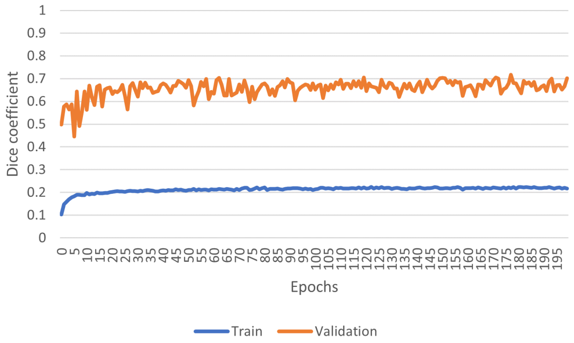


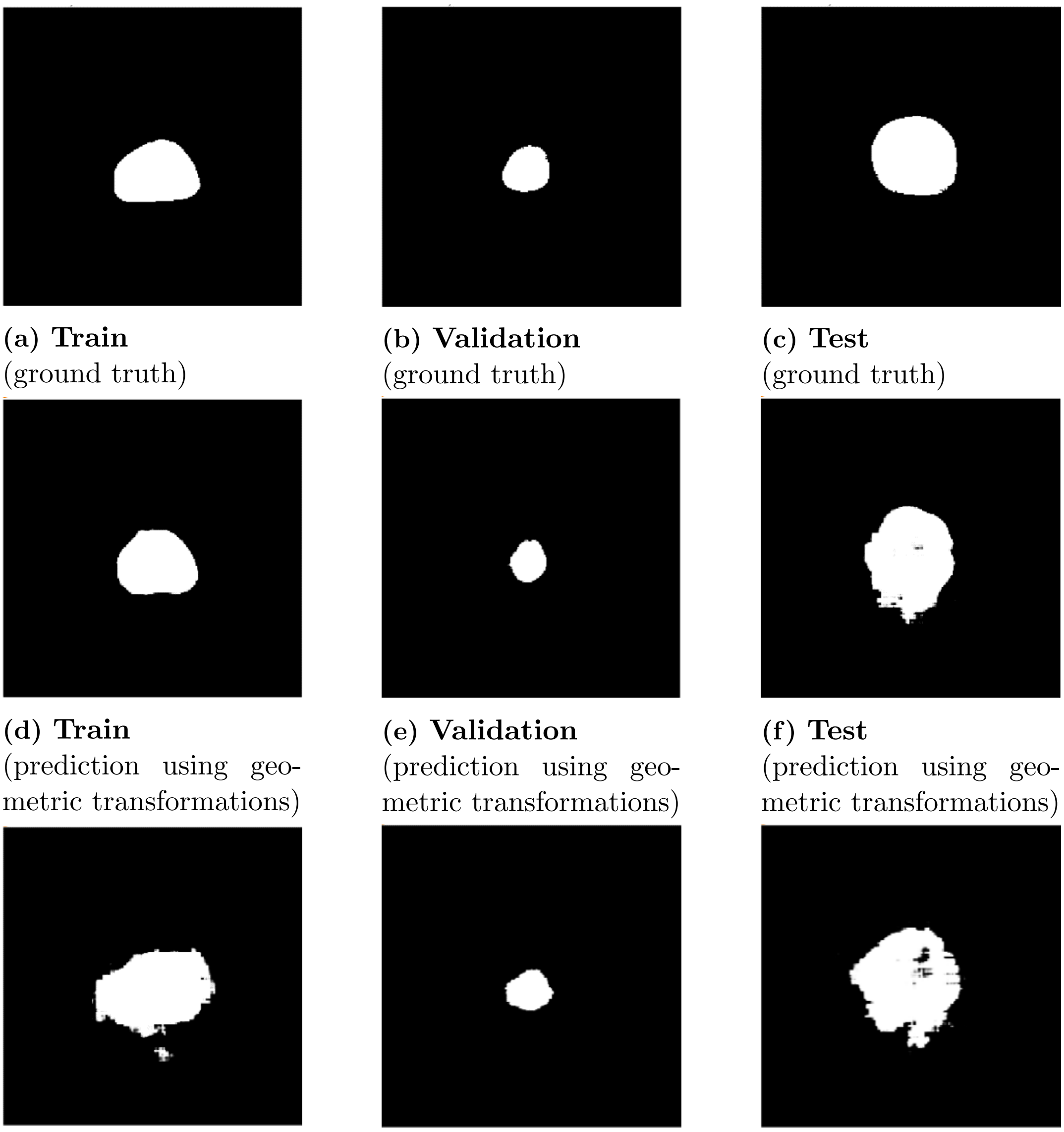
Figure 4.2: Evolution of the dice coefficient for prostate segmentation using the simple U-net with geometric and intensity transformations.

For the validation set, the network returned the best results at epoch 178, and the results for the dice coefficient, sensitivity, specificity, and accuracy are represented in Table 4.2.

Table 4.2: Best results of the dice coefficient, sensitivity, specificity, and accuracy for prostate segmentation using a simple U-net with geometric and intensity transformations at epoch 178.

Evaluation	DSC	Sensitivity	Specificity	Accuracy
Train	0.2188	0.2727	0.9771	0.9631
Validation	0.7172	0.8439	0.9912	0.9889
Test	0.7206	0.8677	0.9907	0.9890

Looking at obtained results, with geometric transformations the network was able to achieve higher values for all the sets, whereas with geometric and intensity transformations the train set had more difficulty. However, with the geometric and intensity transformations the network achieved an higher sensitivity for the validation and test set. To evaluate the networks prediction capability using geometric transformations without and with intensity transformations, during training, images from different sets were evaluated. The results of the network's prediction using these combinations of transformations are shown in Figure 4.3.



(a) Train (ground truth)

(b) Validation (ground truth)

(c) Test (ground truth)

(d) Train (prediction using geometric transformations)

(e) Validation (prediction using geometric transformations)

(f) Test (prediction using geometric transformations)

(g) Train (prediction using geometric and intensity transformations)

(h) Validation (prediction using geometric and intensity transformations)

(i) Test (prediction using geometric and intensity transformations)

Figure 4.3: Example of ground truth prostate masks from each set (a, b, c). Prostate segmentation prediction using a simple U-net and geometric transformations at epoch 63 (d, e, f). Prostate segmentation prediction using a simple U-net and geometric and intensity transformations at epoch 178 (g, h, i).

4.1.2 Target segmentation

Considering all the data, the simple U-net performed poorly for the target segmentation, with a dice coefficient close to zero, and it could not predict anything. This behavior is not unexpected, considering the number of images with a positive target in the entire data and the size of the target itself. Therefore, the images were filtered to include only those that contained the prostate to perform this task. The network ran for 200 epochs with a batch size of 8, and by applying geometric transformations, the dice coefficient evolved as represented in Figure 4.4.

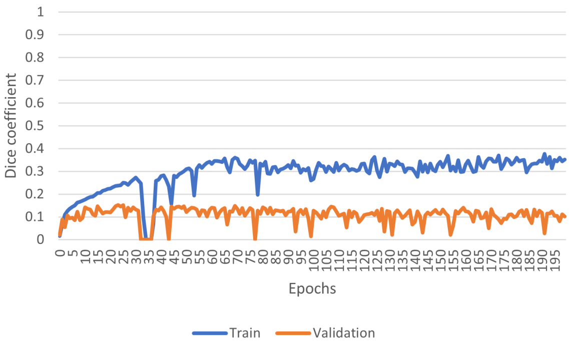


Figure 4.4: Evolution of the dice coefficient for target segmentation using the simple U-net with geometric transformations.

For the validation set, the network returned the best value for the dice coefficient at epoch 24. Therefore, the results for the dice coefficient, sensitivity, specificity, and accuracy for this epoch are represented in Table 4.3.

Table 4.3: Best results of the dice coefficient, sensitivity, specificity, and accuracy for target segmentation using a simple U-net with geometric transformations at epoch 24.

Evaluation	**DSC**	**Sensitivity**	**Specificity**	**Accuracy**
Train	0.2379	0.2879	0.9990	0.9934
Validation	0.1512	0.1630	0.9994	0.9935
Test	0.1338	0.1442	0.995	0.9990

The network had more difficulty performing the target segmentation by applying geometric and intensity transformations. The network ran for 200 epochs, and the dice coefficient evolved very slowly, achieving 0.0145 for training and 0.0161 for validation. Whereas for the sensitivity, the network returned the value 0. For both combinations of augmentation, the network could not predict the target in the correct place.

4.2 Cascade U-NET

The cascade U-net was used to perform simultaneous prostate and target segmentation. The network ran for 200 epochs with a batch size of 4 and a learning rate of 1×10^{-4}. Only the geometric transformations were considered because the simple U-net performed better using only these. The evolution of the dice coefficient of the prostate and target segmentation is represented in Figure 4.5 and Figure 4.6, respectively.

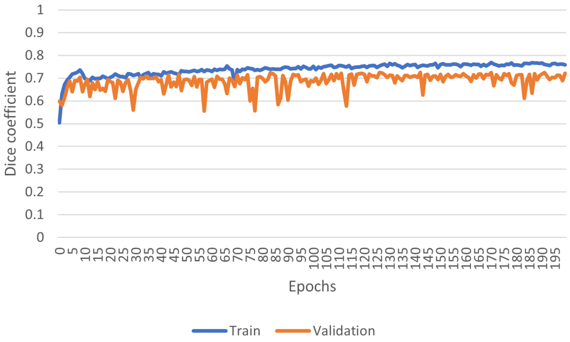


Figure 4.5: Evolution of the dice coefficient for prostate segmentation using the cascade U-net with geometric transformations.

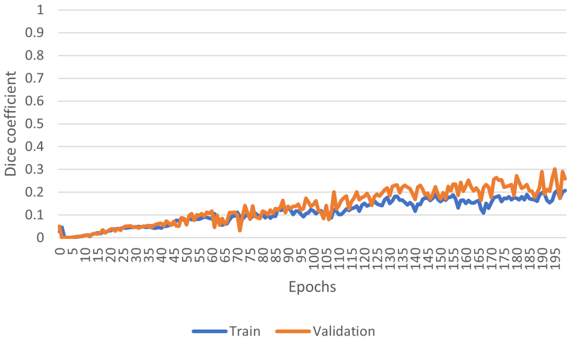
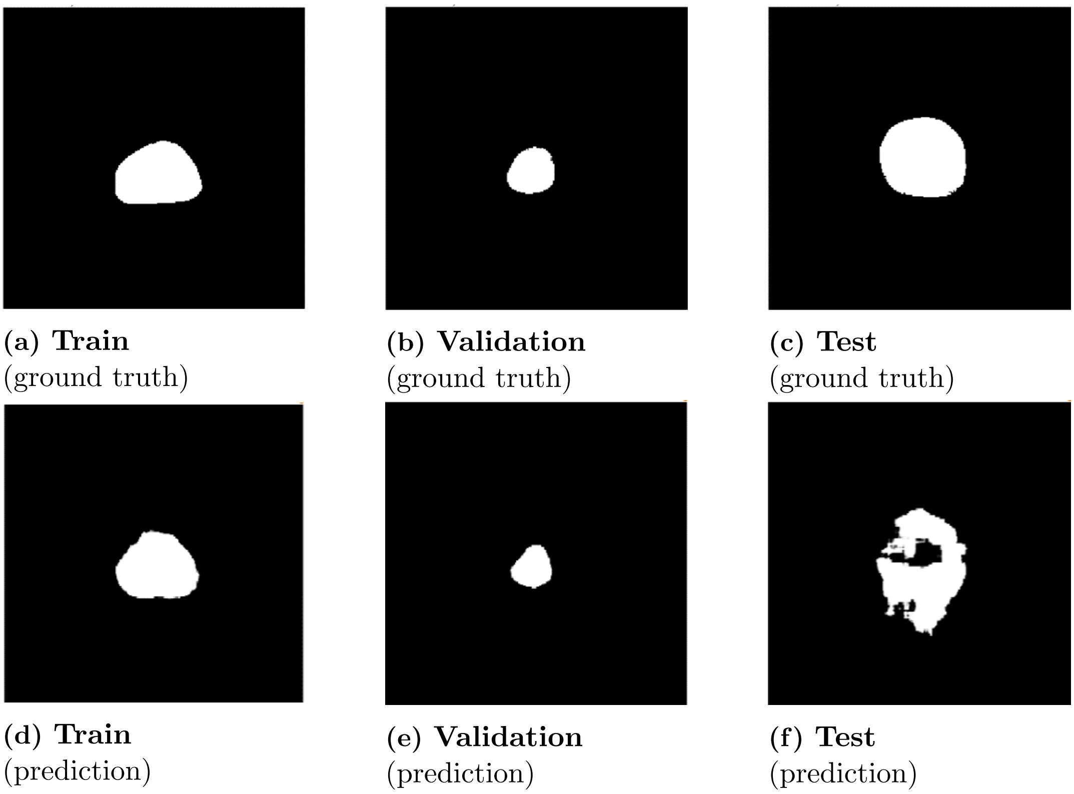


Figure 4.6: Evolution of the dice coefficient for target segmentation using the cascade U-net with geometric transformations.

Table 4.4: Results of the dice coefficient, sensitivity, specificity, and accuracy for prostate and target segmentation using a cascade U-net with geometric transformations at the last epoch.

Segmentation type	**Evaluation**	**DSC**	**Sensitivity**	**Specificity**	**Accuracy**
Prostate Segmentation	Train	0.7595	0.7758	0.9972	0.9934
	Validation	0.7219	0.7425	0.9967	0.9933
	Test	0.7389	0.7316	0.7716	0.9941
Target Segmentation	Train	0.2074	0	1	0.9994
	Validation	0.2595	0	1	0.9995
	Test	0.2850	0	1	0.9997

The cascade U-net achieved lower values for the DSC and the sensitivity of the prostate compared with the simple U-net. Therefore, returning a poorer prediction, shown in Figure 4.7. For the target, even though the dice coefficient increased over training, the sensitivity was zero, and the network was not able to segment it.



(a) **Train** (ground truth)

(b) **Validation** (ground truth)

(c) **Test** (ground truth)

(d) **Train** (prediction)

(e) **Validation** (prediction)

(f) **Test** (prediction)

Figure 4.7: Example of ground truth prostate masks from each set (a, b, c). Prostate segmentation prediction using a cascade U-net and geometric transformations at epoch 200 (d, e, f).

4.3 First adapted cascade U-NET

Taking advantage of the fact that the prostate segmentation using the simple U-net returned promising results, now the interest relies on getting better results for the target segmentation. Therefore, the third model was implemented. The model used only geometric transformations, and the weights from the simple U-net for prostate segmentation using geometric with and without intensity transformations were considered. The network ran for 500 epochs with a batch size of 4 using a lower learning rate of 1×10^{-6}. Using the weights from the prostate segmentation using geometric transformations returned an evolution of the dice coefficient represented in Figure 4.8.

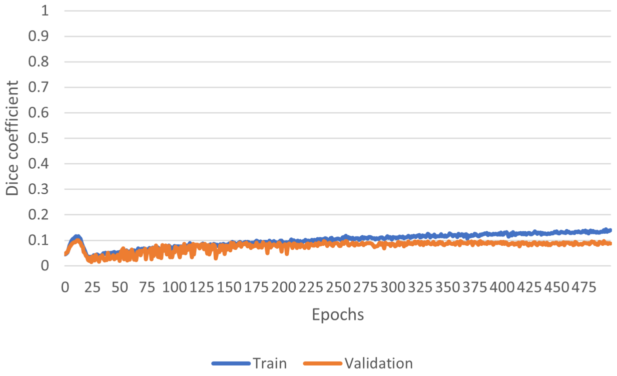


Figure 4.8: Evolution of the dice coefficient for target segmentation using the first adapted cascade U-net with geometric transformations.

The highest value for the validation was achieved at the beginning of training and, therefore, will not be considered. Thus, the results from the last epoch will be considered and are represented in Table 4.5. For the test set, the model was evaluated at the last epoch.

Table 4.5: Results of the dice coefficient, sensitivity, specificity, and accuracy for target segmentation using the first adapted cascade U-net with geometric transformations at the last epoch.

Evaluation	DSC	Sensitivity	Specificity	Accuracy
Train	0.1401	0.1335	0.9998	0.9990
Validation	0.0870	0.0816	0.9990	0.9989
Test	0.0782	0.0769	0.9996	0.9991

Considering the weights from the prostate segmentation using geometric and intensity transformations returned an evolution of the dice coefficient represented in Figure 4.9.

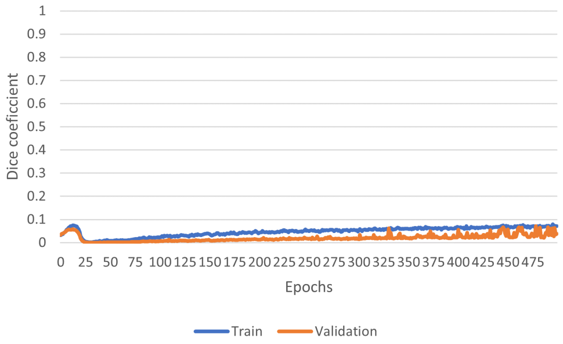


Figure 4.9: Evolution of the dice coefficient for target segmentation using the first adapted cascade U-net with geometric and intensity transformations.

Here, the peak for the validation was achieved at epoch 479. Therefore the results from this epoch were considered, which are represented in Table 4.6.

Table 4.6: Best results of the dice coefficient, sensitivity, specificity, and accuracy for target segmentation using the first adapted cascade U-net with geometric and intensity transformations at epoch 479.

Evaluation	**DSC**	**Sensitivity**	**Specificity**	**Accuracy**
Train	0.0701	0.0637	0.9998	0.9989
Validation	0.0717	0.0577	0.9998	0.9989
Test	0.0618	0.0432	0.9998	0.9992

It is possible to see that the network performed better using the weights of the prostate segmentation using only geometric transformations. With these transformations the network predicted the target in the wrong place, whereas, using geometric and intensity transformations it predicted mostly nothing. The results are still far from optimal, so the second adapted cascade U-net was implemented.

4.4 Second adapted cascade U-NET

It was already possible to see that the network performs more poorly when using the weights from the prostate segmentation with geometric and intensity transformations. Only the weights from the prostate segmentation with geometric transformations were considered. The network ran for 200 epochs with a learning rate of 1×10^{-6}. The evolution of the dice coefficient for the target segmentation is represented in Figure 4.10.

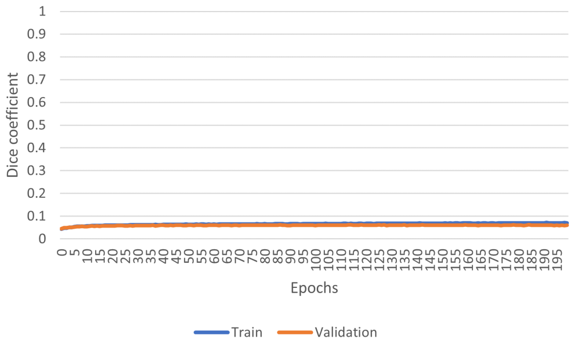


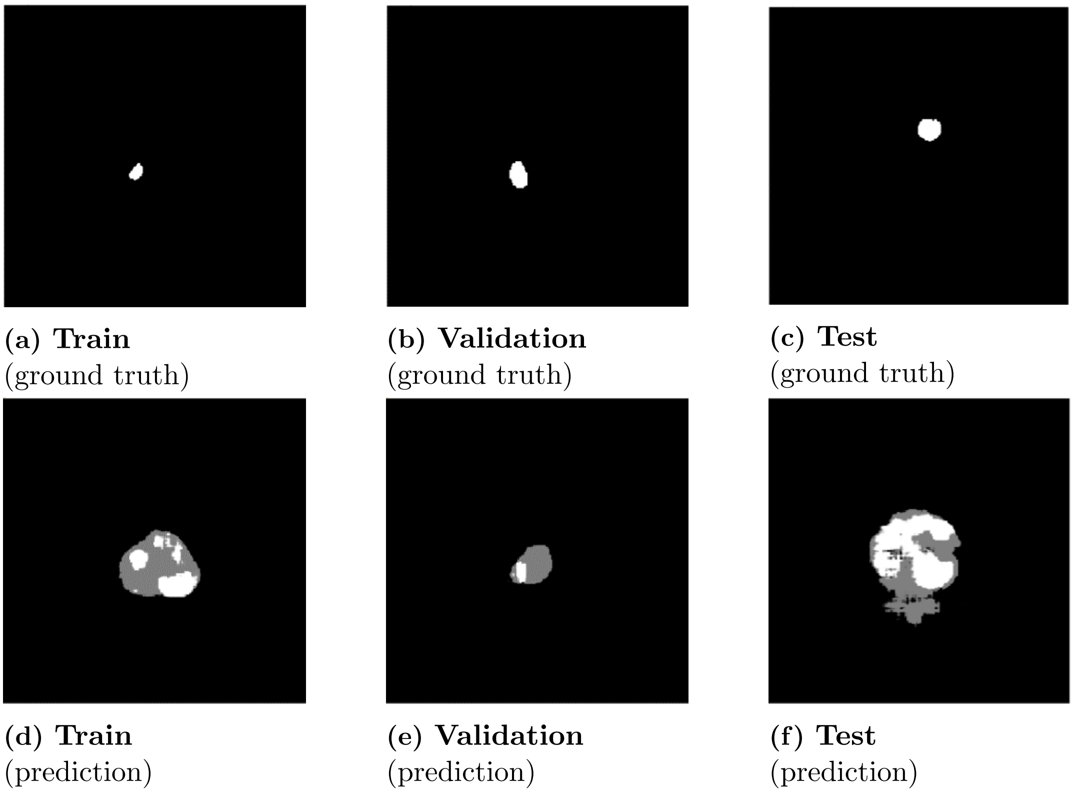
Figure 4.10: Evolution of the dice coefficient for target segmentation using the second adapted cascade U-net with geometric transformations.

This architecture achieved the best results at epoch 185, and the results for the dice coefficient, sensitivity, specificity, and accuracy are represented in Table 4.7.

The results of the network's prediction using geometric transformations for the second adapted cascade U-net are shown in Figure 4.11. Here, the lighter gray comes from the loaded weights from the simple U-net and in white is the prediction of the target.

Table 4.7: Best results of the dice coefficient, sensitivity, specificity, and accuracy for target segmentation using the second adapted cascade U-net with geometric transformations at epoch 185.

Evaluation	**DSC**	**Sensitivity**	**Specificity**	**Accuracy**
Train	0.0707	0.5189	0.9714	0.9925
Validation	0.0623	0.4150	0.9743	0.9919
Test	0.0499	0.3689	0.9731	0.9914



(a) Train (ground truth)

(b) Validation (ground truth)

(c) Test (ground truth)

(d) Train (prediction)

(e) Validation (prediction)

(f) Test (prediction)

Figure 4.11: Example of ground truth target masks from each set (a, b, c). Target segmentation prediction using the second adapted cascade U-net and geometric transformations at epoch 185 (d, e, f).

Higher learning rates were tested for this network, which retrieved better results for the evaluated metrics for training but performed worse for the validation and test set.

4.5 Final discussion

Considering the first network used, U-net, several conclusions were drawn using this model. While trying different loss functions, it was possible to see the difference between them and the importance of choosing the loss function for a specific task. While looking at the existing optimizers, Adam seemed to be the more promising but had some fluctuations that sometimes led to catastrophic interference. This event led to the network forgetting everything it had already learned. The rectified Adam is an excellent alternative to this optimizer as it is more stable, has fewer fluctuations, and adapts the learning rate during training. This adaptation is also helpful in choosing a learning rate as the same will not have as much impact as using the Adam optimizer. This optimizer has therefore shown to perform well with the task at hand.

Considering the different segmentations, using the simple U-net, it was possible to achieve good results overall for prostate segmentation. Whereas, for the target, there were several complications.

Using augmentation turned the model more robust, but it was clear that the intensity transformations were too extreme for both types of segmentation. For the prostate segmentation, as it is an easier task, the network was able to return good values for the validation set compared to the training one. Whereas, too much augmentation was applied for target segmentation as the network returned shallow values with zero sensitivity.

Considering all the MRI images from the reviewed patients, the percentage of images with a target is limited, and therefore, the task at hand ends up being complicated to perform. It was possible to see the difficulty of this task by looking at the results returned from the network, with a low dice coefficient and sensitivity of zero and specificity of one, indicating that the network mostly predicted everything as not having a target. As there are only targets inside the prostate, the images containing the prostate were considered, lowering the total number of images and increasing the percentage of images with a positive target. This improved the network's performance but not enough to predict in the right place.

This led to implementing the cascade U-net with the intention that the path from the decoding path of the target segmentation would learn enough from the decoding path of the prostate segmentation. For this network, all the images were used. The network could not predict the target, achieving a sensitivity of zero and specificity of one; therefore, it mostly predicted everything as not having a target.

Thus, the implementation of the adapted cascade U-net was done. Considering the first adaptation of the cascade U-net, it achieved worse results than the simple U-net. This can mainly be justified with the encoder "forgetting" what it had learned for the location of the prostate, and therefore, not being able to predict the target at all or predicting it in the wrong place. An approach to solve this problem was the second adapted cascade U-net. This network maintained the information acquired from the weights of the prostate segmentation and achieved the best results for target segmentation from all the implemented networks.

5

Conclusions and Future Work

Establishing early, rapid, and rigorous diagnostic methods and techniques is crucial with the significant rise in prostate cancer incidence and mortality risk. Some of these procedures rely on medical imaging tools to identify prostate cancer. A clinician then analyzes and reviews the images that have been obtained. However, detecting prostate cancer is primarily reliant on expert knowledge and is subject to substantial fluctuation, which may result in an inaccurate diagnosis. Computational methods such as machine learning models have been proven as effective tools to address these issues. However, because cancer diagnosis is complex, most traditional methods become ineffective. Therefore, deep learning (DL) techniques may be used to automatically extract and learn from a large number of data features to overcome this issue.

Since prostate cancer is more challenging to diagnose than other tumors due to the organ's anatomy, in this thesis, different machine learning models were implemented to help with the diagnosis. These models were based on the U-net model, a model that has shown itself effective for image segmentation problems. It was seen that the simple U-net achieved good segmentation accuracy for the prostate itself but not for the target. To identify prostate cancer, it was necessary to change the simple U-net to overcome this problem. Therefore, three models were implemented, a cascade U-net and two adaptations from this network. The cascade U-net was implemented to perform multitask segmentation to improve the overall segmentation accuracy for the prostate and target. However, this network was not sufficient to achieve the pretended results. This led to the other two models that used the weights from the simple U-net on a cascade U-net type architecture. From the set of models tested, the last implemented network returned the best results for target segmentation and mostly predicted the target in the correct area of the prostate by using its location information, even though it did not achieve a high value for the dice similarity coefficient (DSC). Even though the results of this network seem promising for identifying prostate cancer, future work must be implemented to increase the accuracy of the algorithm and improve the overall performance.

Work has been started for the inclusion of ultrasound data to improve prostate imaging (see Appendix C). The integration of this imaging method in combination with the MRI allows the extraction of useful underlying information. However, the implementation and network structures to do this work are beyond this thesis's scope. As ultrasound is cheaper than MRI but more challenging to understand because of insufficient anatomical information, registration between them is something to pursue and apply in future work since it can be a game-changer for ultrasound images. To sum up, it may be possible to use ultrasound images to identify prostate cancer by enhancing the last network and applying the registration methods.

Bibliography

[1] H. Sung, J. Ferlay, R. L. Siegel, M. Laversanne, I. Soerjomataram, A. Jemal, and F. Bray, "Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA: A Cancer Journal for Clinicians*, vol. n/a, no. n/a.

[2] P. Rawla, "Epidemiology of Prostate Cancer," *World Journal of Oncology*, vol. 10, pp. 63–89, Apr. 2019.

[3] "Prostate Cancer Imaging: Practice Essentials, Radiography, Computed Tomography," Apr. 2020.

[4] B. Turkbey, A. M. Brown, S. Sankineni, B. J. Wood, P. A. Pinto, and P. L. Choyke, "Multiparametric Prostate Magnetic Resonance Imaging in the Evaluation of Prostate Cancer," *CA: a cancer journal for clinicians*, vol. 66, pp. 326–336, July 2016.

[5] R. Kalantar, G. Lin, J. M. Winfield, C. Messiou, S. Lalondrelle, M. D. Blackledge, and D.-M. Koh, "Automatic Segmentation of Pelvic Cancers Using Deep Learning: State-of-the-Art Approaches and Challenges," *Diagnostics*, vol. 11, p. 1964, Nov. 2021.

[6] J. Yang, G. Sharp, and M. Gooding, *Auto-Segmentation for Radiation Oncology: State of the Art (Series in Medical Physics and Biomedical Engineering)*. CRC Press, 1 ed., 2021.

[7] S. Kazemifar, A. Balagopal, D. Nguyen, S. McGuire, R. Hannan, S. Jiang, and A. Owrangi, "Segmentation of the prostate and organs at risk in male pelvic CT images using deep learning," *Biomedical Physics & Engineering Express*, vol. 4, p. 055003, July 2018.

[8] L. Geng, J. Wang, Z. Xiao, J. Tong, F. Zhang, and J. Wu, "Encoder-decoder with dense dilated spatial pyramid pooling for prostate MR images segmentation," *Computer Assisted Surgery*, vol. 24, pp. 13–19, Oct. 2019.

[9] M. N. N. To, D. Q. Vu, B. Turkbey, P. L. Choyke, and J. T. Kwak, "Deep dense multi-path neural network for prostate segmentation in magnetic resonance imaging," *International Journal of Computer Assisted Radiology and Surgery*, vol. 13, pp. 1687–1696, Nov. 2018.

[10] F. Zabihollahy, N. Schieda, S. Krishna Jeyaraj, and E. Ukwatta, "Automated segmentation of prostate zonal anatomy on T2-weighted (T2W) and apparent diffusion coefficient (ADC) map MR images using U-Nets," *Medical Physics*, vol. 46, no. 7, pp. 3078–3090, 2019.

[11] Y. Zhu, R. Wei, G. Gao, L. Ding, X. Zhang, X. Wang, and J. Zhang, "Fully automatic segmentation on prostate mr images based on cascaded fully convolution network," *Journal of Magnetic Resonance Imaging*, vol. 49, no. 4, pp. 1149–1156, 2019.

[12] Z. Dai, E. Carver, C. Liu, J. Lee, A. Feldman, W. Zong, M. Pantelic, M. Elshaikh, and N. Wen, "Segmentation of the Prostatic Gland and the Intraprostatic Lesions on Multiparametic Magnetic Resonance Imaging Using Mask Region-Based Convolutional Neural Networks," *Advances in Radiation Oncology*, vol. 5, pp. 473–481, Feb. 2020.

[13] Z. Liu, W. Jiang, K. H. Lee, Y. L. Lo, Y. L. Ng, Q. Dou, V. Vardhanabhuti, and K. W. Kwok, *A Two-Stage Approach for Automated Prostate Lesion Detection and Classification with Mask R-CNN and Weakly Supervised Deep Neural Network*. Springer., 2019.

[14] R. Alkadi, F. Taher, A. El-baz, and N. Werghi, "A Deep Learning-Based Approach for the Detection and Localization of Prostate Cancer in T2 Magnetic Resonance Images," *Journal of Digital Imaging*, vol. 32, pp. 793–807, Oct. 2019.

[15] Y. Zhu, L. Wang, M. Liu, C. Qian, A. Yousuf, A. Oto, and D. Shen, "MRI-based prostate cancer detection with high-level representation and hierarchical classification," *Medical Physics*, vol. 44, no. 3, pp. 1028–1039, 2017.

[16] S. Kohl, D. Bonekamp, H.-P. Schlemmer, K. Yaqubi, M. Hohenfellner, B. Hadaschik, J.-P. Radtke, and K. Maier-Hein, "Adversarial Networks for the Detection of Aggressive Prostate Cancer," *arXiv:1702.08014 [cs]*, Feb. 2017. arXiv: 1702.08014.

[17] I. Gubins and R. C. Veltkamp, "Deeply Cascaded U-Net for Multi-Task Image Processing," *arXiv:2005.00225 [cs]*, May 2020. arXiv: 2005.00225.

[18] "How does the prostate work?," Aug. 2016.

[19] "Prostate: Functions, diseases, and tests," Aug. 2020.

[20] S. S. P. DSc, *Gray's Anatomy: The Anatomical Basis of Clinical Practice*. Elsevier, 41 ed., 2015.

[21] D. Ilic, M. M. Neuberger, M. Djulbegovic, and P. Dahm, "Screening for prostate cancer," *Cochrane Database of Systematic Reviews*, no. 1, 2013.

[22] "Tests for Prostate Cancer | Prostate Cancer Diagnosis."

[23] R. García-Figueiras, S. Baleato-González, A. R. Padhani, A. Luna-Alcalá, J. A. Vallejo-Casas, E. Sala, J. C. Vilanova, D.-M. Koh, M. Herranz-Carnero, and H. A. Vargas, "How clinical imaging can assess cancer biology," *Insights into Imaging*, vol. 10, p. 28, Mar. 2019.

[24] M. A. Morgan, "Ultrasound (introduction) | Radiology Reference Article | Radiopaedia.org."

[25] G. Katti, S. A. Ara, and A. Shireen, "Magnetic Resonance Imaging (MRI) – A Review."

[26] V. S. Khoo, D. P. Dearnaley, D. J. Finnigan, A. Padhani, S. F. Tanner, and M. O. Leach, "Magnetic resonance imaging (MRI): considerations and applications in radiotherapy treatment planning," *Radiotherapy and Oncology*, vol. 42, pp. 1–15, Jan. 1997.

[27] H. C. Demirel and J. W. Davis, "Multiparametric magnetic resonance imaging: Overview of the technique, clinical applications in prostate biopsy and future directions," *Turkish Journal of Urology*, vol. 44, pp. 93–102, Mar. 2018.

[28] A. Stabile, F. Giganti, A. B. Rosenkrantz, S. S. Taneja, G. Villeirs, I. S. Gill, C. Allen, M. Emberton, C. M. Moore, and V. Kasivisvanathan, "Multiparametric MRI for prostate cancer diagnosis: current status and future directions," *Nature Reviews Urology*, vol. 17, pp. 41–61, Jan. 2020.

[29] T. Barrett, "What is multiparametric-MRI of the prostate and why do we need it?," *Imaging in Medicine*, vol. 7, no. 2, 2015.

[30] M. Czarniecki, "Prostate Imaging-Reporting and Data System (PI-RADS) | Radiology Reference Article | Radiopaedia.org."

[31] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016. `http://www.deeplearningbook.org`.

[32] T. Mitchell, *Machine Learning.* McGraw-Hill Education, 1997.

[33] F. Chollet, *Deep learning with Python.* Simon and Schuster, 2021.

[34] X.-D. Zhang, "Machine Learning," pp. 223–440, Singapore: Springer, 2020.

[35] P. Sydenham and R. Thorn, *Handbook of Measuring System Design, 3 Volume Set.* Wiley, 1 ed., 2005.

[36] A. Krogh, "What are artificial neural networks?," *Nature Biotechnology*, vol. 26, pp. 195–197, Feb. 2008.

[37] M. T. Hagan, H. B. Demuth, and M. Beale, *Neural network design.* PWS Publishing Co., 1997.

[38] S. Sharma and S. Sharma, "Activation functions in neural networks," *Towards Data Science*, vol. 6, no. 12, pp. 310–316, 2017.

[39] A. Al-Masri, "How Does Back-Propagation in Artificial Neural Networks Work?," Jan. 2019.

[40] C. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer, 2006.

[41] M. A. Nielsen, "Neural Networks and Deep Learning," 2015.

[42] S. Jadon, "A survey of loss functions for semantic segmentation," *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pp. 1–7, Oct. 2020. arXiv: 2006.14822.

[43] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, Jan. 2017. arXiv: 1412.6980.

[44] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into Imaging*, vol. 9, pp. 611–629, Aug. 2018.

[45] O. Ronneberger, P.Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," vol. 9351, pp. 234–241, 2015. (available on arXiv:1505.04597 [cs.CV]).

[46] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *CoRR*, vol. abs/1411.4038, 2014.

[47] C. Labs, "Understanding U-Net Architecture For Image Segmentation," Dec. 2018.

[48] A. de Gelder and H. Huisman, "Autoencoders for Multi-Label Prostate MR Segmentation," *arXiv:1806.08216 [cs, eess]*, June 2018. arXiv: 1806.08216.

[49] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *arXiv:1502.03167 [cs]*, Mar. 2015. arXiv: 1502.03167.

[50] D. Hendrycks and K. Gimpel, "Gaussian Error Linear Units (GELUs)," *arXiv:1606.08415 [cs]*, July 2020. arXiv: 1606.08415.

[51] A. Mishra, "Metrics to Evaluate your Machine Learning Algorithm," May 2020.

[52] S. Natarajan, A. Priester, D. Margolis, J. Huang, and L. Marks, "Prostate MRI and Ultrasound With Pathology and Coordinates of Tracked Biopsy (Prostate-MRI-US-Biopsy)," 2020. type: dataset.

[53] "Introduction to gradients and automatic differentiation | TensorFlow Core."

[54] G. Zaccone, *Getting started with TensorFlow.* Packt Publishing Birmingham, 2016.

[55] S. S. Girija, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *Software available from tensorflow. org*, vol. 39, no. 9, 2016.

[56] J. Brownlee, "TensorFlow 2 Tutorial: Get Started in Deep Learning With tf.keras," Dec. 2019.

[57] J. S. SEP 4 and . . M. Read, "The Train, Validation, Test Split and Why You Need It," Sept. 2020.

[58] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the Variance of the Adaptive Learning Rate and Beyond," *arXiv:1908.03265 [cs, stat]*, Oct. 2021. arXiv: 1908.03265.

[59] A. Berger, "Magnetic resonance imaging," *BMJ : British Medical Journal*, vol. 324, p. 35, Jan. 2002.

[60] S. Currie, N. Hoggard, I. Craven, M. Hadjivassiliou, and I. Wilkinson, "Understanding MRI: Basic MR physics for physicians," *Postgraduate medical journal*, vol. 89, Dec. 2012.

[61] J. Jones, "Larmor frequency | Radiology Reference Article | Radiopaedia.org."

[62] J. Chappelow, B. N. Bloch, N. Rofsky, E. Genega, R. Lenkinski, W. DeWolf, and A. Madabhushi, "Elastic registration of multimodal prostate MRI and histology via multiattribute combined mutual information," *Medical Physics*, vol. 38, no. 4, pp. 2005–2018, 2011.

A

Magnetic Resonance Imaging

The foundations of this method rely on the properties of the atoms. An atom is composed of protons and neutrons, which possess a spin property. A spin is associated with an electrical charge, generating a magnetic field, the magnetic moment, in nuclei with impaired nucleons, that is, a different number of protons and neutrons, causing them to behave like a small magnetic bar [25]. For imaging, hydrogen atoms are used, as they comprise single and unpaired protons that act as a magnetic pole and are the most abundant in the body in the form of water and fat [25, 59]. Conventionally, as the hydrogen protons spin in the body, their axes are randomly oriented, canceling each other out. However, when the patient is placed in a strong external magnetic field B_0, the proton axes align to form a magnetic vector oriented along the axis of the MRI scanner, either with (parallel) or against (anti-parallel) the external field, with more protons aligned parallel as it requires less energy, being, therefore, the preferred state [60].

When protons are exposed to an external static magnetic field, their axes oscillate or wobble around the B_0 axis with a slight tilt, called precession, which can be compared to the movement of a spinning top. The precession frequency, also called Larmor frequency, measures the velocity of precession, the number of times the protons precess per second. It is proportional to the strength of the applied magnetic field and can be determined by the Larmor equation [60]:

$$\omega = \gamma B \tag{A.1}$$

where ω is the Larmor frequency in MHz, γ the gyromagnetic ratio in MHz/T [61] - a constant specific to each atomic species [60] - and B the intensity of B_0 in T [61].

This precession produces longitudinal and transverse components in the magnetic moments of the protons, where some cancel each other out, creating a non-null net magnetization vector parallel with the applied magnetic vector referred to as longitudinal magnetization. As the magnetic force of the patient is in the same direction as the external field, it cannot be measured, and it is perceived as a stationary vector. Therefore, a magnetization that lies at an angle to B_0 becomes required, which can be accomplished by switching RF pulses on and off. The purpose of these pulses is to transfer energy to the protons and make them fall out of alignment with B_0. This happens when the pulses have the same frequency as the precessional frequency of the protons, a phenomenon called resonance, and hence the term magnetic resonance imaging [60].

By triggering RF pulses, some protons are excited, which moves them to a higher energy state, anti-parallel to B_0. This results in a decrease in the overall longitudinal magnetization, given that these protons and those parallel to B_0 cancel each other out. In addition, the RF pulse makes the protons move *in phase*, in the same direction with each other at the same time, leading to a transverse magnetization with a vector in the x-y plane that moves in unison with the precessing protons at the Larmor frequency. When a conductive coil gets placed in the neighborhood of the transverse magnetization and voltage is induced across it, generating a current, MR signals are collected. Once the RF pulse is turned off, the protons dephase, that is, they phase out with each other and precess separately, returning to a lower energy state, i.e., the proton relaxation. There are two different forms of relaxation: transverse, also called T2, when the transverse magnetization begins to disappear; and longitudinal, also called T1, when the longitudinal magnetization begins to return to its original value [60].

B

Loss functions

The **binary cross-entropy (BCE)** is a logarithmic loss function created for multi-label tasks in which the data is confined to a binary value indicating class membership and is combined with a model that employs sigmoid functions as the final activation, resulting in a zero to one output vector. [6]. It is defined as [42]:

$$L_{BCE}(y, \hat{y}) = -(ylog(\hat{y})) + (1-y)log(1-\hat{y})) \tag{B.1}$$

where y is the expected value and $\hat{y}$ is the predicted value by the prediction model.

The **weighted binary cross-entropy (WCE)** is a variant of the BCE that may be used with multi-label or multi-class data. It is applied to skewed data and corrects the imbalance using a per-class scaling or weighting factor [6]. It is defined as [42]:

$$L_{W-BCE}(y, \hat{y}) = -(\beta \times ylog(\hat{y})) + (1-y)log(1-\hat{y})) \tag{B.2}$$

where β is used to tune the false negatives and false positives.

The **focal loss (FL)** is a kind of Binary Cross-Entropy variation that works best in situations when the classes are highly imbalanced. It incorporates a focusing mechanism into cross-entropy loss, lowering the proportional value of high-confidence predictions, allowing the model to concentrate more on learning complex examples [6].

To learn how the Focal loss is derived from the cross-entropy, it is necessary to first look at the BCE loss [42]:

$$CE = \begin{cases} -log(p), & \text{if } y = 1 \\ -log(1-p), & \text{otherwise} \end{cases} \tag{B.3}$$

Focal Loss uses the following notation to express the estimated probability of class:

$$p_t = \begin{cases} p, & \text{if } y = 1 \\ 1-p, & \text{otherwise} \end{cases} \tag{B.4}$$

As a result, Cross-Entropy may now be expressed as:

$$CE(p, y) = CE(p_t) = -log(p_t) \tag{B.5}$$

Using a modulating factor, $(1 - p_t)^\gamma$, Focal Loss recommends that simple cases be down-weighted and training be focused on hard negatives:

$$FL(p_t) = \alpha_t(1 - p_t)^\gamma log(p_t) \tag{B.6}$$

where, $\gamma > 0$ and when $\gamma = 1$ Focal Loss works similarly to a cross-entropy function, with α assuming values between [0,1] and being adjusted by inverse class frequency or regarded as a hyperparameter.

The **tversky loss** is a variation of the dice loss. It can be used if a higher sensitivity towards false negatives or false positives is needed in segmentation tasks, and the tversky index (TI) is given by the following equation [6]:

$$TI(p, \hat{p}) = \frac{p\hat{p}}{p\hat{p} + \beta(1 - p)\hat{p} + (1 - \beta)p(1 - \hat{p})} \tag{B.7}$$

where p corresponds to the expected, $\hat{p}$ corresponds to the predicted, and β is a hyperparameter corresponding to the amount of penalization given to p and $\hat{p}$. The performance of the loss function may be changed by modifying this value. The tversky loss function becomes identical to the dice loss function when $\beta = 0.5$.

The **focal tversky loss (FTL)** is similar to the focal loss, emphasizing difficult examples while down-weighting simple ones. With the aid of the γ coefficient, this loss function also tries to learn hard examples, such as with little ROIs [42]. Its equation is given by:

$$FTL = \sum_c (1 - TI_c)^\gamma \tag{B.8}$$

with TI indicating the tversky index and a value of γ between [1,3].

The **combo loss** is the combination of the weighted sum of dice loss and a modified cross-entropy [42]. This one is the most frequent choice for combining two different loss functions. The dice loss is robust against minor class imbalances, but false positives or false negatives cannot be weighted. However, two terms in the weighted binary cross-entropy function can be changed to increase or decrease the false-negative or false-positive penalty. Therefore, the combination of these two loss functions leads to a partially class imbalanced loss function, resulting in variables that are sensitive to false predictions [6].

B.1 Results using different loss functions

The simple U-net for target segmentation was considered for testing the different loss functions and evaluating their performance. The network ran for 50 epochs on a small dataset using the Adam optimizer and a learning rate of 1×10^{-6}. As the accuracy and specificity were equal or almost equal to 1, they were not presented. But, the dice coefficient and sensitivity are represented in the results to evaluate their performance in Table B.1. The presented results were obtained for the epoch that achieved the best dice coefficient for validation. The binary cross-entropy and weighted cross-entropy returned the best results at the first epoch, whereas, using the remaining loss functions, the results were obtained from the last epoch.

Table B.1: Results using different loss functions.

Loss function	**Evaluation**	**Dice Coefficient**	**Sensitivity**
Binary Cross-Entropy	Train	1.67×10^{-3}	1.54×10^{-2}
	Validation	1.74×10^{-3}	5.09×10^{-6}
Weighted Cross-Entropy	Train	1.79×10^{-3}	2.45×10^{-2}
	Validation	1.70×10^{-3}	3.60×10^{-6}
Focal Loss	Train	4.11×10^{-3}	0
	Validation	3.91×10^{-3}	0
Tversky Loss	Train	1.80×10^{-2}	5.45×10^{-1}
	Validation	1.60×10^{-2}	2.76×10^{-1}
Focal Tversky Loss	Train	1.64×10^{-2}	5.26×10^{-1}
	Validation	1.52×10^{-2}	2.81×10^{-1}
Combo Loss	Train	2.31×10^{-3}	0
	Validation	2.12×10^{-3}	0
Log Cosh Dice Loss	Train	2.43×10^{-2}	5.06×10^{-1}
	Validation	2.03×10^{-2}	2.59×10^{-1}

C

Ultrasound data inclusion

Considering the pre-processing done for the MRI images, the same was done for the US images. For these, to fit the STL file, it was necessary to rotate the US data by changing the order of the axis (x, y, and z). Afterward, the generated grid of the prostate STL file and the target are saved. When there are several targets for the same US volume, they are merged and stored.

From the spreadsheet that contains the coordinates of each biopsy, it is possible to relate the MRI to the US acquisition using the series number, which is also present in the DICOM header from both modalities. The coordinates from the tip and the base of each modality are retrieved from the spreadsheet and saved in different lists. As the coordinates from both modalities correspond to the same point, it is possible to relate them. This is accomplished by applying a function that maps one collection of point coordinates to another and, therefore, returning the transformation needed to be applied to the whole US data. After applying this transformation, the vertices from the US prostate volume are used to draw lines in the direction of the center of the MRI prostate volume, which is then intersected by the MRI prostate volume allowing get the shrinkwrap of the US prostate volume onto the MRI prostate volume, represented in Figure C.1.

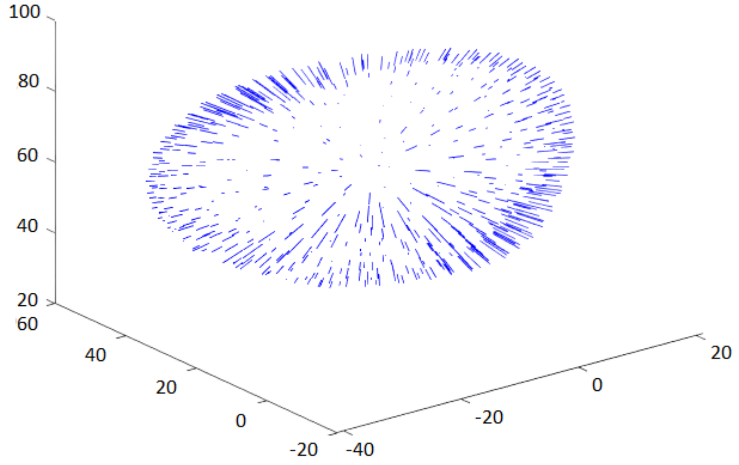


Figure C.1: Shrinkwrap of the US prostate volume onto the MRI prostate volume.

Differences in image intensities and the morphology of the underlying anatomy on scenes corresponding to different modalities and procedures make registration of multimodal imaging difficult. Ultrasound imaging of the prostate has a low soft-tissue resolution, but high-resolution MRI images of the prostate show interior anatomical features with higher clarity [62]. As it is critical to maintain the information from the data elements to obtain a consistent mapping from one method to the other and the generation of the corresponding images from each method, one issue arose due to this transformation.