Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Michal Zelenčík

Vision transformers as a support for prostate cancer detection

Master's thesis in Informatics
Supervisor: Gabriel Kiss
Co-supervisor: Frank Lindseth, Mattijs Elschot
June 2023

NTNU
Norwegian University of
Science and Technology

Michal Zelenčík

Vision transformers as a support for prostate cancer detection

Master's thesis in Informatics
Supervisor: Gabriel Kiss
Co-supervisor: Frank Lindseth, Mattijs Elschot
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Abstract

This work research the possibility of using Vision Transformers for prostate cancer detection and tumor segmentation on MRI scans. In the first part we created an architecture, which outperformed state-of-the-art models implemented in MONAI library. We opted for own implementation to have more freedom in doing architecture changes. It has "U" shape symmetrical structure with shifted window self-attention computation approach and skip connection between appropriate encoder-decoder blocks for better recovering of spatial information. In the next stage we were dealing with image registration techniques, their applicability on mpMRI channels and overall score of the model. We found out, that even though rigid and affine registration look the most natural to human eye observer, b-spline registration model achieved the best results despite that the details in the scans looks disturbed or even damaged. In the last part of the project, we investigated the multi-objective loss optimization technique, which promises to improve the generalizability of the model and avoid overfitting. A model utilizing this method achieved exceptionally good results with great potential for future work.

Sammendrag

Dette arbeidet undersøker muligheten for å bruke Vision Transformers for å detektere prostatakreft og segmentere kreft i MR bilder. Vi har laget en tilpasset arkitektur som ga bedre resultater enn toppmoderne modeller implementerte i MONAI-biblioteket. Vi valgte egen implementering for å ha større frihet til å gjøre arkitekturendringer. Den har et "U"-form og en symmetrisk struktur med forskjøvet vindu, selvoppmerksomhet beregning og tilkobling mellom passende koder-dekoder blokker for bedre gjenoppretting av romlig informasjon. I neste steg testet vi flere bilderegistreringsteknikker, deres anvendelighet til forskjellige MRI-kanaler og virkingen av å bruke registrering på sluttresultatet for modellen. Vi fant ut at selv om rigid og affin transformasjon ser mest naturlig ut for en menneskelig observatør, oppnådde B-spline -registreringsmodellen de beste resultatene til tross for at detaljene i skanningene ser forstyrret ut eller til og med skadet. I den siste delen av prosjektet undersøkte vi den multi-objektive optimeringsteknikker, som har et potensiale til å forbedre generaliserbarheten til modellen og unngå overfitting. En arkitektur med denne metoden oppnådde eksepsjonelt gode resultater med stort potensial for fremtidig arbeid.

Contents

Figures

Tables

Code Listings

Acronyms

K Key. 13–17

Q Query. 13–17

V Value. 13–17

ADC apparent diffusion coefficients. 10, 29

AI artificial intelligence. 2–4, 8, 27, 29, 47, 54, 59, 60, 64

AP average precision. 27, 35, 36, 44–48, 52, 53, 56, 58

AUROC area under receiver operating characteristic curve. 27, 35, 36, 44–48, 52, 53, 56, 58

CNN Convolutional Neural Network. 11, 19, 36, 37, 63

CNNs Convolutional Neural Networks. 2, 10–12

CT computed tomography. 1

DRE digital rectal exam. 7, 8

DWI diffusion-weighted imaging. 9, 10

FN false negative. 34

FP false positive. 33–36, 48

GELU Gaussian Error Linear Unit. 37, 38

IP image processing. 2

MHSA multi-head self-attention. 12, 17, 19

ML machine learning. 3, 5, 10, 15, 25, 26, 47, 59, 60, 62

MLP multi-layer perceptron. 12, 17, 38, 39, 45

mpMRI multi-parametric magnetic resonance imaging. 1, 2, 5, 8

MR magnetic resonance. 5, 9, 10

MRI magnetic resonance imaging. 2–4, 8–10, 12, 20–22, 24, 27, 29, 30, 49, 53, 59, 60

MTL multi-task learning. 3, 4, 24, 25, 27, 54, 61, 64

PCa prostate cancer. 2, 3, 8, 25, 26, 36

PET positron emission tomography. 1, 2, 21

PSA prostate specific antigen. 5, 7, 27

RNN recurrent neural networks. 11

ROC curve receiver operating characteristic curve. 36

SA self-attention. 2, 11, 13, 14, 18, 20, 37, 39, 43–47, 55, 59, 60, 62

SotA state-of-the-art. 2, 4, 10, 19, 20, 59

SW-SA shifted window based self-attention. 38, 44

TN true negative. 34, 48

TP true positive. 33, 34, 36

TRUS transrectal ultrasound. 7, 8

ViT Vision Transformer. 2, 4, 5, 12, 19, 36, 37, 45, 53, 59, 60, 64

ViTs Vision Transformers. 2–5, 11, 12, 20, 45, 62–64

W-SA window based self-attention. 38

Chapter 1

Introduction

Cancer is generally a very serious disease that can affect almost anyone. It is characterized by the uncontrolled growth of abnormal cells which can start anywhere in the human body. Under normal circumstances, healthy cells grow and multiply, creating new cells to replace old or damaged ones. But sometimes this process breaks down and the damaged cells begin to spread and form potentially cancerous tumors.

In order to have the best possible chance of a complete cure, it is very important to diagnose cancer at its earliest stage. There is usually no single test that can confirm cancer, but rather a series of tests and evaluations of the patient's medical history. One of the most common tests is a physical exam, where the doctor may feel lumps, see spots on the skin, or encounter enlargement of a particular organ that may indicate cancer. Urine or blood tests are also very helpful indicators, because in many cases of cancer are specific substances in the blood either increased or decreased. These methods cannot definitively confirm or refute the final diagnosis and serve only as preliminary tests that further lead to more serious procedures such as biopsies. During a biopsy, the doctor inserts a needle into the infected area and takes tissue samples for further analysis.

1.1 Motivation

Precise detection of prostate cancer is usually done by an invasive biopsy, which is not completely safe and can be accompanied with a life-threatening complication called sepsis. Simsir et al.[1] made study on 2023 patients which underwent prostate biopsy and 62 (3.06%) of them developed sepsis within 5 days after the biopsy. Shahait et al.[2] also researched the prevalence of the sepsis after the prostate biopsy and found even more severe incidence, where 9.4% out of 265 patients suffered with this complication.

One possible solution to overcome this undesirable complication while maintaining high reliability of the cancer diagnosis can be tumor detection on scans of various medical imaging procedures like computed tomography (CT), multiparametric magnetic resonance imaging (mpMRI) or positron emission tomo-

graphy (PET) scan. We are in this work focusing particularly on mpMRI which is a non-invasive method that uses strong magnetic fields and radio waves to produce detailed images of the inside of the human body. Extensive research has been carried out whether these magnetic fields and radio waves used during MRI scan could pose a risk to the human body, but no evidence has been found[3][4], making MRI one of the safest medical procedures available.

All MRI scans are processed and evaluated by radiologist experts, who can spot cancerous area. This is very responsible task that demands focus and time. Since the artificial intelligence (AI) has in the last decade leaped a giant step forward and computers are nowadays able to recognize particular objects in the images, we want to fully exploit these great capabilities and utilize then as a support for prostate cancer detection on mpMRI scans.

1.2 Goals and research questions

Artificial intelligence and image processing methods have an enormous potential to support PCa detection. It exploits quantitative properties of data that are difficult to understand for humans, but computers are able to learn these complex patterns to make predictions. For a long time has been Convolutional Neural Networks (CNNs) state-of-the-art methods that started new era in computer vision by enabling multi-channel input processing. The novelty was to convolve data by a series of filters which slide over the input to extract specific features. The output was pooled and flattened, and then passed to the fully connected layers for the final classification or segmentation.

However, the field of AI evolves quickly and recently has been proposed new approach for computer vision called Vision Transformer (ViT)[5] with self-attention (SA) mechanism that matched or outperformed state-of-the-art models in image classification and object detection. Since then ViTs become new standard for image processing with many architectures emerging particularly in medical image processing[6][7][8][9][10][11].

The main interest of this work is to use Vision Transformers for prostate tumor segmentation and create a tool that can support the work of radiologists in practice. Based on aforementioned models we speculate that ViTs can effectively detect tumors on MRI images and help to overcome possible threats of the biopsies. Our first research question therefore is:

Can be Vision Transformers used as a support for prostate cancer detection?

We will use publicly available dataset of prostate MRI scans with segmentation labels originally proposed for PI-CAI challenge[1], which are fully anonymized, so there is no danger of any information leak. Each scan consists of three channels from axial (top-down) view, which unfortunately do not depict the same region. They can have for example different zoom, they can be shifted or rotated. These

[1] see `https://pi-cai.grand-challenge.org/`

variations can have misleading effect for the neural network, as the label is always created according to only one specific channel. In order to unify all channels to depict the same region that correspond with label we need to register each scan to its main channel. There are several types of image registration and we will explore their effect on the performance of our model with second raised research question:

What is the impact of image registration on the overall performance?

In 1990s Caruana[12] proposed paper, suggesting that it may be easier for machine learning model to learn several tasks at once than learning individual tasks separately. He introduced multi-task learning (MTL) as an effective domain-specific inductive bias for training neural networks and demonstrated, that MTL can improve generalization performance even if the correlation between tasks is not so obvious.

Similarly Baxter[13] made research where the model was trained with multiple related tasks, so it was able to predict solution to several problems in given environment and the tasks acted as an inductive bias one for another. He demonstrated, that learning multiple related tasks can potentially result in much better generalization performance than learning single task.

In our dataset are also provided prostate delineation segmentations for each scan and therefore we will research the possibility of inclusion MTL approach into our model. The general idea is to create a model that will simultaneously predict tumor segmentation as well as prostate delineation. The network will have two objectives which will share some parts of the architecture and serve as an inductive bias one for another. We hypothesise that parallel optimization of these two tasks can improve predictability performance of our model and will help to prevent overfitting. In this part of the project we will take a closer look into MTL and our third research question is:

Can simultaneous optimization of multiple objectives and sharing some parts of the architecture between them improve the prediction accuracy?

Nonetheless, the main goal of this master thesis is to begin a research of using artificial intelligence models, in particular Vision Transformers, for prostate cancer detection on MRI scans with vision to create an effective and reliable tool that can reduce the workload of medical personnel and contribute to higher standard of the patient's healthcare.

1.3 Contributions

A system that is able to accurately detect and highlight prostate cancer would have massive usage in medical area. The overall aim of this project is to improve the diagnosis of prostate cancer by introducing novel and upcoming AI technologies,

which are generally better than current state-of-the-art AI tools. Our hypothesis is that Vision Transformers will lead to an AI system that significantly improves the accuracy and generalizability of prostate cancer detection on MRI scans. The envisioned system will help radiologists to make more accurate decisions and at the same time make their work easier.

The specific contributions of this project are:

1. Create an effective and accurate ViT based architecture for prostate cancer detection and tumor segmentation.
2. Examine different image registration methods and find out which one best suits our machine learning model.
3. Design multi-task learning model for simultaneous multi-objective optimization where tasks share some parts of the network and experiment, whether this model can achieve better results.

Chapter 2

Background

Artificial intelligence is a hot topic nowadays as it allows computers to build and memorize relations in the input data. This work is based on a concept of supervised learning paradigm where we have mpMRI prostate scans with segmented lesions by human expert radiologists and our goal is to create a computer program that will be able to detect cancer lesion on a new, previously unseen scan. We will establish a training process utilizing existing scans to learn relationships between mpMRI channels and segmented labels - whether the scans contain cancer, where is it located and what is its shape. The model stores its "knowledge" as a gigantic matrices and therefore it is still a great mystery to a human observer which parts of the input scan are actually important for the machine learning model.

This chapter is structured as follows. We will firstly focus on a brief description of the prostate, and prostate cancer diagnosis methods, their advantages and disadvantages. We will focus on mpMRI procedure and very basic magnetic resonance image acquisition principles. Later we will introduce not so long ago proposed machine learning architectures called Vision Transformers, their core components and base principles, as we think that model based on ViT architecture is suitable for our task. We will present some of the features from existing ViT models, their asset for image processing and usability in our project. Finally we will outline more advanced machine learning methods like image registration or multi-objective optimization and discuss their possible impact on ViT performance.

2.1 Prostate and prostate cancer

The prostate is a small, walnut-shaped gland found in men that sits below the bladder and surrounds the urethra. (see Figure 2.1). It is considered as a one of the male reproductive organs producing fluid that carry sperm during ejaculation. It produces also prostate specific antigen (PSA), which is a protein produced by a normal as well as malignant prostate cells.

When human cells contain all necessary information for their function and reproduction, they are considered healthy and work properly. In rare cases, how-

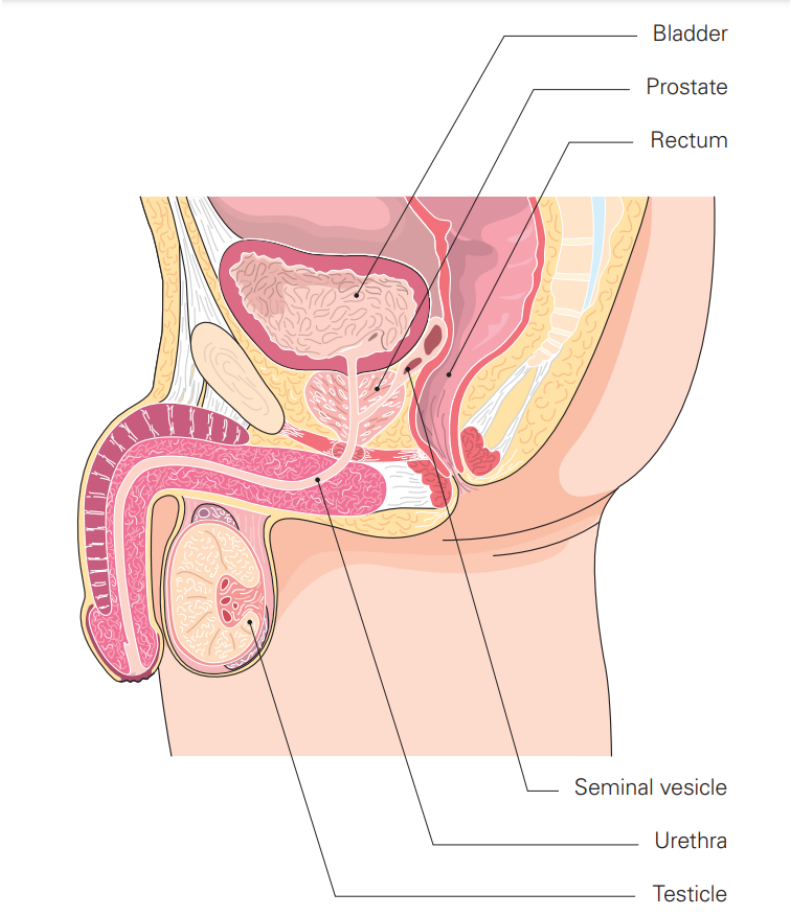


Figure 2.1: The anatomy of male genitalia

Source: [14]

ever, the cells begin to multiply uncontrollably and form a group of abnormal cells. This uncontrolled growth and forming abnormal structures, which after a while form detectable lesion is in general considered as cancer. It can affect any organ, tissue or even blood and lymphatic system. Over time can these cells invade other organs and spread across body, which is known as metastasis and typically have lethal consequences. The cancer can be slow-growing, aggressive or in most cases somewhere in between with average development. Once the cancer has been diagnosed, it is hard to estimate the growth speed and researchers are still trying to find suitable evaluation methods[14].

Saad et al.[14] further presented, that only 14% of men (in Canada) have diagnosed prostate cancer, while the results of autopsy studies revealed another 30% having latent form with passive cancer cells located exclusively in prostate. According to current data, prostate cancer is one of the few types that can remain latent for a long time before developing into a clinically significant type. The exact cause is still unknown and is the subject of further research[14].

2.1.1 Diagnosis of significant types

Diagnosis of prostate cancer can be tricky task as it often develops without symptoms and patients are feeling healthy. If the tumor stay small enough to cause noticeable health problems, it can even spread to the pelvic lymph nodes or bones without any pain[14]. Vast majority of cases is therefore detected by the routine checks, when subjects feel perfectly healthy and don't observe any symptoms. In not so trivial cases can a tumor for example continue to grow and press again urethra making it difficult to urinate. In the most advanced cases cancer metastasizes across the whole body and becomes generalized.

Digital rectal exam (DRE)

The back of the prostate touches the rectum and therefore doctors can perform digital rectal exam (DRE). DRE is the most common prostate cancer screening method, since it is easy to perform and most tumors are located in this area. It is not painful, but some patients may consider it unpleasant, as it involves sticking doctor's finger into patient's rectum and palpate prostate gland. Doctor checks for any irregularities or hardening, because under the normal circumstances is back of the prostate smooth and rubbery. These signs don't guarantee the cancer and it can be calcification or stone located in the prostate. It is not a proper diagnostic method as it doesn't allow examine whole gland, but it is a good indicator for more advanced procedures like TRUS biopsy (see below).

Prostate specific antigen (PSA) test

Prostate specific antigen (PSA) is considered as the most useful marker for prostate cancer[15]. PSA is a protein produced by prostate that helps liquefy semen and certain amount can be also find in blood. Prostate cancer causes increased levels of PSA, because the cells are more messy and more PSA will get into the blood. On the one hand higher level of PSA can be an indication of cancer, but unfortunately it can also indicate other conditions like benign prostatitis or as presented in [16], it is not rare among men with PSA levels of 4.0 (ng/ml) or less (which is considered as normal) to detect high-grade cancer with biopsy. Therefore it is, similarly like DRE, insufficient method to make final statement about diagnosis, but rather relevant sign for biopsy[17].

Transrectal ultrasound (TRUS) biopsy

Even though DRE and PSA test are useful, their results are questionable and insufficient to make final statement about the diagnosis.

TRUS biopsy is in many articles considered as a standard procedure to confirm prostate cancer[17][2][18]. It involves a doctor inserting an ultrasound instrument equipped with biopsy needle to the patient's rectum (similarly to the

DRE). This instrument transmits ultrawaves directed to the prostate, which enables to visualize prostate and help to guide needle when it pierces the rectum wall and moves toward the different areas of the prostate to collect several tissue samples[14]. Doctors usually collect up to 12 samples to minimize probability of missing the tumor[18].

One of the major issues with this procedure is the possibility of spreading post-biopsy sepsis or bacterial infection. Sometimes can be bacteria transferred on a biopsy needle directly from the rectum to the prostate causing infection. Saad et al. [14] state, that $1\% - 4\%$ of patients who undergo biopsy experience also bacteria infection which can cause general indisposition or fever. Shahait et al. [2] made study with 265 patients, where the prevalence of post-biopsy sepsis was found to be 9.4%.

Multiparametric magnetic resonance imaging (mpMRI)

There is approximately $15\% - 46\%$ cases in TRUS biopsy when doctors doesn't detect existing tumor (false positives) because of uncertain needle positioning or lesion could be small and located outside the region detectable by DRE and TRUS[17]. Also in 38% of cases is underestimated severity of the lesion, when compared with the Gleason score at prostatectomy[1][19].

Multiparametric magnetic resonance imaging (mpMRI) is a radiological imaging method to make detailed 3D images of the inside of the human body. It uses strong magnetic field, radio waves, and special computer software, which can be programmed for several different pulse sequences or parameters that highlight specific differences between tissues(see Section 2.2). In the article by Penzkofer et al.[20] is mpMRI considered as the most useful and accurate modality to detect, characterize and stage prostate cancer. Also Yakar et al.[21] conclude, that mpMRI is very promising technique for detecting and classifying PCa, but further studies need to be done to achieve standardized imaging protocols.

In general, the main goal of mpMRI is to minimize the necessity of potentially harmful biopsies, as it is considered as one of the most accurate and safest procedures. The aim of this master thesis is to step up even higher and introduce novel techniques in artificial intelligence (AI) for automated prostate cancer detection on mpMRI scans. As the AI is nowadays reaching human-like performance in many areas (for example see [22]), we hope that our system could into a large extent simplify and support work of radiologists.

2.2 Magnetic Resonance Imaging

magnetic resonance imaging (MRI) is nowadays very common procedure which uses strong magnetic fields and radio waves to provide comprehensive view of the inside of the human body without any ionizing radiation. There has been research

[1] prostatectomy is a surgical procedure for the partial or complete removal of the prostate

carried out whether MRI procedure could pose a threat to human health but no evidence has been found, classifying it as one of the safest medical procedures [3][4]. It is not, however, recommended in situations when a metal implant is fitted inside the body such as a pacemaker, bone screw or artificial joint. Someone may also feel claustrophobic in the tunnel, but most people are able to manage it with the support from radiologist.

How does MRI works

Hydrogen is a most common atom in human body and hydrogen protons with positive electric charge can act as tiny magnets responsive to outer magnetic field. MRI is a procedure by which strong magnetic fields are engaged to align these randomly oriented protons in the nuclei of the examined tissue. This alignment, however, doesn't ensure that protons will spin synchronously and a radio pulse with specific frequency must be used to make all protons spin simultaneously - in-phase. This state is also called excitation. The nuclei subsequently return to their original states through various relaxation processes during which they emit radio frequency pulses. The time and frequency of the pulses is measured and converted to particular pixel intensity by Fourier transformation. By changing the sequence of the radio pulses it is possible to create different types of MRI images[23].

In general, MRI recognizes tissues and the tissues can be recognized by two relaxation times:

- **T1** - longitudinal relaxation time - It is the time in which excited protons return to balanced state and are ready for next excitation. A T1 weighted image is obtained using short repetition time between radio frequency pulses and a short signal recovery time. A tissue with short T1 time creates strong magnetic resonance signal which is depicted as white and a tissue with long T1 recovery time creates a low intensity signal depicted as dark. For example, hydrogen atoms contained in fat has shortest T1 relaxation time, so they return to the balanced time with fastest T1 time and in the final picture will be illustrated as white[24].
- **T2** - transverse relaxation time - It determines the rate of magnetization loss after excitation. A T2 weighted image is obtained using long repetition time between radio frequency pulses and a long signal recovery time. A tissue with a long T2 recovery time creates a high intensity signal depicted as bright and short T2 recovery time produces low intensity signal depicted as dark. For example, energy transfer between hydrogen atoms in fat is more effective than in fluid, so they loses magnetization with quicker T2 time, producing low intensity signal illustrated as dark[24].

Diffusion-weighted imaging

In the MR image acquisition, diffusion-weighted imaging (DWI) is a technique for creating contrast in images based on different diffusion times of water molecules

in different fluids and tissues. The DWI signal decreases with increasing speed of water molecules travelling in direction of the magnetic field generated by MR machine. In the human body structures is the movement of water molecules usually limited or restricted in different directions - anisotropic[23].

In general, the idea behind DWI is adding two extra magnetic gradient pulses before and after radio frequency pulse. These two pulses cause that molecules moving along the field gradient direction will get phase shift, which results in MR signal loss and thus weaker MR signal. The signal attenuation depends on the degree of diffusion motion and on the strength and duration of magnetic gradient pulses. The higher motion, the higher signal loss[25].

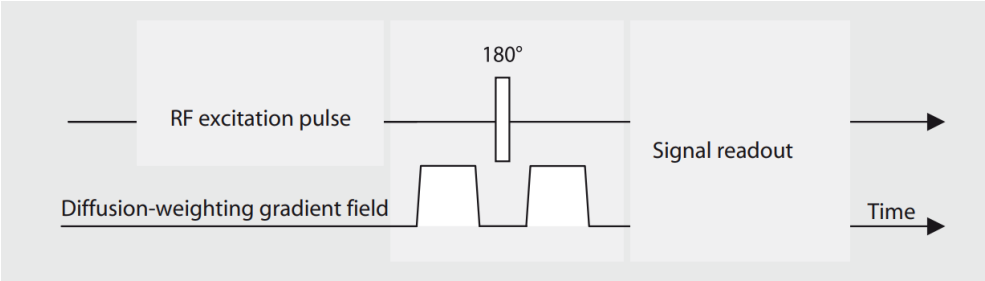


Figure 2.2: Diffusion-weighted imaging scheme

Source: [23]

The amount of diffusion weighting can be regularized with b-value, which represents amplitude, duration and time between successive magnetic gradient pulses. Higher b value produce stronger diffusion effects, but the optimal choice is not exactly specified. In practise it is possible to determine diffusion constants of unknown tissues by repeated scanning with unchanged MRI parameters, but different b values[23]. These constants are also called apparent diffusion coefficients (ADC) and represents the degree of water diffusion in particular tissues. We can also measure amount of diffusion by changing the direction of the magnetic gradient field, providing information about local geometry in more detail. Images depicting mean ADC values of particular area are called ADC maps. A bright area with reduced mobility on a diffusion-weighted image will be dark on the corresponding ADC map because of smaller diffusion constant[23]. For closer look see Figure 2.3.

2.3 Vision Transformers

Since the proposal of Imagenet[26] were all major successes in image processing based mostly on the Convolutional Neural Networks (CNNs), which became state-of-the-art models for image classification and object detection. In short, CNNs are machine learning models, which employ convolution operation to determine the content of an image. They use various filters to firstly detect low level features like vertical and horizontal edges and then in the deeper layers, they stack these

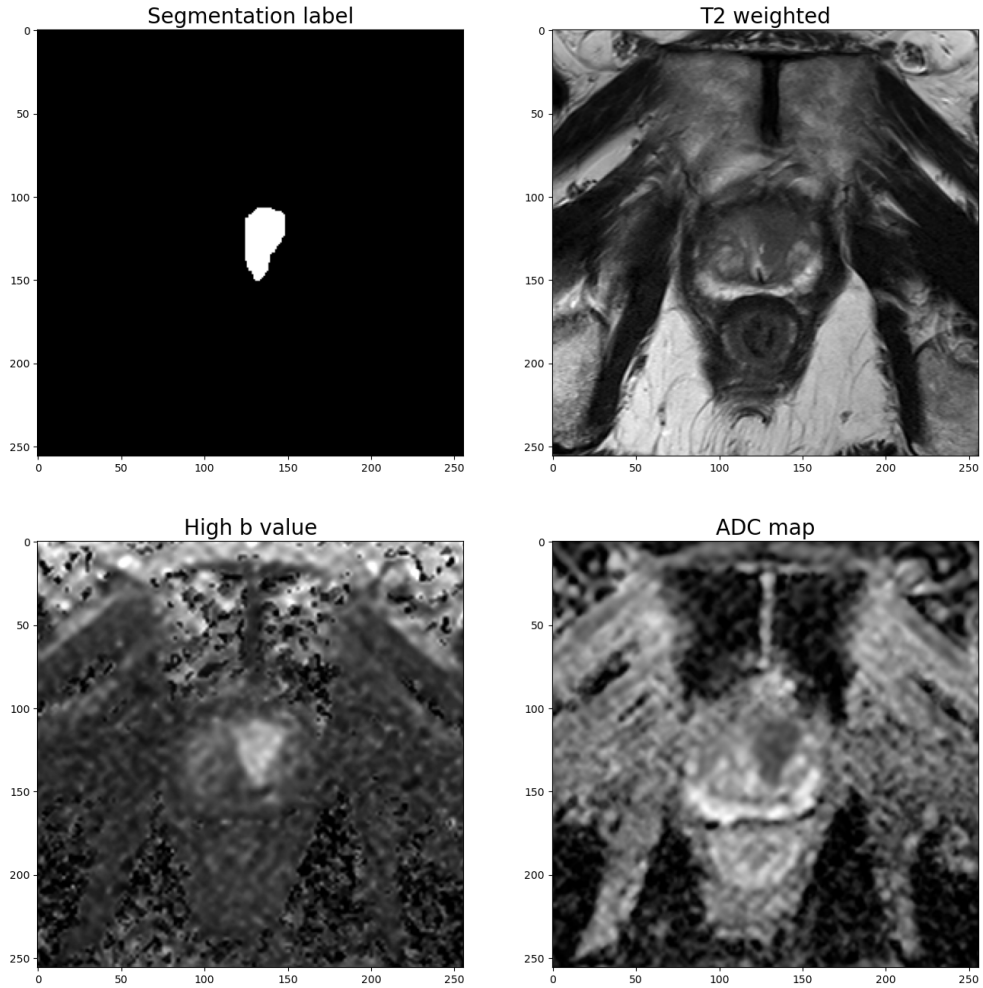


Figure 2.3: In the figure is an example of input scan. Specifically, it is one slice from an axial view of all available channels.

features into more abstract shapes, often understandable only by CNN model, which is able to learn connections between them and classify them using fully-connected layers.

Vision Transformers (ViTs) were introduced in 2021 by Dosovitskiy et al.[5] and quickly established new standard in the image processing. The authors demonstrated, that ViTs are capable of similar or even better performance as CNNs. The architecture of ViTs is based on the original Transformer model proposed by Vaswani et al.[27], which replaced recurrent neural networks (RNN) in natural language processing. The recurrence was substituted with brand new approach called self-attention (SA), which is an effective mechanism to determine relationships between input and output - it eliminates sequential input processing and allow parallelization (see Section 2.3.1). For comparison, if the input is a sentence, RNN would process it word by word, whereas Transformers would handle

it as a whole, optionally with positional encodings.

So there is a question, what is the difference between ViTs and CNNs in terms of input representation and input processing. Raghu et al.[28] analyzed these two groups of machine learning models and found quite interesting differences:

1. ViTs have more uniform representations between lower and higher layers - input in these layers is more similar
2. ViTs include more global information in the lower layers, but local information is also substantial
3. Skip connection in ViTs are even more important
4. ViTs can develop significantly stronger intermediate representations with larger pretraining datasets

2.3.1 ViT architecture

In order to begin training process, we need to provide a Transformer input data. Standard Transformer[27] is designed to process 1D vector of input embeddings. ViTs are an extension of Transformer models and therefore we need to reshape 2D, or in our case 3D MRI scans to a sequence of flattened patches. This array is successively fed into main main Transformer block, which consist of alternating blocks of multi-head self-attention (MHSA) and multi-layer perceptron (MLP). Before each MHSA and MLP is applied layer normalization[29] and after is added skip connection (for closer insight see Figure 2.4)[5].

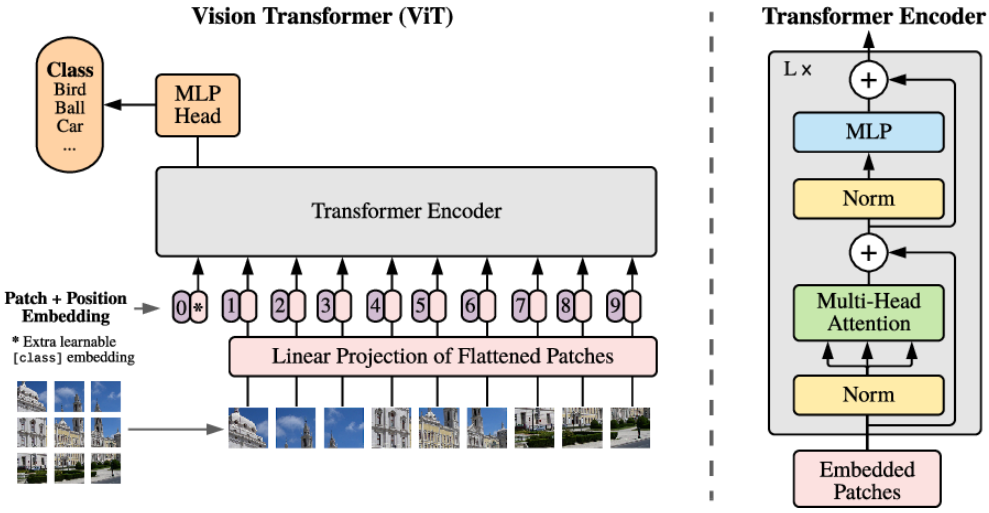


Figure 2.4: In the image is an architecture of ViT block. Image is first split into patches, which are linearly embedded to 1D vectors and fed into main Transformer block with multi-layer perceptron (MLP) head to perform classification. It is common to include into Transformer architecture design several main blocks with one final classification head.

Source: [5]

Patch Embeddings

Similarly as in many different domains, computer vision is dealing with multi-dimensional input in the form of 2D or 3D, usually multichannel, images. To process them in the Transformer it is needed to transform them and reduce dimensions to 1D multi-channel vector. Input volume is first split into non-overlapping patches of predefined shape, which can be understood as groups of neighboring pixels/voxels with all their channels.

$$\texttt{Image(D} \times \texttt{H} \times \texttt{W} \times \texttt{C)} \rightarrow \texttt{Image(N} \times \texttt{(P}_D\texttt{.P}_H\texttt{.P}_W\texttt{.C))} \tag{2.1}$$

From the Equation 2.2 we can see that each of the `N` groups is subsequently embedded into single vector with `D` number of channels (length of the vector), which is usually noticeably higher to compensate for spacial reduction.

$$\texttt{Image(N} \times \texttt{(P}_D\texttt{.P}_H\texttt{.P}_W\texttt{.C))} \rightarrow \texttt{Image(N} \times \texttt{D)} \tag{2.2}$$

Creating patch embeddings can be summarized as a process of reducing spacial resolution of the input, when the spatial information is embedded into 1D multi-channel vectors. It uses set of trainable weights, so each patch is encoded in a way that best fits the model.

Self-Attention

Attention is in the original paper[27] described as a mapping of queries and key-value pairs to an output.

Self-attention block of the Transformer takes three input representations: Query (`Q`), Key (`K`), and Value (`V`). In this context `Q` can be understood as a set of embeddings to calculate similarity for and `K` can be understood as a set of embeddings to calculate similarity against. Final attention context vector is computed as a weighted sum of `V` and weight matrix calculated by "softmaxed" similarity function between `Q` and `K` (see Equation 2.6).

A step-by-step procedure for computing self-attention (SA) runs as follows:

1. Compute similarity scores between queries encoded in matrix `Q` and keys encoded in matrix `K`.

$$\texttt{Q} \cdot \texttt{K}^T = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{m1} & s_{m2} & \cdots & s_{mn} \end{bmatrix} \tag{2.3}$$

2. Scale similarity scores to avoid problem of small gradients. Authors in the original paper[27] used scaling factor $\frac{1}{\sqrt{d}}$, where d is the length (number of

channels) of each data in the input vector X.

$$\frac{\mathtt{Q}\cdot\mathtt{K}^T}{\sqrt{d}} = \begin{bmatrix} \frac{s_{11}}{\sqrt{d}} & \frac{s_{12}}{\sqrt{d}} & \cdots & \frac{s_{1n}}{\sqrt{d}} \\ \frac{s_{21}}{\sqrt{d}} & \frac{s_{22}}{\sqrt{d}} & \cdots & \frac{s_{2n}}{\sqrt{d}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{s_{m1}}{\sqrt{d}} & \frac{s_{m2}}{\sqrt{d}} & \cdots & \frac{s_{mn}}{\sqrt{d}} \end{bmatrix} \tag{2.4}$$

3. Apply softmax function to put values of similarity matrix between 0 and 1 with sum equal 1.

$$sof\,tmax(\frac{\mathtt{Q}\cdot\mathtt{K}^T}{\sqrt{d}}) = sof\,tmax\left(\begin{bmatrix} \frac{s_{11}}{\sqrt{d}} & \frac{s_{12}}{\sqrt{d}} & \cdots & \frac{s_{1n}}{\sqrt{d}} \\ \frac{s_{21}}{\sqrt{d}} & \frac{s_{22}}{\sqrt{d}} & \cdots & \frac{s_{2n}}{\sqrt{d}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{s_{m1}}{\sqrt{d}} & \frac{s_{m2}}{\sqrt{d}} & \cdots & \frac{s_{mn}}{\sqrt{d}} \end{bmatrix}\right) \tag{2.5}$$

   At this point we have matrix which expresses which data in Q are most similar to data in K.
4. Multiply V with similarity scores from previous step. This will produce final output representation for every entry in V.

$$Attention(Q,K,V) = sof\,tmax(\frac{Q\cdot K^T}{\sqrt{d}})\cdot V \tag{2.6}$$

Source: [27]

Unlike RNNs or LSTMs which have available only one or several previous data points, self-attention processes all inputs at once in parallel, which gives it huge benefit.

Multi-head self-attention

In addition to standard self-attention, Vaswani et al.[27] found it beneficial to linearly project Q, K and V several times with different learned linear projections. On each of these projections they subsequently execute standard self-attention function obtaining multidimensional output, which is subsequently and projected to yield final values (see Figure 2.5 *right*).

$$\begin{aligned} Multihead(\mathtt{Q},\mathtt{K},\mathtt{V}) &= concat(head_1,\ldots,head_h)\cdot W^O \\ \text{where } head_i &= Attention(X\cdot W_i^{\mathtt{Q}}, X\cdot W_i^{\mathtt{K}}, X\cdot W_i^{\mathtt{V}}) \end{aligned} \tag{2.7}$$

Source: [27]

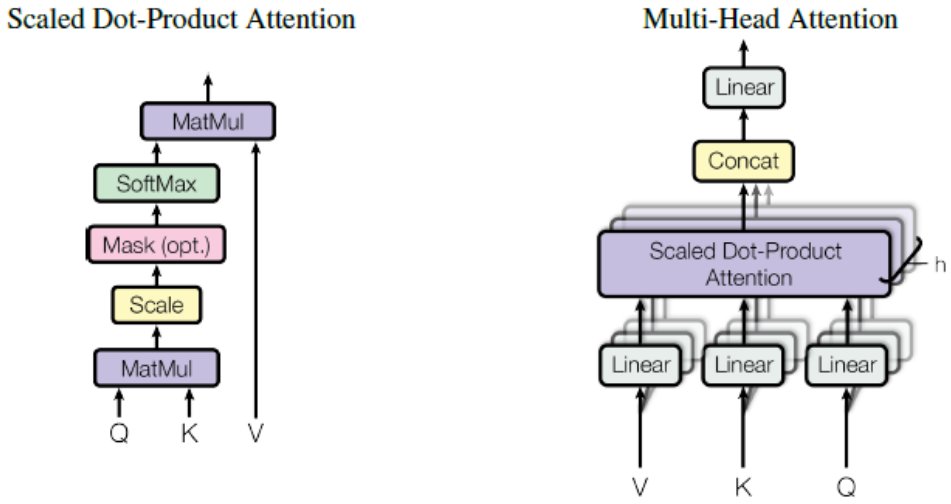


Figure 2.5: Self-attention

Source: [27]

The idea behind multi-headed self-attention is to extract multiple features from the input that would be impossible to get only with one head. Or in other words each head can specialize to extract specific feature. Multi-headed self-attention procedure runs as follows:

1. Compute linearly projected version of `Q`, `K`, `V` input vectors

$$\mathtt{Q} = X \cdot W^{\mathtt{Q}} \tag{2.8}$$

$$\mathtt{K} = X \cdot W^{\mathtt{K}} \tag{2.9}$$

$$\mathtt{V} = X \cdot W^{\mathtt{V}} \tag{2.10}$$

2. Perform standard self-attention function for each head
3. Concatenate outputs from each head
4. Multiply concatenated outputs with weight matrix W^O to get final output

2.3.2 Encoder-Decoder structure

Similarly to most of the advanced machine learning models, original Transformer[27] has also encoder-decoder architecture. The role of the Encoder is to extract features from the input sequence which are processed by the Decoder to produce desired output.

Encoder

One Encoder block consists of `N` layers and each of them is composed from the following components:

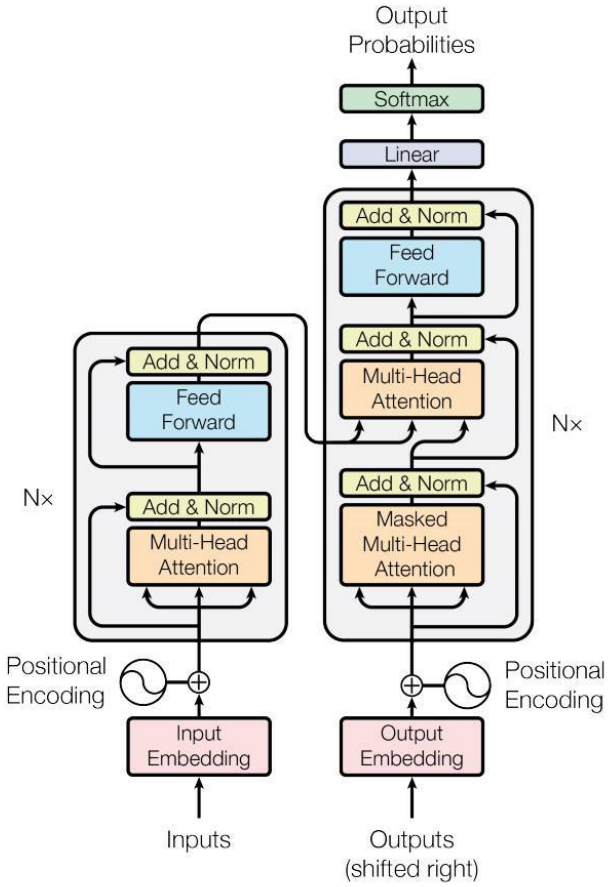


Figure 2.6: In the figure is shown Encoder-Decoder structure of the original Transformer[27]

Source: [27]

1. Multi-head self-attention (MHSA) mechanism with `h` heads. Each head receives differently projected input `X` in the form of `Q`, `K`, `V` matrices and produce output specialized on a specific task.
2. Multi-layer perceptron (MLP) that is applied independently to each position and contain two linear layers with ReLU activation in between. Linear transformations are the same for all input points i.e. the number of input neurons is equal to the number of channels, but they differ from layer to layer.

$$\text{MLP}(\mathtt{X}) = ReLU(W_1 \cdot \mathtt{X} + b_1) \cdot W_2 + b_2 \tag{2.11}$$

Source: [27]

3. Around two previously mentioned blocks are employed residual skip connections[30]. He et al.[30] claim, that residual skip connections could contribute to easier optimization of the network and network can achieve better final accuracy.
4. Each block is further followed with layer normalization[29]. Layer normalization was designed to overcome shortcomings of batch normalization, in particular the problem with mini-batch size or its hard applicability in RNNs. Unlike in batch normalization, in layer normalization all hidden units of a layer share normalization terms, but different training cases have different normalization terms. In addition, layer normalization does not have any restriction on the mini-batch size and therefore it can be used with batch size 1, which is beneficial especially for transformers as they demand large space for training and reducing the batch size can help to save the memory[31][29].

A quite important thing to notice is that since Encoder process all inputs in parallel, it is unaware of the relative positions of the data in the sequence. Positional information has to be manually injected in the form of positional encodings, or in our case images have to be sliced to the embedding vectors everytime in the same way i.e. patches of the image has to be sorted in one specific manner.

Decoder

Decoder has very similar architecture as the Encoder:

1. First block in the decoder is masked multi-head self-attention. While the encoder is designed to access all inputs in parallel regardless their position, masking operation ensures, that output on the position `i` can depend only on the known outputs up to position `i` and not further. This is achieved by applying mask on the similarity scores produced by matrix multiplication `Q` and `K`.

$$masked(\mathtt{Q} \cdot \mathtt{K}^T) = \begin{bmatrix} s_{11} & -\infty & \cdots & -\infty \\ s_{21} & s_{22} & \cdots & -\infty \\ \vdots & \vdots & \ddots & \vdots \\ s_{m1} & s_{m2} & \cdots & s_{mn} \end{bmatrix} \tag{2.12}$$

2. Second block is the standard MHSA, similar to the one in the encoder. It receives Query (`Q`) from the previous block, while Key (`K`) and Value (`V`) is received from the encoder block on the same level (see Figure 2.6).
3. Third block is the multi-layer perceptron (MLP) similar to the one in the encoder.
4. Each of the three aforementioned blocks is surrounded with residual skip connection and followed with layer normalization.

2.3.3 Shifted windows

Similarly to original Transformer[27] designed for natural language processing, Vision Transformer[5] developed for image processing also incorporates global self-attention (SA) mechanism, where the similarity scores are computed each-to-each between all inputs. This nature leads to quadratic complexity with respect to the length of the input sequence (number of input entries), making it unfeasible for images with higher resolution. Liu et al.[32] proposed effective approach to deal with this problem, which brings self-attention (SA) computation into local, non-overlapping regions - windows, which are arranged to evenly partition the image.

The main idea behind this approach is undeniably to reduce SA computational complexity, while at the same time the fact that there is probably very little connection between the pixels in one corner and the pixels in the opposite corner. Since each window contains predefined number of patches $M \times M$ (in 2D image), this method can reduce complexity of the general self-attention with quadratic complexity of:

$$\Omega(MSA) = 4hwC^2 + 2(hw)^2C \tag{2.13}$$

Source: [32]

to

$$\Omega(W-MSA) = 4hwC^2 + 2M^2hwC \tag{2.14}$$

Source: [32]

where $h \times w$ is the number of patches. From the Equation 2.14 we can see that window based SA has linear complexity when the window size M is fixed, making Transformer model variations usable for images with high resolution or for 3D image volumes, where complexity grows with spatial resolution even more aggressive.

However, this approach in the form described so far is not complete because it doesn't include inter-window information which is important for the full modelling power. To deal with this issue authors introduced shifted window partitioning system which involves two partitioning configurations in a consecutive manner. As illustrated in Figure 2.7, the first configuration uses standard window partitioning strategy starting from top-left, whereas the second configuration uses window configuration that is shifted from the original placement by $[\frac{M}{2}, \frac{M}{2}]$.

The whole procedure with shifting windows runs as follow[32]:

1. $\text{output}^i = \textbf{W-MSA}(norm(\text{output}^{i-1})) + \text{output}^{i-1}$
2. $\text{output}^i = MLP(norm(\text{output}^i)) + \text{output}^i$

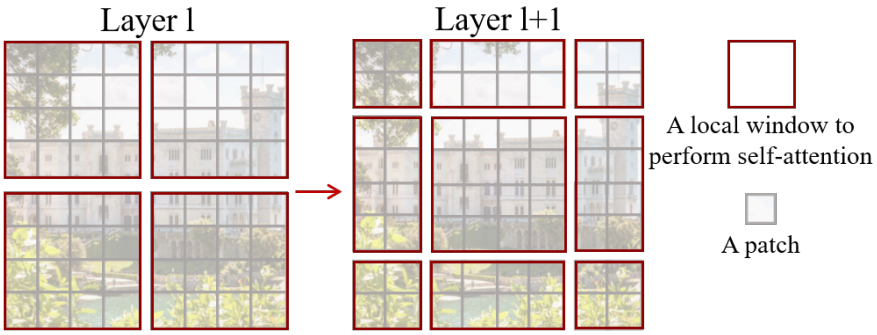


Figure 2.7: Shifted window approach for computing self-attention. Image patches are first divided into windows, attention scores are computed individually in each window, windows are than shifted and whole procedure is repeated.

Source: [32]

3. $\text{output}^{i+1} = \textbf{SW-MSA}(norm(\text{output}^{i})) + \text{output}^{i}$
4. $\text{output}^{i+1} = MLP(norm(\text{output}^{i+1})) + \text{output}^{i+1}$

where *W-MSA* is window based multi-head self-attention and *SW-MSA* is shifted window based multi-head self-attention.

Authors proved, that this approach is very effective in image classification, object detection and semantic segmentation and their model called Swin Transformer[32] achieved state-of-the-art performance on COCO object detection and ADE20K semantic segmentation.

2.3.4 "U" shape structure

Inspired by success of CNN based U-Net[33] network which achieved good results in various medical segmentation tasks, there has been several ViT architectures[6][7][8][9] adopting its "U" shape structure. The main idea of the U-Net is to supplement classic contracting path consisting of alternating layers of convolutions, activation functions and pooling operations with symmetrical expanding path, where pooling operation is replaced with up-sampling.

The role of the contacting path which is also known as an encoder branch, is to reduce spatial resolution and expand feature information usually represented as a number of channels. During each pooling operation is spatial resolution lowered to one half and number of channels is doubled. In the expansive path, known as a decoder branch, is low-resolution, high-dimensional output subsequently expanded with up-sampling operation and combined it with high-resolution features from the encoder block on the same level to yield more precise segmentations. Up-sampling operation, symmetrically to pooling, doubles spatial resolution and shrinks number of channel to half. Ronneberger et al.[33] highlight the benefit of combining low resolution features with rich contextual information and high dimensional features from the encoder, where the spatial information is better encoded.

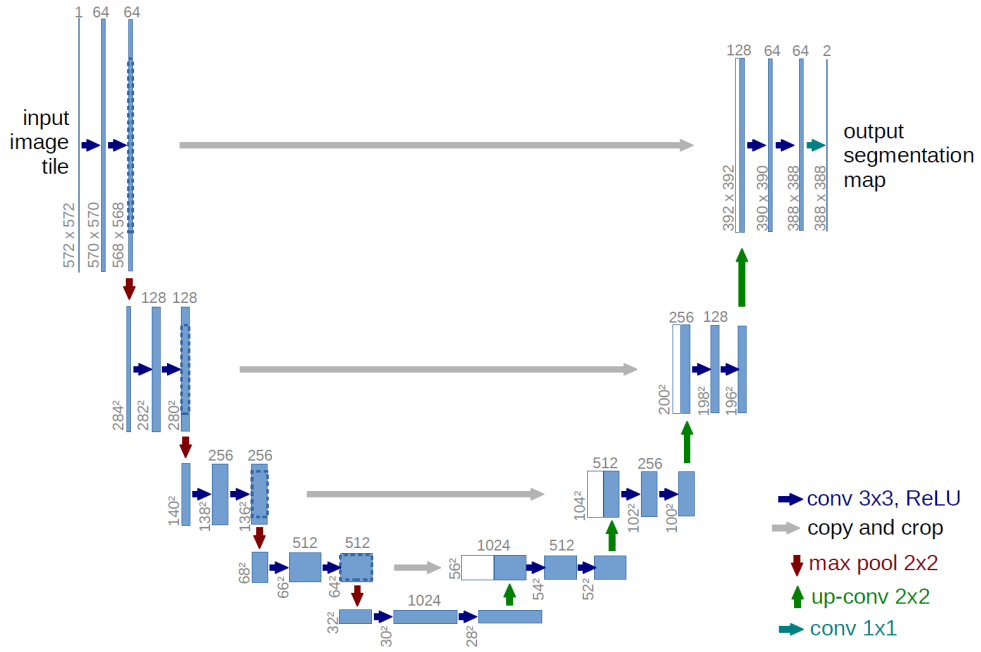


Figure 2.8: U-Net architecture

Source: [33]

In case of Vision Transformers (ViTs) it is possible to to utilize this principle by replacing convolution operation with self-attention (SA) mechanism. The encoder branch than likewise consists from several encoder blocks, which have a task to extract deep feature representations. These representations are further feed into decoder branch, up-sampled and similarly merged with features from the encoder via skip connections to restore spatial information[6]. Some models also include bottleneck block in the most bottom layer which has same purpose as an encoder block, but without any skip connection emerging to a decoder.

Vision Transformers with U-Net structure[6][7][9] achieved excellent to state-of-the-art performance in medical image segmentation tasks and therefore we also acquire this structure to our model for prostate tumor segmentation.

2.4 Image registration

Our dataset was acquired from multiple patients at different times and with different MRI machines. Image registration is a fundamental technique of unifying multiple image data to the same coordinate system[34][35] i.e. it is used to align images or image channels taken from possibly different sensors at different time points[36] with some kind of object deformation or translation. For instance heart or lungs with the cardiac or respiratory cycles results in the change of organ position and deformation[37]. Prostate is a static organ anchored in pelvis and therefore the scans are not as much deformed, however, every patient is positioned

slightly differently and it is also likely that patients could during MRI procedure move. This subject movement cannot be completely eliminated in the scanner and therefore consecutive correction is needed as a pre-processing step[38]. Registration process includes alignment of these distinctions into a common coordinate system, so every pixel in every scan represents matching biological prostate point in the same relative location.

There has been conducted extensive research for medical image registration and its benefits. For example in[39] authors did registration of MRI and PET brain images with tumor and this alignment found as an important factor in interpreting high-resolution PET images. Eberl et al.[40] measured accuracy in the Hoffman brain phantom studies with conclusion that registration simplifies comparison of the data acquired from different machines at different time frames. Mäkelä et al.[41] consider registration as a preliminary and necessary step to compare anatomical cardiac information and that aligning these images into common reference frame allows more comprehensive analysis of cardiac functions. As it is common that prostate MRI scans have various shift according to the exact position of the patient, we hypothesize that registration would be beneficial in order to integrate this dissimilar data and our model will be able to detect lesions with higher accuracy.

2.4.1 Definition

In the image registration we have one fixed image I_F which is used as a template and a moving image I_M which needs to be registered i.e. aligned to the I_F in terms of space and voxel intensity. The formal definition of 3D image registration then could be defined as a mapping between I_F and I_M in the following way[34]:

$$I_M(x,y,z) = g(I_F(f(x,y,z))) \tag{2.15}$$

where f is a voxel-wise 3D spacial mapping transformation and g is a voxel-wise 1D intensity transformation, which is mostly needed only in case of for example sensor type change, or in a case of view angle change when light hits the surface with different glare[34].

Klein et al.[36] have defined it as an optimization problem, where cost function is optimized with respect to the spatial transformation $f(x,y,z)$. In general, it is a process of finding appropriate transformations by deforming and aligning moving image I_M to match fixed image I_F. To measure relationship between these two images the similarity metric is used i.e. loss objective - a quantitative criterion to be optimized, which tells how well is moving image I_M matching fixed image I_F. Its derivative indicates in which direction we should move I_M for better alignment. We can define this metric for anything we want to optimize, for example spacial location, pixel intensities or any other feature[42].

An example of such simple linear image registration can be seen in the Figure 2.9.

2.4.2 Registration types

Registration type is an essential element for determining the class of transformations and mappings between images.

1. **Rigid** - This is the most fundamental type of registration, where objects retains their relative shape and size as it consist only from rotation, translation and zoom operations and except for addition of translation vector it preserve linear properties of the object[34]. It is a subset of a more general affine transformations described below. It doesn't change the essence of specific MRI image and preserve internal structure of the scan. It doesn't take into account internal prostate deformations and differences. An example of this registration type can be seen in the Figure 2.9.

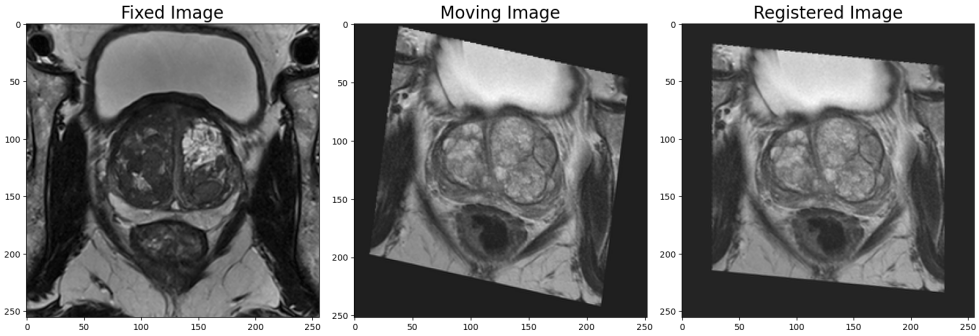


Figure 2.9: In the figure can be seen rigid registration, where moving image is aligned to the fixed image with rigid registration type.

2. **Affine** - This group, in addition to rigid type, include also shearing and scaling along given axis. It is able to complement for more advanced deformations, while keeping affine mathematical properties - it projects parallel straight lines into parallel straight lines.

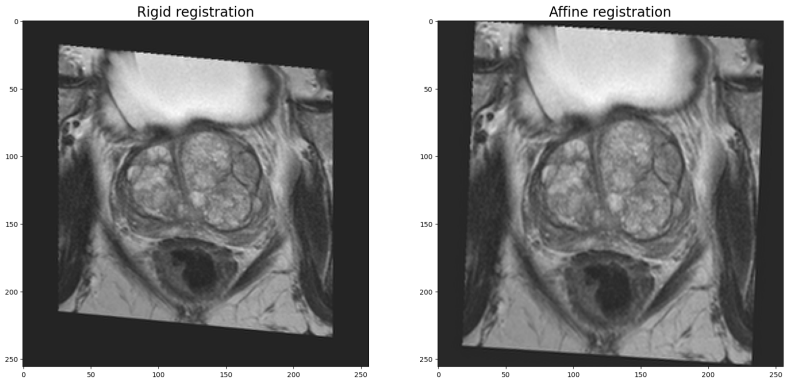


Figure 2.10: In the figure is presented comparison between rigid and affine registration results of the same fixed and moving image from the Figure 2.9.

3. **Non-rigid** - This type of registration is need when comparability between two images can not be achieved without any local deformation for example because of some biological differences or image acquisition technique[43]. This naturally results in many degrees of freedom and parameter space has likewise more dimensions. These algorithms are usually based on splines[2] and many of them either include rigid/affine transformations or are run after them[37]. The examples of non-rigid registration (affine followed with b-spline) can be seen in Figure 2.11 and Figure 2.12.

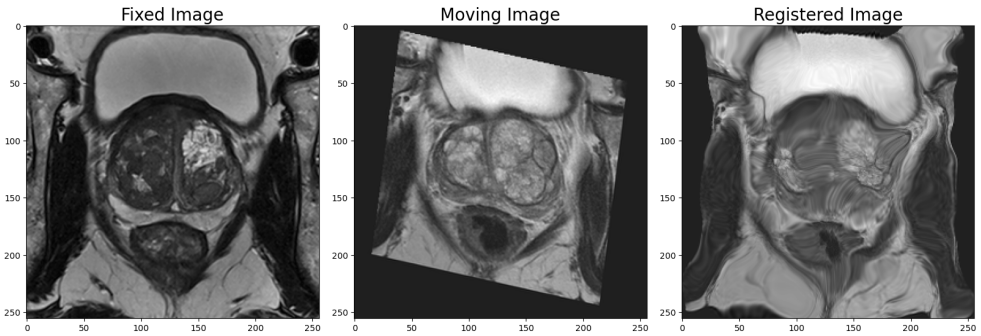


Figure 2.11: In the figure is presented non-rigid registration example between two different images, where internal biological structures of the prostate were significantly warped. The algorithm was able to perfectly align upper part where the contours were clearly outlined, however, in the middle part with a lot more noise it created unwanted deformations.

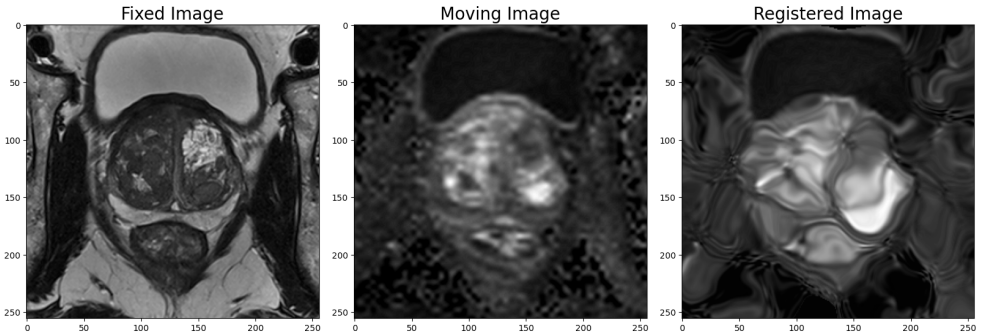


Figure 2.12: This is a non-rigid registration example of the `hbv` channel (moving image) to the `t2w` channel (fixed image) both belonging to the same scan. The algorithm again made unwanted deformations in the center area and considerably damaged internal information.

4. **Groupwise** - This is a single optimization procedure across many images stacked in the image vector usually based on the concept of mutual information[45]. It is designed to account for dissimilarities spread across the whole dataset eliminating bias towards chosen fixed image.

[2] spline - piecewise polynomial function[44]

2.5 Multi-Task learning

The general approach in machine learning is to improve one model for one task to perform as good as possible. A big problem with this technique, however, is the selection of inductive bias. This means, how to make model super-robust to produce generalized and accurate predictions and at the same time how to prevent it from being over-fitted to the limited number of training samples[13].

Human beings are able to train and improve in several tasks at once. Experience or knowledge from one task may help to improve the other. For example weight lifting may improve endurance performance in cycling. Machine learning draws inspiration from this approach in multi-task learning (MTL) mechanism, which has potential to improve generalization performance of the model and can act as an inductive bias. It uses simultaneous optimization of multiple domain-specific tasks which share some parts of the architecture, so the signals from one task works as an inductive bias for the other i.e. what is learned by one task could help the other task to train better[46], thus improving generalization performance of all tasks at the same time. MTL is formally defined as follows

> Given m learning tasks $\tau_{i=0}^{m}$ where all the tasks or a subset of them are related but not identical, multi-task learning aims to help improve the learning of a model for τ_i by using the knowledge contained in the m tasks[47][48].

Zhang et al.[47] further pointed on two factors that emerge from this definition, specifically task relatedness and the definition of the task, which should be considered during the design of a MTL model.

The main concern and the main task of this master thesis which we aim to improve is prostate tumor segmentation. It is single, very difficult task of recognizing complex patterns in the 3D input MRI scan. To fully exploit MTL approach we will propose second Decoder branch for prostate zonal segmentation. Both of these task are problems from supervised learning paradigm, where model is trying to find mapping between input scans and segmentation labels. Formally, we have 2 supervised learning tasks τ_i, where each is associated with our dataset, which consists of 3-channel 3D input scans X, and two sets of 1-channel 3D labels y_i. In supervised MTL learning we then aim to improve two functions $f_i(x)$ such that $f_i(x)$ is a good approximation of y_i[47].

Although there are several categories of MTL[47][48], we will only focus on deep-learning feature-based MTL approach which is the scope for this work. It is based on an assumption, that given tasks have similar feature representation encoded in the shared hidden layers of the network[47]. This part of the network is also called feature extractor and its purpose is to extract features from the input data. It is followed by task-specific heads, which further process these features to make predictions. We will further present our idea and architecture in Section 4.5.

2.5.1 Hard and Soft parameter sharing

Within the area of deep-learning is commonly used either hard or soft type of MTL parameter sharing.

1. **Hard parameter sharing** is a most common technique proposed by Caruana [12], where all tasks share some fixed, usually bottom part of the network, followed by separate task-specific output layers.
2. **Soft parameter sharing** is a technique at which tasks doesn't share any layers and each task has a single model with corresponding parameters, but the parameters in some specified layers are regularized between tasks - stimulated to be similar[49].

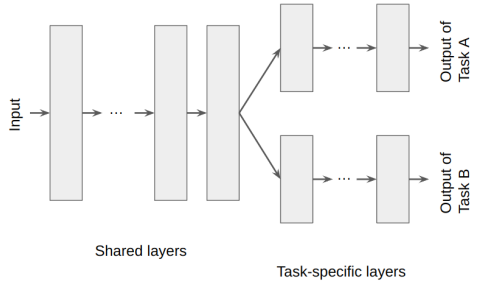


Figure 2.13: Hard parameter sharing principle

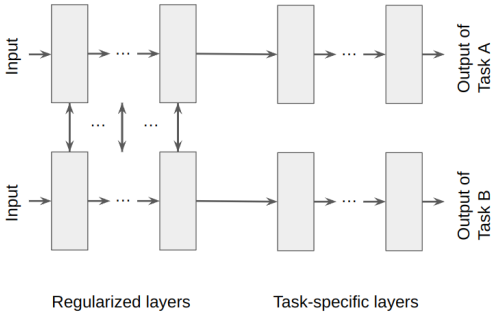


Figure 2.14: Soft parameter sharing principle

2.6 Related work

This thesis has the same objective as the PI-CAI challenge[3] and therefore the most relevant related works to this master thesis are just the ones proposed for it. PI-CAI (Prostate Imaging: Cancer AI) is a competition that aims to validate different machine learning algorithms and evaluate their performance in PCa detection against labels produced by radiologists. In this section we provide an overview of distinguishing features from the 5 most successful works which share some pre-processing and post-processing routines provided by event organizers.

Z-SSMNet: A Zonal-aware Self-Supervised Mesh Network for Prostate Cancer Detection and Diagnosis in bpMRI[50]

In the first work[50] authors made quite broad model showed in Figure 2.15. They based it on the MNet[51] which was designed to balance various spacing information through alternating 2D convolutions in different axes and 3D convolutions.

[3] see `https://pi-cai.grand-challenge.org/`

The importance of each convolution is further encoded with other learnable parameters so the model can decide, which axis of the volume is important in given processing step to convolve. Authors of MNet claim that this approach can make it easier for machine learning model to learn from image with various spacing between axes.

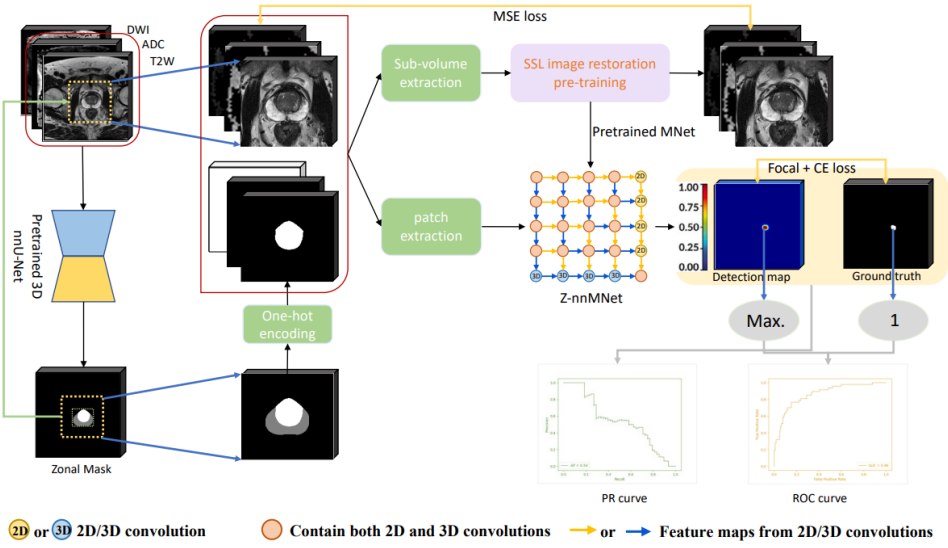


Figure 2.15: In the figure is an overview of Z-SSMNet model that consists of zonal mask generation part, self-supervised pre-training part and mesh network part for PCa detection.

Source: [50]

The first interesting feature of Z-SSMNet is the zonal mask generation. Authors trained standard 3D nnU-Net[52] to predict prostate zonal segmentations to guide the network for learning region-specific information and to more precisely crop the region of interest (prostate). They subsequently used this region with added 2.5cm for region-of-interest cropping of scans on the contrary to basic centre crop. For this task they used 3 external publicly available datasets with prostate zonal segmentation labels.

The second thing worth mentioning that authors used to tune their model is self-supervised pre-traing routine described in [53]. This method incorporates various data augmentations techniques of the input where the networks is train to recover original channels to specialize weights of the model for given task before actual training. It was demonstrated that it significantly outperforms learning from scratch with random weights.

Due to heterogeneous nature between scans from multiple facilities authors integrated pre-trained network into the nnU-Net framework to form the Z-nnMNet that can pre-process data in an adaptive way.

Their ranking score achieved on the validation set was 0.800 points.

Deep learning for detection and diagnosis of prostate cancer from bpMRI and PSA: Guerbet's contribution to the PI-CAI 2022 Grand Challenge

Authors of the second work[54] described their progress in several steps:

1. They used provided prostate delineations made by AI and trained nnU-Net[52] to predict prostate segmentation and cropped scans along the area of interest in the pre-processing phase.
2. In the second step they used modified version of the nnU-Net[52] to predict prostate delineation and also tumor segmentation.
3. Additional pre-trained Retina model[55] was used to detect tumor lesions.

Final tumor segmentation was done by the nnU-Net from step 2, but corresponding probabilities were computed by ensembling output probabilities from step 2 with detection score from step 3 as well as PSA value provided as additional information.

They claim that their model achieved on the validation set AUROC of 0.854 and AP of 0.489 which results in final score of 0.672 points.

Prostate Lesion Estimation using Prostate Masks from Biparametric MRI

In the third work[56] authors experimented with nnU-Net[52]. They for example used prostate delineations as a fourth input channel for more accurate tumor localization or in another experiment they made an ensemble model from five nnU-Nets. They also experimented with using clinical markers such as PSA value and ADC intensity values for more precise benign/malignant tumor classification and false positives elimination.

Authors made many experiments with various pre-processing routines and achieved ranking scores from 0.734 to 0.770 with single model and scores from 0.712 to 0.810 with ensemble models. Their best model was ensembled nnU-Net semi with prostate gland delineations as an additional input, cropped images, evaluated PSA values and evaluated ADC maps.

The Prostate Imaging: Cancer AI (PI-CAI) 2022 Grand Challenge (PIMed Team)

The model that finished fourth[57] uses SPCNet[58] - a convolutional neural network proposed to detect aggressive cancer, indolent cancer, and normal tissue on MRI scan. As the authors weren't satisfied with high number of false positives, they added custom decision head which simply outputs yes/no decision making from it multi-task learning model. This new model called SPCNet-Decision with two objectives clearly outperforms original SPCNet.

One interesting thing is that authors trained ProGNet[59] on prostate gland segmentations and during backpropagation computed gradients only within the prostate gland boundaries. For the final prediction they ensembled outputs from the standard 3D UNet with residual connections[33] and SPCNet-decision models.

The aforementioned ensembling model achieved ranking score 0.773 points.

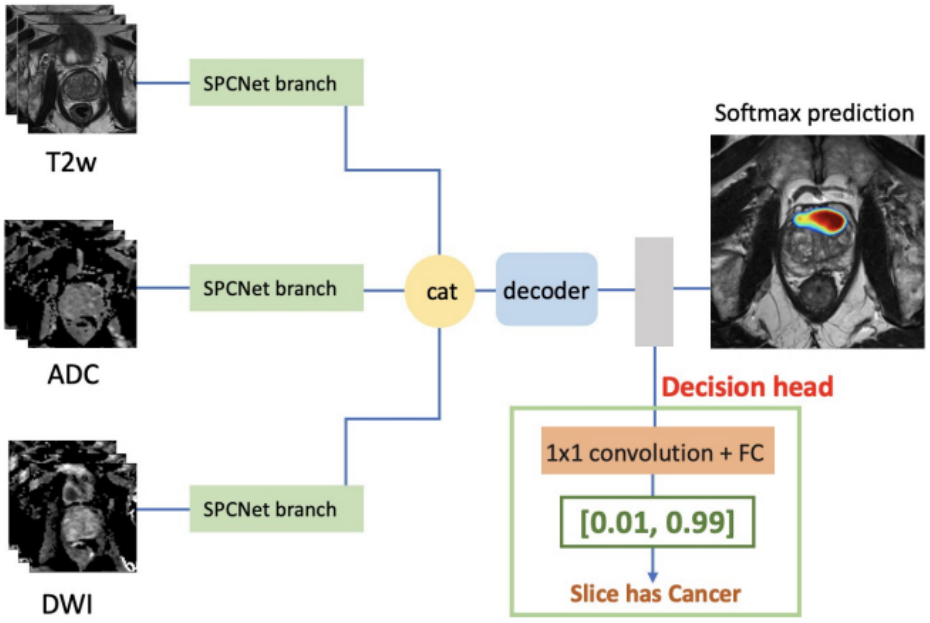


Figure 2.16: In the figure is an overview of SPCNet-Decision model that consists of tumor segmentation part and decision head.

Source: [57]

Implementation method of the PI-CAI challenge (Swangeese Team)

In the fifth work[60] authors used preprocessing tools provided by event organizers and trained segmentation and classification networks. For the semantic segmentation they used ITUnet[61] that was originally a network structure designed for organ segmentation tasks of clinical medical images and changed it from 3D to 2D. For the image classification they use EfficientNet-b5[62] without any further specification how they did the predictions.

Their model achieved ranking score 0.784 on the validation dataset.

Chapter 3

Methods

This chapter provides detailed insight into our working procedures and aims to give basic understanding of our intentions. Firstly we will present techniques used for data pre-processing and result evaluation which were necessary to unify obtained data and quantify the performance of researched models. Later in this chapter we will introduce the baseline model originally developed in the preparatory project for this master thesis[31], against which we will compare quantitative results from our experiments.

3.1 Dataset

For this work we are using public dataset[63][64] originally proposed for PI-CAI challenge (Prostate Imaging: Cancer AI)[1]. This dataset consist from 1500 anonymized MRI scans obtained between 2012-2021 at Radboud University Medical Center, University Medical Center Groningen and Ziekenhuis Groep Twente.

1294 out of 1500 samples have assigned tumor segmentation labels made by human experts and 1499 scans have tumor segmentation labels generated by artificial intelligence. AI generated labels are, however, not verified by human experts and their validity is therefore questionable.

Every scan in the dataset have 5 channels:

1. **t2w** - Axial - top-down T2 weighted image
2. **sag** - Sagittal - left-right T2 weighted image
3. **cor** - Coronal - front-back T2 weighted image
4. **hbv** - Axial high b-value ($\geq$ 1400 s/mm2) diffusion-weighted image
5. **adc** - Axial apparent diffusion coefficients map

We will for our purposes use t2w, hbv and adc channels, dropping sagittal and coronal T2 weighted images due to their different resolution.

[1] See `https://pi-cai.grand-challenge.org/`

3.2 Data pre-processing

Data pre-processing is an essential step in creating a machine learning model. It is generally known, that neural networks could achieve better performance if the raw input data are transformed to united format, so each part of the network can better specialize to detect specific feature. As described in the section above, our dataset comes from different places and different MRI machines, so it is not surprising that provided scans have various resolution, zoom or exposure. It depends on their origin as well as on the capabilities of radiologists that generated them. We will use several basic pre-processing techniques implemented in `picai_prep`[65] package to transform scans to similar appearance:

Resampling

First thing to notice is that different scans have different resolution caused by various amount of zoom. On top of that, hbv and adc axial view channels within one scan have also different resolution than the main t2w channel. To deal with these issues we first need to resample - upsample or downsample resolution of hbv and adc channels to match the resolution of t2w image.

The implementation in `picai_prep`[65] package uses `ResampleImageFilter()` class from `SimpleITK`[66][67][68] toolkit with b-spline interpolator for hbv and adc channels and nearest neighbor interpolator for the label (see Code listing 3.1). We also provide example of input channels resampling in the Figure 3.1.

Code listing 3.1: Scan resampling function

```
def resample_to_first_scan(self):
    """Resample scans and label to the first scan"""
    # set up resampler to resolution, field of view, etc. of first scan
    resampler = sitk.ResampleImageFilter()  # default linear
    resampler.SetReferenceImage(self.scans[0])
    resampler.SetInterpolator(sitk.sitkBSpline)

    # resample other images
    self.scans[1:] = [resampler.Execute(scan) for scan in self.scans[1:]]

    # resample annotation
    resampler.SetInterpolator(sitk.sitkNearestNeighbor)
    if self.lbl is not None:
        self.lbl = resampler.Execute(self.lbl)
```

Source: `picai_prep`[65] package

Interpolation is a process of estimating new data based on the existing information. For example if we know that the value at point "A" is 3 and the value at point "C" is 9, we can estimate the value at point "B" by linear interpolation to 6. In the image processing area if is often related to reconstructing analogous or continues image signal from existing digital or point-based image. It can be used for image

enlargement, when we try to estimate best color or intensity for the new voxels based on their surrounding[69].

The nearest neighbor is a basic interpolation method, whose principle is to take closest value to desired voxel. It is the simplest approach and require shortest processing time. Its advantage is preserving original voxel intensities, but the resulting image have pixelated appearance - each point appear to be bigger[38]. These properties are ideal for label resampling, when we want to preserve binary intensity character in cost of not perfectly round edges.

The exact explanation of B-spline interpolation method is, however, beyond the scope of this work. In short, the image is first transformed to the image of basis coefficients and then is at each new voxel computed linear combination of these basis functions[38] to best fit the template. It is especially useful when the image is likely to require multiple rotations, translations or distortions in several specific steps. For the exact definition please read *Briand and Monasse*[69].

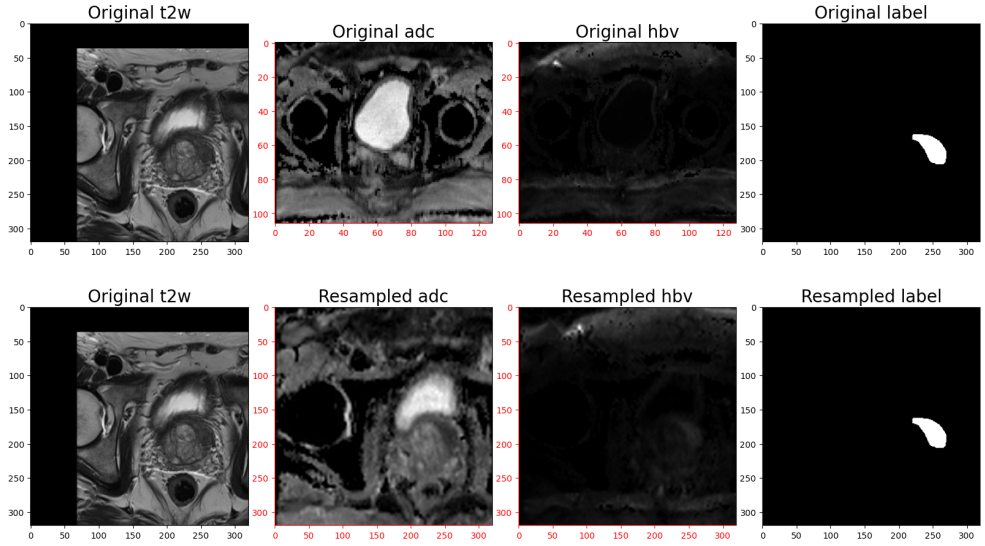


Figure 3.1: In the top row is displayed one example slice from unprocessed input scan. We can see that several things do not fit: t2w channel is shifted from the center, adc and hbv channel depict different slice as t2w, they have different zoom and also different resolution. In the bottom row is depicted the same slice, but after resampling. Specifically, after executing function in Code listing 3.1: t2w channel served as a template and therefore remained unchanged, but we can see, that adc and hbv now visually fit t2w and, in addition, also label fit darker area on adc as well as lighter area on hbv. Label remained also unchanged, because it already had the same resolution as t2w.

Center crop/pad

Next step is to adjust spatial resolution across the scans with center crop or pad to our predefined size (32, 256, 256). Resampling and center crop should ensure

that all samples in the dataset will depict at least similar region of the prostate with the same amount of details.

High intensities clipping

This step is done to remove very high intensities, which could be denoted as outliers. In general, outliers could cause very high gradients during back-propagation and could distract neural network training.

Zero-mean normalization

Next issue is different exposure among the scans - different range of voxel intensities. Some samples are for example overexposed or underexposed, which is caused by large variety of input sources. It is good for neural to have input samples with similar range of values, so the network weights can better specialize during back-propagation. Therefore we are doing normalization individually to each channel according to the formula

$$X_{out} = \frac{X_{in} - \mu}{\sigma} \tag{3.1}$$

where μ is the mean and σ is the standard deviation. This is also known as Z-Score or zero-mean normalization and it transform input data in way, that its mean is zero and standard deviation is 1.

Data augmentations

Data augmentation is a technique to artificially introduce variations into the dataset via various operations on the original data such as rotating or warping. It is one of the regularization techniques to increase variability of the data and thus potentially also improve robustness of the model and decrease over-fitting. We used same data augmentation techniques as in preparatory project[31] listed in Table 3.1.

3.3 Evaluation

In order to compare performance of the models in several experiments we need to introduce quantitative evaluation methods, which are described in this section.

3.3.1 Training routine

Five fold cross-validation

For the training we are using five fold cross-validation method to train five separate models in each experiment. The idea is to split the dataset into five equivalent parts and each time use four parts as a training dataset and the remaining part use as a validation set to evaluate model on an unseen data (see Figure 3.2). At

	Augmentation values	
	Range	Probability
Scaling	[0.7 - 1.4]	0.2
Rotation	[−30° - 30°]	0.2
Gaussian Noise		0.1
Gaussian Blur		0.2
Brightness Multiplicative		0.15
Contrast Augmentation		0.15
Simulate Low Resolution	[0.5 - 1.0]	0.25
Inverted Gamma	[0.7 - 1.5]	0.1
Gamma	[0.7 - 1.5]	0.3
Mirror		0.25

Table 3.1: Data augmentations used in training process

the end we take the mean of all five models and this value is considered as a final rating of given experiment.

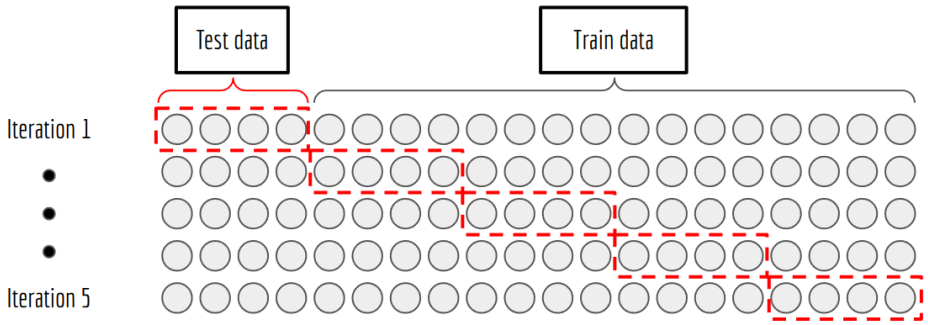


Figure 3.2: Example of five fold cross-validation partitioning. In most cases are samples in the dataset ordered according some characteristic, so the dataset is firstly shuffled or the samples are partitioned randomly, ensuring various types of data in each split.

Source: [31]

3.3.2 Quantitative evaluation

The main task of this work is to classify each pixel into two classes - tumor or not tumor. This is a binary classification problem, whose result verification can be summarized in confusion matrix with four categories[31]:

1. true positive (TP) - expected tumor and predicted tumor
2. false positive (FP) - doesn't expected tumor, but predicted tumor

3. true negative (TN) - doesn't expected tumor and doesn't predicted tumor
4. false negative (FN) - expected tumor, but doesn't predicted tumor

Precision and Recall

As described in [70] or [71], a naive or dummy approach for evaluation is using precision and recall, because these metrics are biased towards their objective and don't handle negative examples correctly.

Precision expresses the ratio between detected true positive and all detected positives - how many of the detected positives are actually positives. Its exact definition is

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

Precision is used, when we want our prediction to be as precise as possible in exchange for not detecting all objects.

Recall expresses the ratio between detected true positive and all positives that should have been detected - have many objects (true positive) were actually detected from total number of objects from given class i.e. how good is the model in detecting given class. It is defined as follows

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

Recall is used, when we want detect as much object from given class as possible in exchange for detecting unknown, but usually very high number of false positive.

In an ideal case are both, Precision and Recall equal 1:

- all detected objects have predicted correct class - Precision $= 1$
- all objects from all classes were detected - Recall $= 1$

Average precision

In our binary classification problem the model assign to every pixel/voxel a probability value of belonging to desired class. All voxels with probability higher than some pre-defined threshold can be classified as given class (in our case tumor) and all the others as not from class (background or unimportant). If the threshold is too low, more voxels are assigned as a tumor - the model will make more positive predictions, most likely also FP predictions, but the chance that some detection will be missed is lower. In term of precision and recall we can say that recall is high and precision is low. On the other hand with high threshold the model can miss some valid positives, but the chance that detected TP voxels are correctly labeled is higher - recall is low and precision is high. If we plot precision against recall along all possible thresholds we get so called precision-recall curve (see Figure 3.3). Based on this curve we can select threshold that fits our requirements the

most, but this is in general not so straightforward task, since we need to choose optimal trade-off between precision and recall[31].

One a solution to the issue of selecting most suitable threshold could be average precision (AP) score. AP summarize precision and recall values across all possible thresholds into one scalar value. It can be understood as the area under precision-recall curve and it is defined as follows:

$$AP = \int_{r=0}^{1} p(r)dr \tag{3.4}$$

From the Figure 3.3 we can see, that AP is high when recall as well as precision are high at the same time. It expresses how good is the model in detecting all positive examples, without the emphasis on correctly classifying negative examples - it is high when model can correctly handle positives.

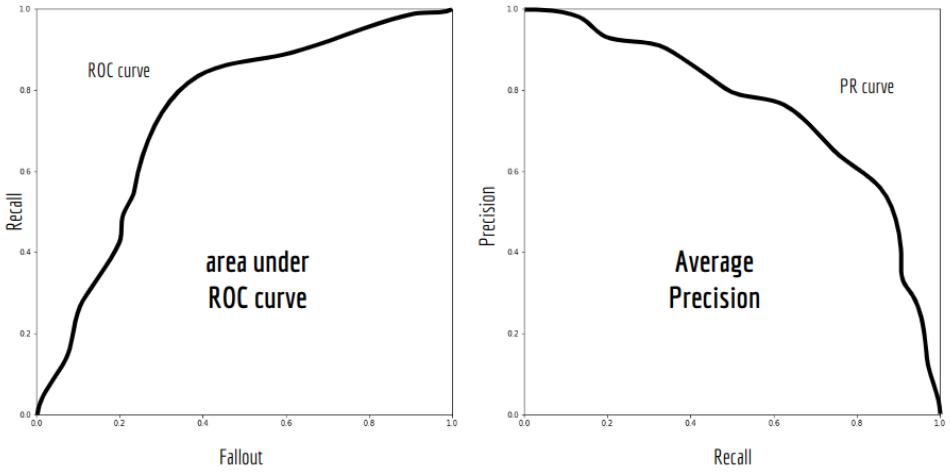


Figure 3.3: In the figure is an example of ROC curve and PR curve. In the ideal case we want AUROC equal 1 - across all probabilities that some negative example will be classified as positive (fallout), to detect all objects from given class (recall equal 1) i.e. even with zero fallout (this means no FP) we want to detect all objects.
And also we want AP to be equal 1 - across all probability levels of how many objects from given class were detected (recall) we want that all objects will be from given class (precision equal 1) i.e. even with recall equal 1 (this means all objects from given class were detected) we want all detections to belong to given class - no FP.

Source: [31]

Area under receiver operating characteristic curve

In a wast majority of research papers about result evaluation, for example [70] [71], is promoted the opinion that using only precision and recall can be insuf-

ficient and misleading. They suggest using also receiver operating characteristic curve, which expresses how the number of true positives (TP) varies with the number of false positives (FP). Ideally it is desired to increase only the number of TP, while the number of FP keep as low as possible. ROC curve is a graph showing dependency between the Fallout (Eq. 3.5) and Recall (Eq. 3.3) across all possible thresholds.

Fallout is the ratio between the number of negative samples wrongly categorized as positive and the total number of actual negative samples i.e. probability that negative sample will be classified as positive.

$$Fallout = \frac{FP}{FP + TN} \tag{3.5}$$

area under receiver operating characteristic curve (AUROC) is a metric expressing how good is the model in distinguishing between classes - how likely will model correctly classify given sample. Likewise AP, it represents the area under ROC curve across all possible classification thresholds in a single scalar value. Ideal model has AUROC equal 1, which means that it always make correct prediction. On the other hand model with AUROC equal 0 will always make incorrect prediction and model with AUROC 0.5 makes predictions that seems random.

In this work we are using exactly same evaluating approach as in preparatory project[31] to be able compare results with both metrics - average precision (AP) and area under receiver operating characteristic curve (AUROC). To get these scores we are still using `picai_eval`[65] package[2], which provides also merged score as a final rating.

3.4 Baseline model

This work is based on a preparatory project[31] for this master thesis, where we did comparison between CNN-based and ViT-based models for PCa detection and tumor segmentation. We opted also for our own implementation of ViT-based transformer with sliding window and U-Net like structure to have more flexibility in term of architecture customization. We wanted to utilize following modifications:

1. **Variable patch size** - we speculated that fixed patch size (2, 2, 2) is too small and we wanted to use larger sizes. Note that original ViT model for image processing[27] uses patches of size 16×16. Larger patch size means that more neighbouring pixels are converted into one embedding vector and therefore the output vector is more generalized.
2. **Variable window size** - in the patch embedding process with image size `[32, 256, 256]` and patch size for example (2, 4, 4) or (4, 4, 4), it can be

[2] see `https://github.com/DIAGNijmegen/picai_eval`

convenient to use SA window size (8, 8, 8) or smaller (4, 8, 8) to avoid any padding.

3. **Variable network depth** - larger patch size results in smaller image dimensions right in the beginning and we can take away one layer from the network saving some memory.
4. **Linear transformation in the up-sample blocks** - replace computationally intensive transposed convolutions in Patch Expanding blocks with classic `Linear` layers followed by reshaping of the outputs.

This solution outperformed CNN-based Unet[33] and ViT-based Swin UNETR[7] (see Table 3.2), whose implementations were acquired from MONAI[72] - open-source framework for medical imaging research[3].

Results comparison from [31]				
Model	Mean Best Score	Training time per epoch	#Parameters	Total mult-adds (G)
U-Net	0.618348	184sec	31 643 850	15.98
Swin UNETR	0.638168	685sec	15 704 732	189.87
Proposed model version1	0.659823	653sec	80 899 778	15.97

Table 3.2: Comparison of the results obtained by models tested in [31]
Source: [31]

3.4.1 Best model from the Preparatory project

All experiments in this work are based on our best model from the preparatory project[31], whose architecture is described in this section and can be seen in the Figure 3.4.

Patch Embedding Block

Given a 3D image with three channels as an input, the first step is to split this volume into 3D patches. For this step we used standard 3D convolution layer followed by `GELU` activation function to add some non-linearity and `Linear` layer for further processing (see Code listing 3.2).

The output of this block are embedding vectors - each patch is represented by 1D vector of length `channels[0]`. Note that we are not using positional encodings, because we always keep the same order of the patches.

[3] see `https://monai.io/`

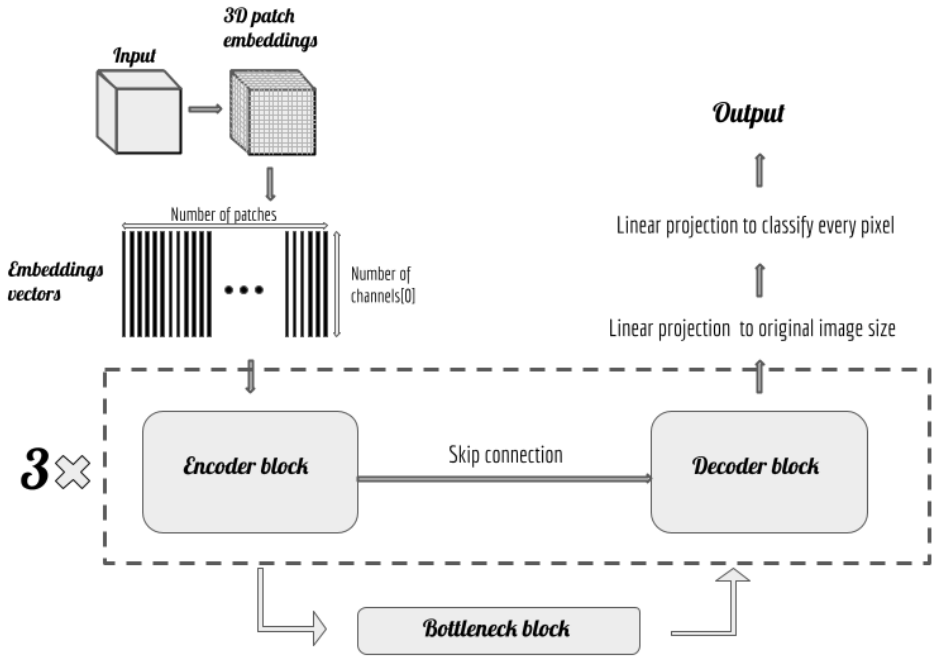


Figure 3.4: In the figure is outlined the architecture of the best model from preparatory project[31]

Source: [31]

Code listing 3.2: Patch Embedding Block code example

```
self.patch_embeddings = nn.Conv3d(in_channels=3,
                                  out_channels=channels[0],
                                  kernel_size=patch_size,
                                  stride=patch_size)
self.activation = nn.GELU()
self.linear = nn.Linear(in_features=channels[0], out_features=channels[0])
```

Encoder block series

Patch Embedding block is followed by three Encoder blocks. The overall architecture of the Encoder block and how input passes through with all skip connections can be seen in the Figure 3.5.

One Encoder block consists from window based self-attention block and shifted window based self-attention block (for closer description of these blocks see Section 2.3.3), each followed by MLP. MLP block consists from two linear layers with `GELU` activation in between and `Dropout` layer at the end. Number of output neurons is the same as an input size and number of hidden neurons is twice as much (see Code listing 3.3). Internal skip connections are implemented as a standard addition.

Code listing 3.3: MLP Block code example

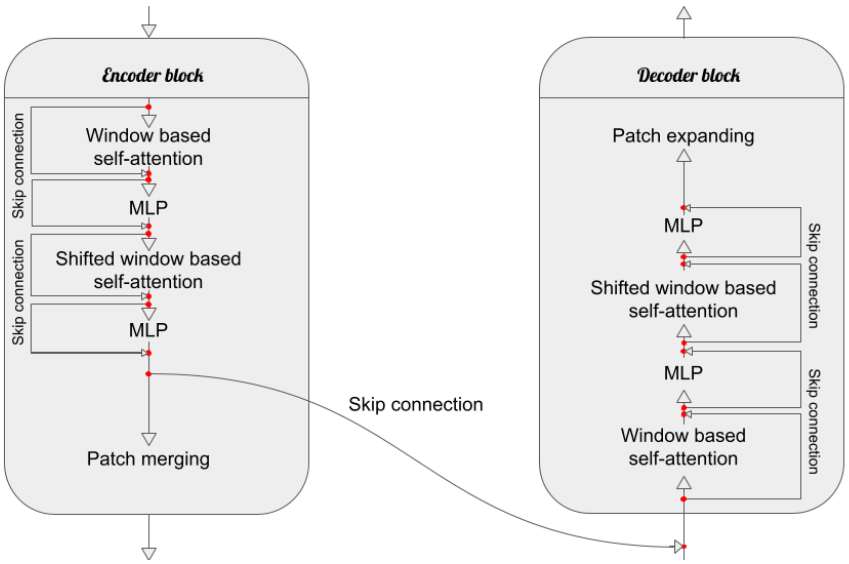


Figure 3.5: Proposed Encoder-Decoder architecture

```
self.linear1 = nn.Linear(in_features=channels, out_features=2*channels)
self.activation = nn.GELU()
self.linear2 = nn.Linear(in_features=2*channels, out_features=channels)
self.drop = nn.Dropout(0.05)
```

The last step of the Encoder is the Patch merging operation that downsample spatial resolution by joining adjacent non-overlapping 2×2 2D patches and doubles number of channels. (see Code listing 3.4).

Code listing 3.4: Patch Merging Block code example

```
self.patch_mergings = nn.Conv3d(in_channels=channels,
                                 out_channels=2*channels,
                                 kernel_size=(1, 2, 2),
                                 stride=(1, 2, 2))
self.activation = nn.GELU()
self.linear = nn.Linear(in_features=2*channels, out_features=2*channels)
```

Bottleneck block

Bottleneck has almost the same architecture as the Decoder block, except that there is no skip connection before input.

Decoder block series

In the overall architecture are also included three Decoders which have same self-attention/MLP scheme as encoder, but Patch merging operation is replaced with Patch expanding operation. It is the exact opposite that symmetrically up-sample

spatial resolution and reduce number of channels to half. We implemented it using `Linear` layer with 4× times bigger output size followed by reshaping.

The input to the Decoder is enriched with skip connection from the Encoder block on the same level in the network. This skip connection is implemented as concatenation along channel dimension and followed by `Linear` layer that halves the number of channels.

Note, that last Decoder block doesn't include Patch expanding operation to maintain appropriate output size. Complete insight on the input and output sizes during forward pass through our proposed model can be seen in table 3.3.

Proposed architecture version1
Image size per layer

Layer	Input size `[D, H, W, C]`	Output size `[D, H, W, C]`	Number of Parameters
Patch embedding	[32, 256, 256, 3]	[16, 64, 64, 192]	55 680
Encoder1	[16, 64, 64, 192]	[16, 32, 32, 384]	1 037 184
Encoder2	[16, 32, 32, 384]	[16, 16, 16, 768]	4 138 752
Encoder3	[16, 16, 16, 768]	[16, 8, 8, 1536]	16 535 040
Bottleneck	[16, 8, 8, 1536]	[16, 16, 16, 768]	42 504 192
Decoder1	[16, 16, 16, 768]	[16, 32, 32, 384]	11 815 680
Decoder2	[16, 32, 32, 384]	[16, 64, 64, 192]	2 958 720
Decoder3	[16, 64, 64, 192]	[16, 64, 64, 192]	667 968
Patch Transpose	[16, 64, 64, 192]	[32, 256, 256, 192]	1 186 176
Classification	[32, 256, 256, 192]	[32, 256, 256, 2]	386
		Total:	80 899 778

Table 3.3: Image size and number of parameters per layer

Source: [31]

Patch Transpose

After last decoder block is the shape of the partial output same as shape of the partial output after Patch embedding block. It is needed to up-sample it to the original size (32, 256, 256). We are doing so in the block by linear projection and reshaping.

Output classification

Partial output from Patch transpose block has the same spatial resolution as an original scan, but 192 channels need to be reduced to the number of output channels.

Chapter 4

Experiments and Results

The core of this work is to perform experiments with our ViT model to test proposed hypothesis and more advanced machine learning approaches like image registration or multi objective optimization. Our limitation is 24GB of graphical memory, so all hyper-parameters were chosen with respect to it. If doesn't stated otherwise, all experiments were performed with hyper-parameters listed in Table 4.1.

Training hyper-parameters	
Number of input channels	3
Number of output channels	2
Spatial dimensions :	[32, 256, 256]
Loss function:	Binary Cross Entropy
Optimizer:	AdamW
Learning rate:	0.00005
Weight decay	0.0001
Number of epochs:	70
Batch size:	1

Table 4.1: Hyper-parameters used for training process

4.1 Experiment 1 - More robust architecture

4.1.1 Description

In the first experiment we wanted to do some minor changes to the architecture of the model emerging from our best model (see Section 3.4.1) and preparatory project[31] future work:

1. Firstly we decreased learning rate from 0.0001 to 0.00005 for smaller learning steps. This change don't have significant effect as we are using `AdamW` optimizer with adaptive momentum.
2. Increase number of epoch from 60 to 70.
3. Shrink patch size from (2, 4, 4) to (1, 4, 4). As the first dimension of the input shape (32, 256, 256) is compared to others considerably smaller, we wanted to preserve the number of these slices and the amount of 3D information after Patch embedding process. The output shape after Patch embedding has therefore changed from (16, 64, 64) to (32, 64, 64).
4. Expand Patch merging and Patch expand kernel size from (1, 2, 2) to (2, 2, 2) to shrink and recover input features across all three dimensions.
5. Add more self-attention (SA) blocks in each Encoder/Bottleneck/Decoder, because SA is the main mechanism of Transformers for evaluating importance of the voxels for the final classification (see Section 2.3.1). An overview of one extended Encoder-Decoder block used in Experiment 1 can be seen in Figure 4.1. As can be noticed, there is no need to shift windows in the

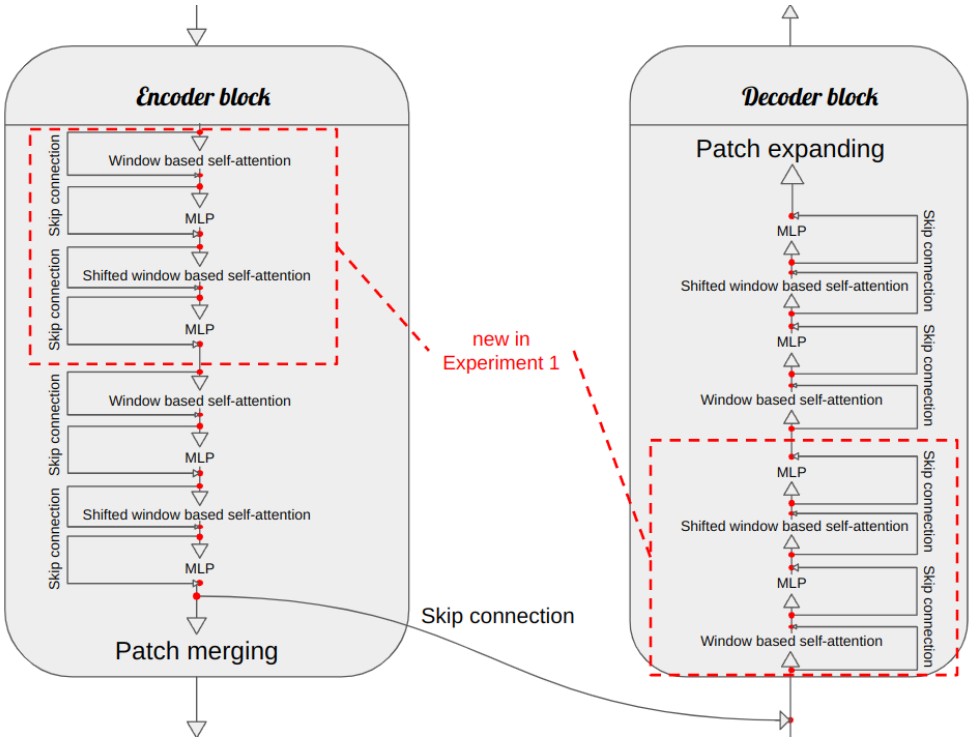


Figure 4.1: In the figure is outlined one Encoder-Decoder block with extended self-attention scheme used in Experiment 1.

Bottleneck block, because with (2, 2, 2) Patch merging kernel size is input already of size (4, 8, 8) and it can be comprehensively processed in one 3D window without any shift. However, we are still using four self-attention blocks to maximize computational power on the low resolution input with

dense features. For better overview of the Bottleneck architecture see Figure 4.2.

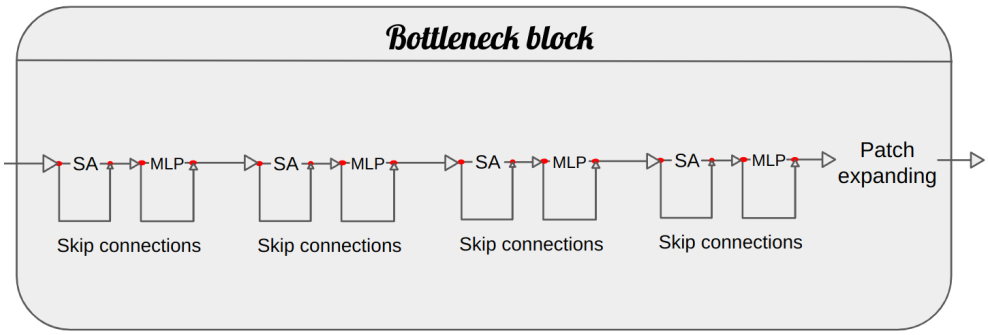


Figure 4.2: In the figure is the new architecture of the Bottleneck, where shifted window based self-attention (SW-SA) block was replaced with classic self-attention (SA) block and two new SA blocks were added. Since the input has the same resolution as the window (4, 8, 8), there is no need for "windowed" approach in the Bottleneck.

6. Reduce number of channels from [192, 384, 768, 1536, 768, 384, 192] to [120, 240, 480, 960, 480, 240, 120] in consecutive layers to compensate for deeper architecture since our computational resources are limited.
7. Added more intense data augmentation for better generalizability of the model (see Section 3.2).

4.1.2 Results

Fold number	Best Score		Best AP		Best AUROC	
	Baseline	Exp 1	Baseline	Exp 1	Baseline	Exp 1
1	0.652	0.690	0.461	0.503	0.844	0.877
2	0.741	0.725	0.596	0.565	0.887	0.885
3	0.626	0.719	0.410	0.529	0.841	0.909
4	0.615	0.622	0.394	0.403	0.836	0.842
5	0.665	0.693	0.479	0.497	0.851	0.890
mean	0.660	0.690	0.468	0.499	0.852	0.881

	Baseline	Experiment 1
Training time per epoch (aprox.):	653sec	814sec
Total number of parameters:	80 899 778	60 635 426
Total mult-adds (G):	15.97	8.02

Table 4.2: Experiment 1 results

4.1.3 Discussion

Parameters of the model could be in the context of Vision Transformers understood as some variables which store learned relationships between inputs. Each self-attention (SA) block has predefined number of these parameters - higher layers in the model have lower number of parameters, because the input has big spacial resolution i.e. there is more voxels to store information into. Vice versa input in lower layers has small spatial resolution so we need to encode necessary information into higher number of channels.

In this experiment we lowered number of SA channels by $37,5\%$, what naturally lowered number of parameters in MLP and SA layers. This change resulted in significant decrease of overall memory demands of the model and we were able to use twice as much self-attention layers and take an advantage from more robust ViT architecture that resulted in better performance.

Experiment summary

+ Higher AP and AUROC
+ Reduced total number of parameters
+ Lower total number of mult-add operations
− Longer training time

4.2 Experiment 2 - Wider 2D self-attention windows

4.2.1 Description

In the Experiment 1 we were using self-attention windows with size (4, 8, 8) to include 3D spatial information into SA computation. It means that these windows take into consideration 4 stacked areas from 8×8 2D region. We hypothesize, that using larger windows across main 2D dimension - (1, 16, 16) - in combination with 3D (4, 8, 8) windows to include spatial information may help to acquire broader relationships between patches and allow model to make more robust predictions.

In this experiment we reuse model and all hyper-parameters from Experiment 1, but we change window size in newly added Encoder/Decoder SA blocks to (1, 16, 16) (see Figure 4.3). As described above, there is no need to change window size in the Bottleneck and we are using the same one as in Experiment 1, depicted in the Figure 4.2.

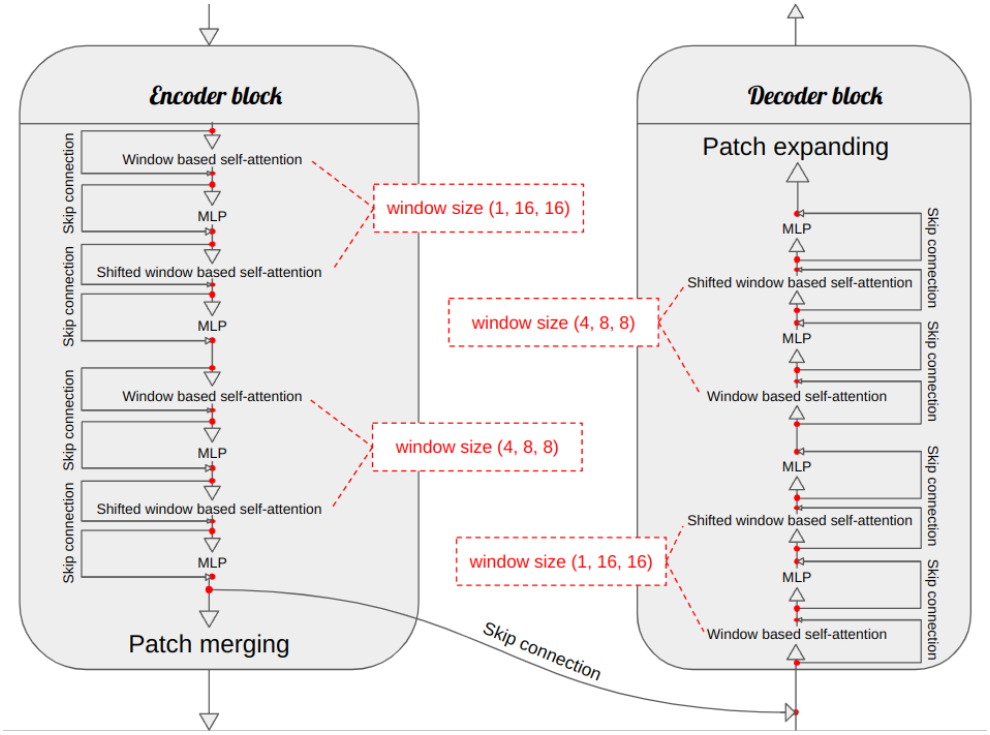


Figure 4.3: In the figure is showed an inclusion of the wider 2D self-attention windows in the Experiment 2.

4.2.2 Results

Fold number	Best Score		Best AP		Best AUROC	
	Exp 1	Exp 2	Exp 1	Exp 2	Exp 1	Exp 2
1	0.690	0.698	0.503	0.514	0.877	0.882
2	0.725	0.749	0.565	0.589	0.885	0.909
3	0.719	0.662	0.529	0.454	0.909	0.869
4	0.622	0.592	0.403	0.360	0.842	0.825
5	0.693	0.667	0.497	0.476	0.890	0.858
mean	0.690	0.674	0.499	0.479	0.881	0.867

	Experiment 1	Experiment 2
Training time per epoch (aprox.):	814sec	814sec
Total number of parameters:	60 635 426	60 635 426
Total mult-adds (G):	8.02	8.02

Table 4.3: Experiment 2 results

4.2.3 Discussion

In this experiment we wanted to include more voxels into self-attention (SA) computation along main 2D dimension. We transformed SA windows in specific blocks from 3D (4, 8, 8) windows to 2D (1, 16, 16) windows with same computational cost

$$4 \times 8 \times 8 \sim 1 \times 16 \times 16 \tag{4.1}$$

The main idea was to include more distant patches into self-attention (SA) computation in exchange for the potential loss of 3D information.

Experiment showed, that relationships between more distant patches are not as much important as 3D spatial information and we can therefore further hypothesise that the information necessary for the final inference is encoded in the input structures locally. Due to lower AP and lower AUROC we will base further experiments on the model from Experiment 1.

Experiment summary

+ The same training time, number of parameters and number of mult-add operations
− Lower AP and AUROC

4.3 Experiment 3 - Artificially annotated data

4.3.1 Description

As described in Section 3.1, we have 1294 out of 1500 images annotated by human experts which we have been using in our experiments so far. However, we have tumor segmentation labels generated by artificial intelligence for every scan and in this simple experiment we want to research the effect of extending our dataset by artificially labeled input scans.

It should be generally better for the machine learning model to have greater variability of the input data, because it allows to adapt and specialize for larger number of more specific examples. Although these images account only for about $\approx$ 14% and the accuracy of their labels is not verified, we hope that addition of new data will be beneficial for more accurate predictions. We will use newly extended dataset with training setup from Experiment 1, because it achieved best overall score.

4.3.2 Results

Fold number	Best Score		Best AP		Best AUROC	
	Exp 1	Exp 3	Exp 1	Exp 3	Exp 1	Exp 3
1	0.690	0.637	0.503	0.483	0.877	0.792
2	0.725	0.711	0.565	0.584	0.885	0.838
3	0.719	0.682	0.529	0.553	0.909	0.811
4	0.622	0.561	0.403	0.336	0.842	0.786
5	0.693	0.657	0.497	0.503	0.890	0.813
mean	0.690	0.650	0.499	0.492	0.881	0.808

	Experiment 1	Experiment 3
Training time per epoch (aprox.):	814sec	950sec
Total number of parameters:	60 635 426	60 635 426
Total mult-adds (G):	8.02	8.02

Table 4.4: Experiment 3 results

4.3.3 Discussion

This experiment showed, that artificially labeled data doesn't help improve performance of the model. From the results in Table 4.4 we can see that this model achieved comparable average precision (AP), but significantly lower area under receiver operating characteristic curve (AUROC). This means that model wasn't so good in distinguishing between classes and had lower confidence in detecting true negative voxels that typically results in higher number of false positive.

Experiment summary

- Lower AP and AUROC
- Longer training time

4.4 Experiment 4 - Image registration

4.4.1 Description

In the Figure 4.4 is showed one slice example of raw data scan. We can see that channels are not aligned in 3D space, they depict different region and have different resolution. On the top of that, segmentation label doesn't correspond with adc and hbv channels. In this experiment we look at three image registration types - rigid, affine and b-spline - to fix aforementioned issues and compare their effect on the final score of the model. Image registration is not a new topic and as described in Section 2.4, there has been extensive research going on especially in

medical image processing. We want to make the most out of this technique and include it in our model for data pre-processing.

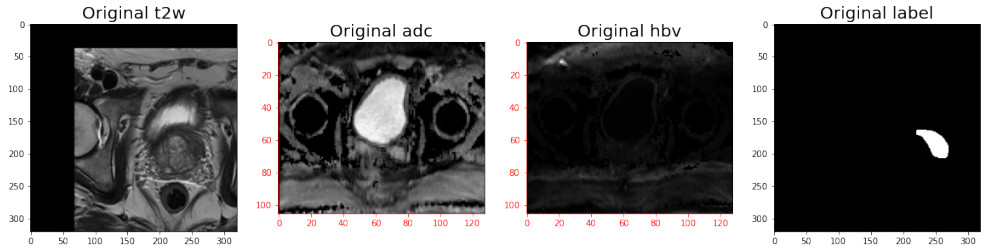


Figure 4.4: In the figure is showed an unprocessed example of provided scans.

We use our best model from Experiment 1 and do three full training cycles with five fold cross-validation. Each training cycle consists from all methods mentioned in section 3.2, but resampling is substituted by a specific registration type.

To register our prostate MRI scans we use `SimpleElastix`[42] multi-lingual library, an extension of `SimpleITK`[66][67][68], designed to simplify access to C++ `Elastix`[36][73] toolbox for rigid and nonrigid image registration.

Raw data rigid registration

Firstly, we want to know how does raw input scan look like after simple rigid registration. We used function in Code listing 4.1 with default rigid parameter map, but with lower number of resolutions as default value 4. Resolutions in this case doesn't characterize spatial resolution, but epochs in multi-resolution registration strategy. The overview and explanation of multi-resolution registration strategy is provided in detail in [74], but generally in each resolution is image downsampled, smoothed or both, which reduce complexity and considerably saves time.

Code listing 4.1: Rigid registration function

```
def register_img(fixed, moving):
    elastixImageFilter = sitk.ElastixImageFilter()
    elastixImageFilter.LogToConsoleOff()
    elastixImageFilter.SetFixedImage(fixed)
    elastixImageFilter.SetMovingImage(moving)

    parameterMap = sitk.GetDefaultParameterMap('rigid')
    elastixImageFilter.SetParameterMap(parameterMap)
    elastixImageFilter.SetParameter('NumberOfResolutions', '3')

    elastixImageFilter.Execute()
    resultImage = elastixImageFilter.GetResultImage()
    return resultImage
```

In the Elastix manual[1] is recommended as a good starting point number of resolutions equal 3. Using higher numbers like 5 or 6 is suitable for images with high spatial resolution with details further away, as they are more blurred and

[1] see `https://usermanual.wiki/Document/elastix490manual.1389615963/help`

more attention could be paid to main guiding shapes. In our case, using default number of resulutions 4 or higher led to significant displacement of moving image and therefore we had to change it to 3. For a visual example of registered image see Figure 4.5.

Raw data affine registration

The next step after rigid registration is little more advanced affine registration, which includes also shearing and scaling along one axis. Algorithm with default affine parameters and number of resolutions 3, however, didn't performed as expected. Adc channel looked at the first sight good, but hbv channel was overexposed and dislocated. We had to further reduce number of resolutions to 2. The best result is displayed in Figure 4.5.

Raw data b-spline registration

The most advance registration type researched in this work is the b-spline. As described in Section 2.4, it has tendency to quite disturb information about internal structures in the scan and therefore we have to choose parameters more carefully. We use number of resolution 3, but with very low number of steps 10, in comparison to the default value 256 (see Code listing 4.2). The example scan after b-spline registration is depicted in Figure 4.5 together with all other types.

Code listing 4.2: Paramter setting for b-spline registration

```
parameterMap = sitk.GetDefaultParameterMap('bspline')
elastixImageFilter.SetParameterMap(parameterMap)
elastixImageFilter.SetParameter('NumberOfResolutions', '3')
elastixImageFilter.SetParameter('GridSpacingSchedule', ['1.9','1.41', '1'])
elastixImageFilter.SetParameter('MaximumNumberOfIterations', '10')
```

Experiment process:

1. Convert raw `.mha` scans to `.nii.gz` format and save
2. Iterate over `.nii.gz` scans and:
    a. Unify spacing of all scans to (3.0, 0.5, 0.5) with appropriate change of resolution
    b. Register adc and hbv channels of each scan to its t2w channel with rigid (see Code listing 4.1), affine and b-spline transformation
    c. Resample label to the dimension of t2w channel
    d. Center crop/pad each new scan and also label
    e. Save rigid, affine and b-spline samples with label and their three channels - original t2w and registered adc and hbv
3. Continue as in Experiment 1

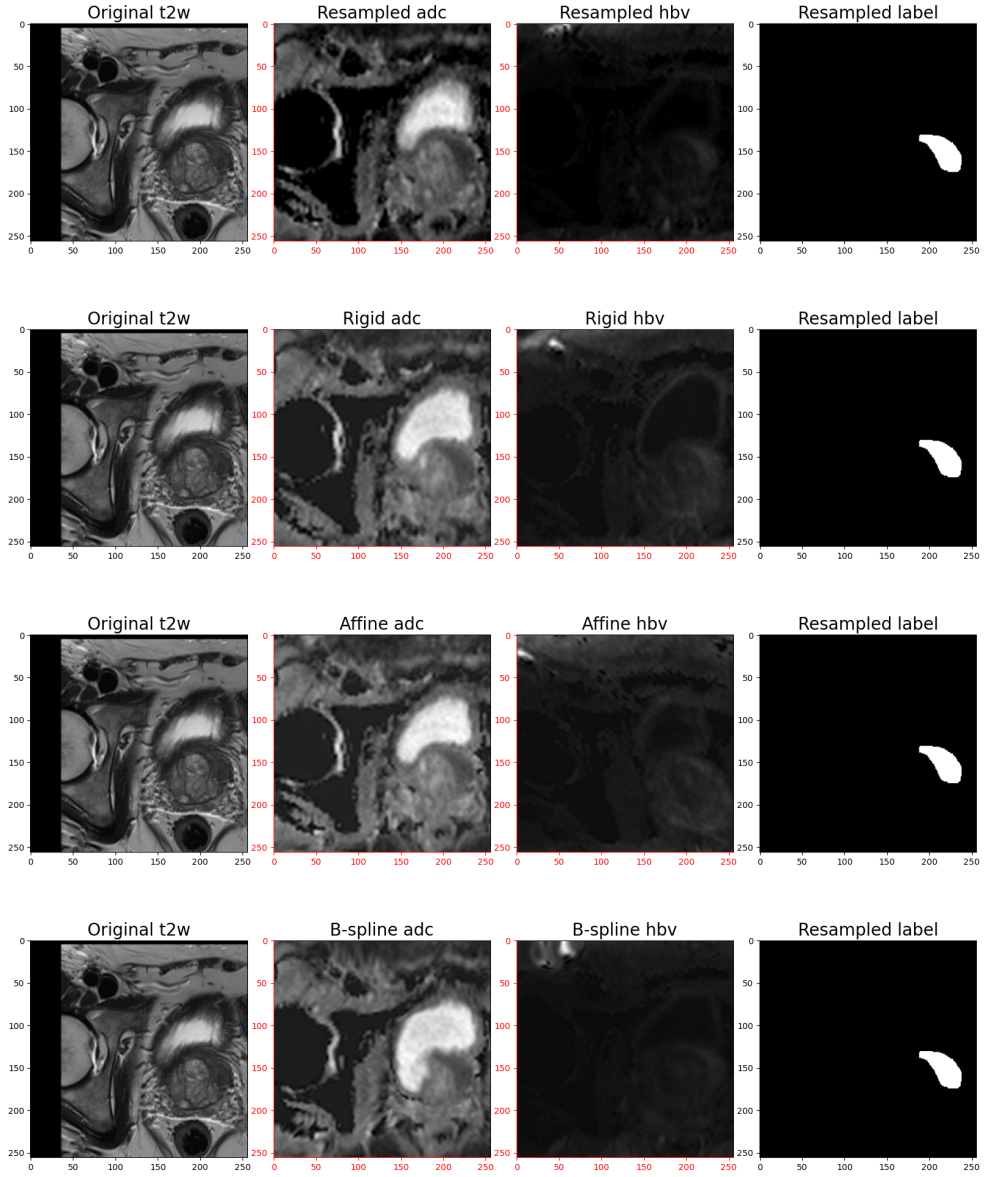


Figure 4.5: In the figure is presented a comparison between resampling method from Section 3.2 and all researched registration types after center crop/pad operation to predefined size (32, 256, 256). At the first sight all methods produced very similar results with slightly different intensity adjustments and feature delineations. We can for example see that hbv channel after affine registration is little bit stretched and the white area indicating tumor is not as clearly outlined as in rigid type, or for example b-spline registration produced some expected warping of the structures.

4.4.2 Results

Rigid registration

Fold number	Best Score		Best AP		Best AUROC	
	Exp 1	Rigid	Exp 1	Rigid	Exp 1	Rigid
1	0.690	0.619	0.503	0.394	0.877	0.845
2	0.725	0.713	0.565	0.552	0.885	0.874
3	0.719	0.630	0.529	0.426	0.909	0.834
4	0.622	0.551	0.403	0.264	0.842	0.838
5	0.693	0.641	0.497	0.431	0.890	0.851
mean	0.690	0.631	0.499	0.413	0.881	0.848

	Experiment 1	Experiment 4 - Rigid
Training time per epoch (aprox.):	814sec	814sec
Total number of parameters:	60 635 426	60 635 426
Total mult-adds (G):	8.02	8.02

Table 4.5: Experiment 4 - rigid registration results

Affine registration

Fold number	Best Score		Best AP		Best AUROC	
	Exp 1	Affine	Exp 1	Affine	Exp 1	Affine
1	0.690	0.634	0.503	0.418	0.877	0.849
2	0.725	0.708	0.565	0.549	0.885	0.868
3	0.719	0.677	0.529	0.461	0.909	0.894
4	0.622	0.545	0.403	0.306	0.842	0.783
5	0.693	0.635	0.497	0.396	0.890	0.875
mean	0.690	0.640	0.499	0.426	0.881	0.854

	Experiment 1	Experiment 4 - Affine
Training time per epoch (aprox.):	814sec	814sec
Total number of parameters:	60 635 426	60 635 426
Total mult-adds (G):	8.02	8.02

Table 4.6: Experiment 4 - affine registration results

B-spline registration

Fold number	Best Score		Best AP		Best AUROC	
	Exp 1	B-spline	Exp 1	B-spline	Exp 1	B-spline
1	0.690	0.697	0.503	0.520	0.877	0.874
2	0.725	0.719	0.565	0.552	0.885	0.886
3	0.719	0.708	0.529	0.529	0.909	0.887
4	0.622	0.612	0.403	0.397	0.842	0.827
5	0.693	0.719	0.497	0.544	0.890	0.895
mean	0.690	0.691	0.499	0.508	0.881	0.874

	Experiment 1	Experiment 4 - B-spline
Training time per epoch (aprox.):	814sec	814sec
Total number of parameters:	60 635 426	60 635 426
Total mult-adds (G):	8.02	8.02

Table 4.7: Experiment 4 - b-spline registration results

4.4.3 Discussion

In this experiment we compared three main image registration types - rigid, affine and b-spline - on prostate MRI scans for Vision Transformer prediction capability. Our hypothesis was that registration of three channels of each scans together with label can help in the training process to determine more exact relationships and improve accuracy of the predictions. We expected rigid or affine registration to produce best results, as the scans seem to best fit by human eye assessment (see Figure 4.5).

Fold number	Exp 1	Rigid	Affine	B-spline
1	0.690	0.619	0.634	0.697
2	0.725	0.713	0.708	0.719
3	0.719	0.630	0.677	0.708
4	0.622	0.551	0.545	0.612
5	0.693	0.641	0.635	0.719
mean	0.690	0.631	0.640	0.691

Table 4.8: Experiment 4 - Result comparison

Results showed that these two types didn't performed as good as baseline model with b-spline resampling and achieved significantly lower overall score. B-spline registration surprisingly achieved best overall score, even though the detail

in these scans can seem visually violated and skewed. Table 4.8 provide brief outlook on the overall score from all registration types and compare them with the model from Experiment 1.

Experiment summary

+ Research of which registration type has the highest benefit for our task
− Lower or identical overall score

4.5 Experiment 5 - Multi-Task learning model

4.5.1 Description

In addition to the tumor segmentation labels, we have also available prostate delineation labels generated by AI that we want to utilize in this experiment. Although their accuracy is not verified by human experts, we decided to conduct experiment with multi-task learning (MTL) optimization and fully exploit this technique. Our objective is to extend the model by second Decoder branch for prostate delineation prediction.

As pointed out in [46], parallel task optimization process and inter-task knowledge sharing is a promising solution for recognizing more complex objects in real world scenarios and therefore we hope that simultaneous learning to predict tumor segmentations as well as prostate delineations will contribute to better generalizability and the model will improve overall score.

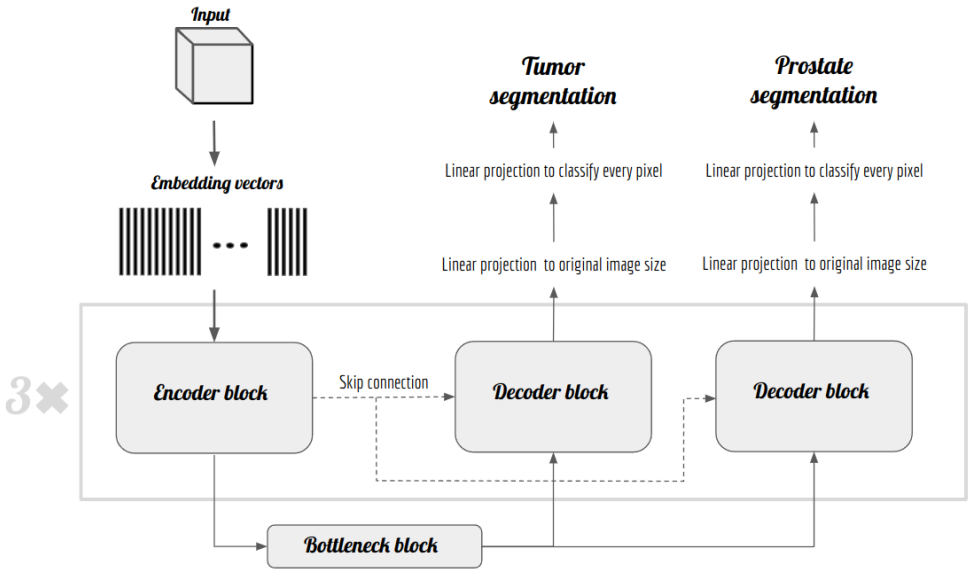


Figure 4.6: In the figure is showed the architecture with two decoder branches used in Experiment 5.

In this experiment we added second decoder branch as showed in the Figure 4.6 that is designed to predict prostate region segmentation. The network has therefore two objectives that share encoder branch and the Bottleneck, but each has own decoder branch specialized for specific task.

Each encoder/decoder block has the same architecture as in Experiment 1, but we had to reduce the complexity by reducing the number of channels and the number of self-attention heads, because the second branch adds new memory demands to our limited processing space:

1. Reduce number of channels from [120, 240, 480, 960, 480, 240, 120] to [81, 162, 324, 648, (324, 324), (162, 162), (81, 81)] in consecutive layers.
2. Reduce number of self-attention heads from [4, 8, 16, 32, 16, 8, 4] to [3, 6, 12, 24, (12, 12), (6, 6), (3, 3)] in consecutive layers.
3. Two previous points results in reduction of channels per attention head from 30 to 27

As long as we have two objectives, we must have also two loss functions defining individual deviations from the target. We use partial loss objectives to calculate final loss as shown in Code listing 4.3.

Code listing 4.3: Calculating final loss objective from two partial ones.

```
x_tumor, x_prostate = model(inputs)

seg_label = torch.unsqueeze(labels[:,0,:,:,:], 1)
deli_label = torch.unsqueeze(labels[:,1,:,:,:], 1)

loss1 = loss_func(x_tumor, seg_label)
loss2 = loss_func(x_prostate, deli_label)
loss = loss1+loss2

optimizer.zero_grad()
loss.backward()
optimizer.step()
```

For the validation and final score calculation we use only tumor segmentation outputs, as it is our main objective.

4.5.2 Results

Fold number	Best Score		Best AP		Best AUROC	
	Exp 1	Exp 5	Exp 1	Exp 5	Exp 1	Exp 5
1	0.690	0.705	0.503	0.541	0.877	0.869
2	0.725	0.749	0.565	0.590	0.885	0.908
3	0.719	0.753	0.529	0.584	0.909	0.923
4	0.622	0.632	0.403	0.404	0.842	0.860
5	0.693	0.723	0.497	0.540	0.890	0.909
mean	0.690	0.712	0.499	0.532	0.881	0.894

	Experiment 1	Experiment 5
Training time per epoch (aprox.):	814sec	990sec
Total number of parameters:	60 635 426	33 051 499
Total mult-adds (G):	8.02	4.33

Table 4.9: Experiment 5 results

4.5.3 Discussion

This experiment exceeded our expectations and we created model with best overall score with considerably lower number of parameters. We used prostate delineations to act as an effective inductive bias for the tumor segmentation task that influenced whole encoder branch for better prediction performance.

It could be also interesting to evaluate the accuracy of prostate delineation segmentation in the second branch, but since we were limited in time and it was not the goal of this work, we did not measure this objective.

Experiment summary

+ Better overall score, AP and AUROC
+ Lower number of parameters
− Longer training time

4.6 Experiment 6 - Multi-class output

4.6.1 Description

Based on the improvement in the Experiment 5 (see Table 4.9) we designed brand new experiment for comparison with with multi-class output. We based it on an assumption that better overall score in the Experiment 5 was achieved thanks to the prostate delineation labels assigned to every scan and therefore allowed better weights specialization of the encoder branch. On the contrary, not every scan

contains segmented tumor, which makes its tumor segmentation label empty - made out of zeros - and we speculate that these empty labels could have disturbed model's weights in the previous Experiments 1-4, where prostate delineations were not used.

In this experiment we used model and all configurations from the Experiment 1, but we have created new labels by adding prostate delineations and tumor segmentations (see Figure 4.7). These new labels contains three possible classes which are during optimization one-hot encoded:

- **0** - Background
- **1** - Prostate delineation
- **2** - Tumor Segmentation

Therefore we had to modify also the number of output channels to 3, so the model is capable to predict probability for each class.

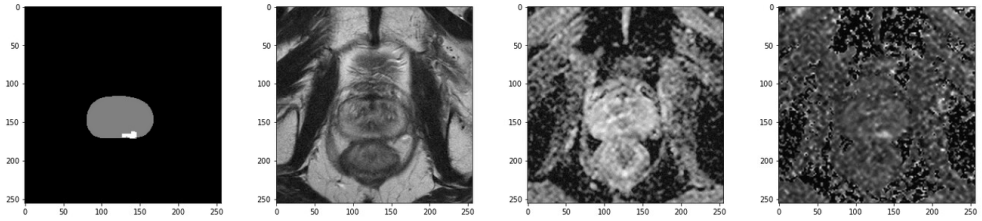


Figure 4.7: In the figure is an example of the scan and the label with prostate delineation and tumor segmentation.

For the training and optimization we use all three classes, but in order to be able to compare model with previous experiments, we do validation only on tumor segmentation. We hypothesize that prostate delineations included with every scan will help to bias weights and the model will achieve better score.

4.6.2 Results

Fold number	Best Score		Best AP		Best AUROC	
	Exp 5	Exp 6	Exp 5	Exp 6	Exp 5	Exp 6
1	0.705	0.688	0.541	0.498	0.869	0.879
2	0.749	0.753	0.590	0.585	0.908	0.921
3	0.753	0.693	0.584	0.500	0.923	0.886
4	0.632	0.603	0.404	0.360	0.860	0.848
5	0.723	0.648	0.540	0.426	0.909	0.869
mean	0.712	0.677	0.532	0.474	0.894	0.881

	Experiment 5	Experiment 6
Training time per epoch (aprox.):	990sec	822sec
Total number of parameters:	33 051 499	60,635,499
Total mult-adds (G):	4.33	8.02

Table 4.10: Experiment 6 results

4.6.3 Discussion

We hoped that inclusion of prostate delineation in every label would help to influence weights and the model will achieve significantly better overall score, but our hypothesis in this experiment wasn't confirmed. One possible cause could be that output with more output channels may need more training epochs for precise adjustment, but it is only our speculation.

Experiment summary

— All validation metrics are lower

Chapter 5

Discussion

5.1 General Discussion

The main objective of this master thesis was to develop a comprehensive machine learning model for prostate cancer detection on MRI scans that can serve as an effective workflow support for radiologists. We based our implementation on the not so long ago proposed approach called Vision Transformer (ViT)[5] with self-attention mechanism and we included in it more advanced concepts like shifted windows, skip connections or U-Net like structure. It achieved satisfying results as it outperformed state-of-the-art models implemented in MONAI framework and achieved comparable performace with PI-CAI challenge top competitors.

*If we compare our best model from the **Experiment 5** with ranking score 0.712 and top PI-CAI challenge competitors (see Section 2.6) with validation scores between 0.750-0.800 we draw a conclusion that together with future improvements our model can serve as a brilliant starting point for introducing AI into practical testing.*

However, as long as the goal of this work wasn't to join the competition and we also used different validation method as challenge competitors that had submitted their models directly for evaluation, our score can be slightly inaccurate when comparing to others.

For the training we used public dataset originally proposed for PI-CAI challenge which consist of 1500 prostate multi-parametric MRI scans. Each scan contain `t2w`, `adc` and `hbv` channels from axial view (top-down), which unfortunately do not show the same area. One channel can have for example different zoom as the other two and therefore it doesn't match the label. This can lead to a serious inaccuracy of the model and therefore channels of each scan needs to be synchronized with each other as well as with label. After visually inspecting several scans, we assumed that `t2w` was the main channel by which the labels were created and registered the remaining two channels and labels to match this channel. In some scans, however, the label could have been created according to other channel and by registering it to `t2w` we introduced error to the dataset. This source of error

can be avoided only by manual inspection of each scan and determining which channel match the label. This in many cases requires professional expert for assessment, as for common observer is hard to recognize what is tumor in the MRI scan.

Another labeling issue is that only 1294 out of aforementioned 1500 scans are annotated by radiologist experts. Remaining scans have provided label only created by artificial intelligence, which is not verified and can be inaccurate. We experimented using these AI annotated scans in the Experiment 3, but it doesn't produced better results.

Technical limitation of our work is 24GB of graphical memory, as ViT based architectures have very high memory demands and even this space is not enough. Extending this space would allow to use architecture with more self-attention blocks, more self-attention heads and more channels to encode inter-voxel relationships. One interesting experiment on better hardware would be to use larger 3D self-attention windows. We tried to use larger 2D windows in exchange for not including 3D information in the Experiment 2, but it turned out that 3D information is more important than larger 2D windows as the Experiment didn't yield better results.

Image registration

To align channels of each scan we researched three image registration types - rigid, affine and b-spline. We wanted to determine which type suits our domain the best. On the contrary to [75] where authors found no differences between rigid and elastic (b-spline could be understood as elastic) registration, our model trained on the b-spline registered scans produced significantly better results.

*However, image registration methods researched in **Experiment 4** did not provide better overall scores than preprocessing method with b-spline resampling implemented in PI-CAI baseline[65] package.*

In the work [43] are authors in favor for rigid registration as for non-rigid (b-spline). They justify their claim with higher computational cost of non-rigid registration and that non-rigid registration is hard to validate. We also claim that scans after rigid or affine registration looks more natural to human observer, because b-spline registered scans seems to have disturbed or damaged internal details, but Experiment 4 showed that machine learning model is able to produce best results specifically on b-spline registered scans.

One thing regarding image registration worth to mention is that with every image registration type was whole dataset modified. Train/validation partitioning remained the same, but separate scans look different and thus we don't use same looking validation sets.

Another interesting thing to notice is that model with b-spline image resampling (Experiment 1) and model with b-spline image registration (Experiment

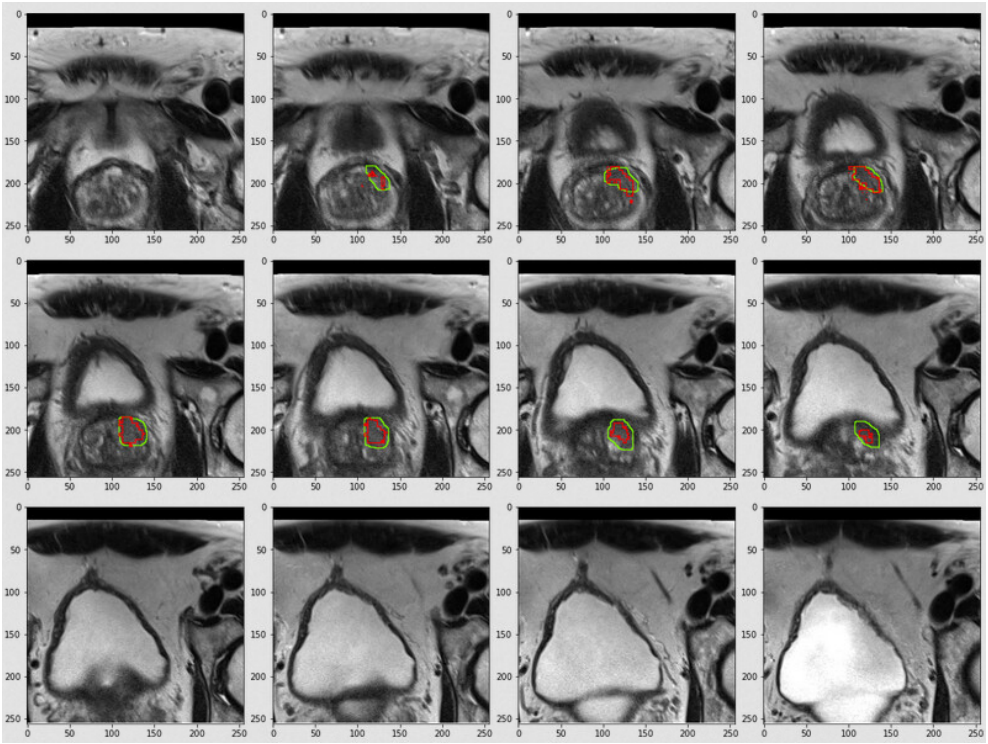


Figure 5.1: In the figure is a visual example of successful tumor detection with our best model from Experiment 5. Green is the tumor segmentation label and red is the prediction with threshold 0.25 .

4) produced almost identical scores, which can mean that these two operations have similar implementation.

Multi-task learning

Although we couldn't find any similar works with multi-task learning optimization and we were unsure how to design overall architecture, our first experiment with MTL model (Experiment 5) yielded surprisingly good results. In [76] is pointed out, that multi-domain models where tasks share earlier layers could achieve better performance than the ones sharing deeper layers and therefore we added second decoder branch for prostate delineation segmentation, sharing encoder branch and Bottleneck between both tasks.

*We can declare that simultaneous optimization of two objectives as implemented in the **Experiment 5**, more specifically tumor segmentation and prostate delineation segmentation that share lower parts of the network helps improve the prediction accuracy and model's overall performance.*

Better results even with significantly lower number of model parameters were achieved thanks to the second decoder branch and we hypothesize, that major

part plays the fact that every input sample contains label with prostate delineation which significantly helps to bias whole encoder branch in a positive way and allows better specialization of the model. On the contrary, tumor segmentation is not presented in every scan which results in totally blank label made only from zeros. As long as majority of labels ($\approx 4:1$) is blank - scan doesn't contain tumor - we speculate, that these samples could have negative effect on the overall training process and can confuse model's weights.

On the other hand, results of the Experiment 6 (see Table 4.10) aren't as good as expected. The main idea of this experiment was that new output channel i.e. prostate delineation segmentation, may help to affect weights of the whole model and help to locate tumor inside prostate zone more precisely. This assumption is not based on any previous work and it is only our hypothesis that needs further research and tuning that may be relevant for the future work.

5.2 Related work

5.2.1 PI-CAI challenge works

Works from PI-CAI challenge presented in Section 2.6 that are most related to our task were proposed in late November 2022 when this thesis was already in progress. They utilize various advanced techniques thanks to which they have achieved top-notch performance.

We particularly like the idea on which is based MNet[51] model used in first work[50], which alternates 2D convolutions along various axes and joins then with 3D convolutions. The amount of each convolution is further encoded using learnable parameters so that the model can automatically determine which plane in 3D space is the most important. We think that it could be interesting to experiment with this approach by alternating larger 2D self-attention windows along different axes and join them with 3D SA windows.

Another interesting idea is to do region-of-interest cropping instead of simple center crop. After the center crop can be prostate area shifted from the middle or it can be even partially cut out from the image making it harder for machine learning model to adjust. It is possible to train separate model using prostate delineations to predict prostate area and based on the bounding box of this area do region-of-interest cropping of the whole prostate.

Since all these models achieved better validation score we suggest to further research their characteristic features and try to use them in the future work.

5.2.2 Other research

Most of the ongoing research related to using ViTs for medical image segmentation is focused on the delineation of the whole organs or its parts[8][9][10][77]. This is into some extent easier task as a tumor segmentation, because whole organs are depicted in every sample and thus network doesn't have to evaluate their

presence. Tumors, on the other hand, may not show up on a scan i.e. tumor is more likely not presented in the scan that makes the whole training process and network specialization harder.

One of the works regarding tumor segmentation is [11], where Peiris et al. proposed robust volumetric transformer for accurate tumor segmentation but it is hard to compare to our work, because they made experiments on a brain tumor segmentation and in addition they used different evaluation metrics. Similarly in [7], Hatamizadeh et al. proposed their model for brain tumor segmentation using Dice score as an evaluation metrics, which is hard to compare.

In [78] authors compared Transformer based U-Net to CNN based U-Nets specifically on prostate tumor segmentation task. They unfortunately don't provide comprehensive quantitative results in the form of tables, but they mention that Transformer based model achieved the best Dice score among tested models.

Other than that, we weren't able to find any other ongoing research related to using Vision Transformers for prostate tumor segmentation which makes our work even more breakthrough in this area.

Chapter 6

Conclusion and Future work

6.1 Conclusion

In this work we studied Vision Transformers and their utilization for automatic prostate cancer detection on MRI scans. We conducted research whose aim was to introduce artificial intelligence into practice as an effective support for radiologist workflow. We designed our own ViT-based architecture with shifted windows approach and U-Net like structure that outperformed other implementations. We further examined various image registration types and their effect on the machine learning model's performance. In contrast with our speculation, b-spline registration provided best results, even though details in the image seems to be damaged when inspecting the image by human eye. In the last stage we were looking into multi-task learning technique where we simultaneously predicted tumor segmentations as well as prostate delineations. These two tasks shared lower parts of the network where prostate delineations acted as an inductive bias for tumor detection and even with half as much number of parameters model achieved the best overall score of 0.712, which is not small, but can be further improved.

In the Appendix A are provided example scans with tumors and it can be seen that predictions match the labels into large extent. However, as presented in section 2.6, it is possible to achieve even more superior performance and therefore we would like to recommend our work for the future research and improvements. Afterwards it could be considered for testing in practical use as a support for prostate cancer diagnosis which may contribute to higher standards of the patient healthcare.

6.2 Future work

The main objective for the future work is to develop robust and accurate model for prostate tumor segmentation that can be put into testing for practical use. This is very extensive task that demands a lot of trying and testing and we have following suggestions that may help to achieve better score:

1. **Use prostate delineations as a fourth input channel**: every scan contains segmented prostate delineation and we assume that using it as a fourth input channel during training may help to indicate more accurate tumor location.
2. **Use Focal loss**: Focal loss focuses training on hardly classifiable examples (parameter gamma) and also on poorly represented classes (parameter alpha) which may be beneficial for our problem, since tumor pixels are poorly represented in comparison to background pixels.
3. **Overlapping patches**: SegFormer[79] model uses so called Overlapped Patch Embedding process and Overlapped Patch Merging process and the authors claim, that using overlapping helps to incorporate an inter-patch information and preserve continuity between patches.
4. **Region-of-interest crop**: Center crop may depict prostate shifted from the middle of the image and therefore it isn't located in the same place of the scans. This has potential to make the learning process more difficult and therefore we suggest to use region-of-interest crop. Use attached prostate delineations to train separate model, predict prostate region and crop the scan around this region.
5. **Self-supervised pre-training**: As presented in [50] and more specifically in [53], self-supervised pre-training helps to specialize weights already before training and model can achieve significantly better performance.

Bibliography

[1] A. Simsir, E. Kismali, R. Mammadov, G. Gunaydin and C. Cal, 'Is it possible to predict sepsis, the most serious complication in prostate biopsy?' en, *Urol. Int.*, vol. 84, no. 4, pp. 395–399, Mar. 2010.

[2] M. Shahait, J. Degheili, F. El-Merhi, H. Tamim and R. Nasr, 'Incidence of sepsis following transrectal ultrasound guided prostate biopsy at a tertiary-care medical center in lebanon,' en, *Int. Braz J Urol*, vol. 42, no. 1, pp. 60–68, Jan. 2016.

[3] M. J. P. van Osch and A. G. Webb, 'Safety of Ultra-High field MRI: What are the specific risks?' *Current Radiology Reports*, vol. 2, no. 8, p. 61, Jul. 2014.

[4] D. Bulas and A. Egloff, 'Benefits and risks of MRI in pregnancy,' en, *Semin Perinatol*, vol. 37, no. 5, pp. 301–304, Oct. 2013.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit and N. Houlsby, 'An image is worth 16x16 words: Transformers for image recognition at scale,' *CoRR*, vol. abs/2010.11929, 2020. arXiv: 2010.11929. [Online]. Available: https://arxiv.org/abs/2010.11929.

[6] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian and M. Wang, *Swin-unet: Unet-like pure transformer for medical image segmentation*, 2021. DOI: 10.48550/ARXIV.2105.05537. [Online]. Available: https://arxiv.org/abs/2105.05537.

[7] A. Hatamizadeh, V. Nath, Y. Tang, D. Yang, H. Roth and D. Xu, *Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images*, 2022. DOI: 10.48550/ARXIV.2201.01266. [Online]. Available: https://arxiv.org/abs/2201.01266.

[8] O. Petit, N. Thome, C. Rambour and L. Soler, *U-net transformer: Self and cross attention for medical image segmentation*, 2021. arXiv: 2103.06104 [eess.IV].

[9] A. Hatamizadeh, Z. Xu, D. Yang, W. Li, H. Roth and D. Xu, *Unetformer: A unified vision transformer model and pre-training framework for 3d medical image segmentation*, 2022. arXiv: 2204.00631 [eess.IV].

[10] B. Yun, Y. Wang, J. Chen, H. Wang, W. Shen and Q. Li, *Spectr: Spectral transformer for hyperspectral pathology image segmentation*, 2021. arXiv: 2103.03604 [eess.IV].

[11] H. Peiris, M. Hayat, Z. Chen, G. Egan and M. Harandi, *A robust volumetric transformer for accurate 3d tumor segmentation*, 2022. arXiv: 2111.13300 [eess.IV].

[12] R. Caruana, 'Multitask learning: A knowledge-based source of inductive bias,' in *International Conference on Machine Learning*, 1993.

[13] J. Baxter, 'A model of inductive bias learning,' *CoRR*, vol. abs/1106.0245, 2011. arXiv: 1106.0245. [Online]. Available: http://arxiv.org/abs/1106.0245.

[14] F. Saad and M. McCormack, *Prostate Cancer* (Comprendre la maladie et ses traitements). Unknown Publisher, 2012, ISBN: 9782923830049. [Online]. Available: https://books.google.no/books?id=fOxmtwAACAAJ.

[15] J. E. Oesterling, D. C. Rice, W. J. Glenski and E. J. Bergstralh, 'Effect of cystoscopy, prostate biopsy, and transurethral resection of prostate on serum prostate-specific antigen concentration,' *Urology*, vol. 42, no. 3, pp. 276–282, 1993, ISSN: 0090-4295. DOI: https://doi.org/10.1016/0090-4295(93)90616-I. [Online]. Available: https://www.sciencedirect.com/science/article/pii/009042959390616I.

[16] I. M. Thompson, D. K. Pauler, P. J. Goodman, C. M. Tangen, M. S. Lucia, H. L. Parnes, L. M. Minasian, L. G. Ford, S. M. Lippman, E. D. Crawford, J. J. Crowley and C. A. Coltman Jr, 'Prevalence of prostate cancer among men with a prostate-specific antigen level < or =4.0 ng per milliliter,' en, *N. Engl. J. Med.*, vol. 350, no. 22, pp. 2239–2246, May 2004.

[17] J.-L. Descotes, 'Diagnosis of prostate cancer,' *Asian Journal of Urology*, vol. 6, no. 2, pp. 129–136, 2019, Prostate Cancer : On the Road of Progress, ISSN: 2214-3882. DOI: https://doi.org/10.1016/j.ajur.2018.11.007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214388219300128.

[18] M. S. Litwin and H.-J. Tan, 'The Diagnosis and Treatment of Prostate Cancer: A Review,' *JAMA*, vol. 317, no. 24, pp. 2532–2542, Jun. 2017, ISSN: 0098-7484. DOI: 10.1001/jama.2017.7248. eprint: https://jamanetwork.com/journals/jama/articlepdf/2633921/jama_litwin_2017_rv_170003.pdf. [Online]. Available: https://doi.org/10.1001/jama.2017.7248.

[19] R. Kvåle, B. Møller, R. Wahlqvist, S. D. Fosså, A. Berner, C. Busch, A. E. Kyrdalen, A. Svindland, T. Viset and O. J. Halvorsen, 'Concordance between gleason scores of needle biopsies and radical prostatectomy specimens: A population-based study,' en, *BJU Int.*, vol. 103, no. 12, pp. 1647–1654, Jun. 2009.

[20] T. Penzkofer and C. M. Tempany-Afdhal, 'Prostate cancer detection and diagnosis: The role of mr and its comparison with other diagnostic modalities – a radiologist's perspective,' *NMR in Biomedicine*, vol. 27, no. 1, pp. 3–15, 2014. DOI: https://doi.org/10.1002/nbm.3002. eprint: https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/pdf/10.1002/nbm.3002. [Online]. Available: https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/abs/10.1002/nbm.3002.

[21] D. Yakar, O. A. Debats, J. G. Bomers, M. G. Schouten, P. C. Vos, E. van Lin, J. J. Fütterer and J. O. Barentsz, 'Predictive value of mri in the localization, staging, volume estimation, assessment of aggressiveness, and guidance of radiotherapy and biopsies in prostate cancer,' *Journal of Magnetic Resonance Imaging*, vol. 35, no. 1, pp. 20–31, 2012. DOI: https://doi.org/10.1002/jmri.22790. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jmri.22790. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/jmri.22790.

[22] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel and D. Hassabis, 'Mastering the game of go with deep neural networks and tree search,' *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.

[23] D. Weishaupt, V. D. Köchli, B. Marincek, J. M. Froehlich, D. Nanz and K. P. Pruessmann, *How does MRI work?: an introduction to the physics and function of magnetic resonance imaging*. Springer, 2006, vol. 2.

[24] Vijayalaxmi, M. Fatahi and O. Speck, 'Magnetic resonance imaging (MRI): A review of genetic damage investigations,' en, *Mutat. Res. Rev. Mutat. Res.*, vol. 764, pp. 51–63, Apr. 2015.

[25] T. A. G. M. Huisman, 'Diffusion-weighted imaging: Basic concepts and application in cerebral stroke and head trauma,' en, *Eur. Radiol.*, vol. 13, no. 10, pp. 2283–2297, Oct. 2003.

[26] A. Krizhevsky, I. Sutskever and G. E. Hinton, 'Imagenet classification with deep convolutional neural networks,' in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou and K. Weinberger, Eds., vol. 25, Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, 'Attention is all you need,' *CoRR*, vol. abs/1706.03762, 2017. arXiv: 1706.03762. [Online]. Available: http://arxiv.org/abs/1706.03762.

[28] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang and A. Dosovitskiy, 'Do vision transformers see like convolutional neural networks?' *CoRR*, vol. abs/2108.08810, 2021. arXiv: 2108.08810. [Online]. Available: https://arxiv.org/abs/2108.08810.

[29] J. L. Ba, J. R. Kiros and G. E. Hinton, *Layer normalization*, 2016. arXiv: 1607.06450 [stat.ML].

[30] K. He, X. Zhang, S. Ren and J. Sun, 'Deep residual learning for image recognition,' *CoRR*, vol. abs/1512.03385, 2015. arXiv: 1512.03385. [Online]. Available: http://arxiv.org/abs/1512.03385.

[31] M. Zelenčík, 'Using vision transformers for automated prostate cancer detection on mri scans,' Dec. 2022.

[32] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin and B. Guo, *Swin transformer: Hierarchical vision transformer using shifted windows*, 2021. DOI: 10.48550/ARXIV.2103.14030. [Online]. Available: https://arxiv.org/abs/2103.14030.

[33] O. Ronneberger, P. Fischer and T. Brox, 'U-net: Convolutional networks for biomedical image segmentation,' *CoRR*, vol. abs/1505.04597, 2015. arXiv: 1505.04597. [Online]. Available: http://arxiv.org/abs/1505.04597.

[34] L. G. Brown, 'A survey of image registration techniques,' *ACM Comput. Surv.*, vol. 24, no. 4, pp. 325–376, Dec. 1992, ISSN: 0360-0300. DOI: 10.1145/146370.146374. [Online]. Available: https://doi.org/10.1145/146370.146374.

[35] Y. Fu, Y. Lei, T. Wang, W. J. Curran, T. Liu and X. Yang, 'Deep learning in medical image registration: A review,' vol. 65, no. 20, 20TR01, Oct. 2020. DOI: 10.1088/1361-6560/ab843e. [Online]. Available: https://doi.org/10.1088%5C%2F1361-6560%5C%2Fab843e.

[36] S. Klein, M. Staring, K. Murphy, M. A. Viergever and J. P. W. Pluim, 'Elastix: A toolbox for intensity-based medical image registration,' *IEEE Transactions on Medical Imaging*, vol. 29, no. 1, pp. 196–205, 2010. DOI: 10.1109/TMI.2009.2035616.

[37] D. L. G. Hill, P. G. Batchelor, M. Holden and D. J. Hawkes, 'Medical image registration,' *Physics in Medicine & Biology*, vol. 46, no. 3, R1, Mar. 2001. DOI: 10.1088/0031-9155/46/3/201. [Online]. Available: https://dx.doi.org/10.1088/0031-9155/46/3/201.

[38] J. Ashburner and K. Friston, 'Rigid body registration,' in *Human Brain Function*, R. Frackowiak, K. Friston, C. Frith, R. Dolan, K. Friston, C. Price, S. Zeki, J. Ashburner and W. Penny, Eds., 2nd, Academic Press, 2003.

[39] S. J. Nelson, M. R. Day, P. J. Buffone, L. L. Wald, T. F. Budinger, R. Hawkins, W. P. Dillon, S. Huhn, M. D. Prados, S. Chang and D. B. Vigneron, 'Alignment of volume MR images and high resolution [18f]fluorodeoxyglucose PET images for the evaluation of patients with brain tumors,' en, *J Comput Assist Tomogr*, vol. 21, no. 2, pp. 183–191, Mar. 1997.

[40] S. Eberl, I. Kanno, R. R. Fulton, A. Ryan, B. F. Hutton and M. J. Fulham, 'Automated interstudy image registration technique for SPECT and PET,' en, *J Nucl Med*, vol. 37, no. 1, pp. 137–145, Jan. 1996.

[41] T. Mäkelä, P. Clarysse, O. Sipilä, N. Pauna, Q. C. Pham, T. Katila and I. E. Magnin, 'A review of cardiac image registration methods,' en, *IEEE Trans Med Imaging*, vol. 21, no. 9, pp. 1011–1021, Sep. 2002.

[42] K. Marstal, F. Berendsen, M. Staring and S. Klein, 'Simpleelastix: A user-friendly, multi-lingual library for medical image registration,' in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2016, pp. 574–582. DOI: 10.1109/CVPRW.2016.78.

[43] W. Crum, T. Hartkens and D. Hill, 'Non-rigid image registration: Theory and practice,' *The British journal of radiology*, vol. 77 Spec No 2, S140–53, Feb. 2004. DOI: 10.1259/bjr/25329214.

[44] C. de Boor, *A Practical Guide to Spline*. Jan. 1978, vol. Volume 27. DOI: 10.2307/2006241.

[45] J.-M. Guyader, W. Huizinga, D. H. J. Poot, M. van Kranenburg, A. Uitterdijk, W. J. Niessen and S. Klein, 'Groupwise image registration based on a total correlation dissimilarity measure for quantitative mri and dynamic imaging data,' *Scientific Reports*, vol. 8, 2018.

[46] R. Caruana, 'Multitask learning,' *Machine Learning*, vol. 28, no. 1, pp. 41–75, Jul. 1997.

[47] Y. Zhang and Q. Yang, 'An overview of multi-task learning,' *National Science Review*, vol. 5, no. 1, pp. 30–43, Sep. 2017, ISSN: 2095-5138. DOI: 10.1093/nsr/nwx105. eprint: https://academic.oup.com/nsr/article-pdf/5/1/30/31567358/nwx105.pdf. [Online]. Available: https://doi.org/10.1093/nsr/nwx105.

[48] Y. Zhang and Q. Yang, 'A survey on multi-task learning,' *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 12, pp. 5586–5609, 2022. DOI: 10.1109/TKDE.2021.3070203.

[49] S. Ruder, 'An overview of multi-task learning in deep neural networks,' *CoRR*, vol. abs/1706.05098, 2017. arXiv: 1706.05098. [Online]. Available: http://arxiv.org/abs/1706.05098.

[50] Y. Yuan, E. Ahn, D. Feng, M. Khadra and J. Kim, *Z-ssmnet: A zonal-aware self-supervised mesh network for prostate cancer detection and diagnosis in bpmri*, 2022. arXiv: 2212.05808 [eess.IV].

[51] Z. Dong, Y. He, X. Qi, Y. Chen, H. Shu, J.-L. Coatrieux, G. Yang and S. Li, *Mnet: Rethinking 2d/3d networks for anisotropic medical image segmentation*, 2022. arXiv: 2205.04846 [eess.IV].

[52] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen and K. H. Maier-Hein, 'nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation,' en, *Nat. Methods*, vol. 18, no. 2, pp. 203–211, Feb. 2021.

[53] Z. Zhou, V. Sodha, J. Pang, M. B. Gotway and J. Liang, 'Models genesis,' *Medical Image Analysis*, vol. 67, p. 101 840, Jan. 2021. DOI: 10.1016/j.media.2020.101840. [Online]. Available: https://doi.org/10.1016%2Fj.media.2020.101840.

[54] N. Debs, A. Routier, C. Abi Nader, A. Marcoux, F. Nicolas, A. Bône and M. M. Rohé, 'Deep learning for detection and diagnosis of prostate cancer from bpmri and psa: Guerbet's contribution to the pi-cai 2022 grand challenge.'

[55] P. F. Jaeger, S. A. A. Kohl, S. Bickelhaupt, F. Isensee, T. A. Kuder, H.-P. Schlemmer and K. H. Maier-Hein, *Retina u-net: Embarrassingly simple exploitation of segmentation supervision for medical object detection*, 2018. arXiv: 1811.08661 [cs.CV].

[56] A. Karagoz, M. E. Seker, M. Yergin, T. A. Kan, M. S. Kartal, E. Karaarslan, D. Alis and I. Oksuz, *Prostate lesion estimation using prostate masks from biparametric mri*, 2023. arXiv: 2301.09673 [physics.med-ph].

[57] X. Li, S. Vesal, S. Saunders, S. John, C. Soerensen, H. Jahanandish, S. Moroianu, I. Bhattacharya, R. Fan, G. Sonn and M. Rusu, 'The prostate imaging: Cancer ai (pi-cai) 2022 grand challenge (pimed team).'

[58] A. Seetharaman, I. Bhattacharya, L. C. Chen, C. A. Kunder, W. Shao, S. J. C. Soerensen, J. B. Wang, N. C. Teslovich, R. E. Fan, P. Ghanouni, J. D. Brooks, K. J. Too, G. A. Sonn and M. Rusu, 'Automated detection of aggressive and indolent prostate cancer on magnetic resonance imaging,' *Medical Physics*, vol. 48, no. 6, pp. 2960–2972, 2021. DOI: https://doi.org/10.1002/mp.14855. eprint: https://aapm.onlinelibrary.wiley.com/doi/pdf/10.1002/mp.14855. [Online]. Available: https://aapm.onlinelibrary.wiley.com/doi/abs/10.1002/mp.14855.

[59] S. J. C. Soerensen, R. E. Fan, A. Seetharaman, L. Chen, W. Shao, I. Bhattacharya, Y.-H. Kim, R. Sood, M. Borre, B. I. Chung, K. J. To'o, M. Rusu and G. A. Sonn, 'Deep learning improves speed and accuracy of prostate gland segmentations on magnetic resonance imaging for targeted biopsy,' en, *J. Urol.*, vol. 206, no. 3, pp. 604–612, Sep. 2021.

[60] H. Kan, L. Qiao, J. Shi and H. An, 'Implementation method of the pi-cai challenge (swangeese team).'

[61] H. Kan, J. Shi, M. Zhao, Z. Wang, W. Han, H. An, Z. Wang and S. Wang, 'Itunet: Integration of transformers and unet for organs-at-risk segmentation,' in *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2022, pp. 2123–2127. DOI: 10.1109/EMBC48229.2022.9871945.

[62] M. Tan and Q. V. Le, 'Efficientnet: Rethinking model scaling for convolutional neural networks,' *CoRR*, vol. abs/1905.11946, 2019. arXiv: 1905.11946. [Online]. Available: http://arxiv.org/abs/1905.11946.

[63] A. Saha, J. J. Twilt, J. S. Bosma, B. van Ginneken, D. Yakar, M. Elschot, J. Veltman, J. Fütterer, M. de Rooij and H. Huisman, *The PI-CAI Challenge: Public Training and Development Dataset*, version v1.0, Zenodo, May 2022. DOI: 10.5281/zenodo.6517398. [Online]. Available: https://doi.org/10.5281/zenodo.6517398.

[64] A. Saha, J. J. Twilt, J. S. Bosma, B. van Ginneken, D. Yakar, M. Elschot, J. Veltman, J. Fütterer, M. de Rooij and H. Huisman, 'Artificial Intelligence and Radiologists at Prostate Cancer Detection in MRI: The PI-CAI Challenge,' 2022. DOI: 10.5281/zenodo.6522364.

[65] A. Saha, J. J. Twilt, J. S. Bosma, B. van Ginneken, D. Yakar, M. Elschot, J. Veltman, J. Fütterer, M. de Rooij and H. Huisman, 'Artificial Intelligence and Radiologists at Prostate Cancer Detection in MRI: The PI-CAI Challenge (Study Protocol),' 2022. DOI: 10.5281/zenodo.6667655.

[66] R. Beare, B. Lowekamp and Z. Yaniv, 'Image segmentation, registration and characterization in r with simpleitk,' *Journal of Statistical Software*, vol. 86, no. 8, pp. 1–35, 2018. DOI: 10.18637/jss.v086.i08. [Online]. Available: https://www.jstatsoft.org/index.php/jss/article/view/v086i08.

[67] Z. R. Yaniv, B. C. Lowekamp, H. J. Johnson and R. Beare, 'Simpleitk image-analysis notebooks: A collaborative environment for education and reproducible research,' *Journal of Digital Imaging*, vol. 31, pp. 290–303, 2017.

[68] B. Lowekamp, D. Chen, L. Ibanez and D. Blezek, 'The design of simpleitk,' *Frontiers in neuroinformatics*, vol. 7, p. 45, Dec. 2013. DOI: 10.3389/fninf.2013.00045.

[69] T. Briand and P. Monasse, 'Theory and Practice of Image B-Spline Interpolation,' *Image Processing On Line*, vol. 8, pp. 99–141, 2018, https://doi.org/10.5201/ipol.2018.221.

[70] D. Chicco, 'Ten quick tips for machine learning in computational biology,' *BioData Mining*, vol. 10, p. 35, Dec. 2017. DOI: 10.1186/s13040-017-0155-3.

[71] D. M. W. Powers, 'Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation,' 2020. DOI: 10.48550/ARXIV.2010.16061. [Online]. Available: https://arxiv.org/abs/2010.16061.

[72] M. Consortium, *Monai: Medical open network for ai*, version 1.1.0, Dec. 2022. DOI: 10.5281/zenodo.7459814. [Online]. Available: https://doi.org/10.5281/zenodo.7459814.

[73] D. Shamonin, E. Bron, B. Lelieveldt, M. Smits, S. Klein and M. Staring, 'Fast parallel image registration on cpu and gpu for diagnostic classification of alzheimer's disease,' eng, *Frontiers in Neuroinformatics*, vol. 7, pp. 50–50, 2014, ISSN: 1662-5196.

[74] H. Lester and S. R. Arridge, 'A survey of hierarchical non-linear medical image registration,' *Pattern Recognition*, vol. 32, no. 1, pp. 129–149, 1999, ISSN: 0031-3203. DOI: https://doi.org/10.1016/S0031-3203(98)00095-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320398000958.

[75] W. Venderink, M. de Rooij, J. M. Sedelaar, H. J. Huisman and J. J. Fütterer, 'Elastic versus rigid image registration in magnetic resonance imaging–transrectal ultrasound fusion prostate biopsy: A systematic review and meta-analysis,' *European Urology Focus*, vol. 4, no. 2, pp. 219–227, 2018, ISSN: 2405-4569. DOI: https://doi.org/10.1016/j.euf.2016.07.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405456916301018.

[76] L. Zhang, Q. Yang, X. Liu and H. Guan, *Rethinking hard-parameter sharing in multi-domain learning*, 2021. DOI: 10.48550/ARXIV.2107.11359. [Online]. Available: https://arxiv.org/abs/2107.11359.

[77] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille and Y. Zhou, *Transunet: Transformers make strong encoders for medical image segmentation*, 2021. arXiv: 2102.04306 [cs.CV].

[78] D. Singla, F. Cimen and C. A. Narasimhulu, 'Novel artificial intelligent transformer U-NET for better identification and management of prostate cancer,' *Molecular and Cellular Biochemistry*, Nov. 2022.

[79] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez and P. Luo, 'Segformer: Simple and efficient design for semantic segmentation with transformers,' 2021. DOI: 10.48550/ARXIV.2105.15203. [Online]. Available: https://arxiv.org/abs/2105.15203.

Appendix A

Prediction examples

In this section we provide an overview of tumor segmentation predictions (red color) of our best model from Experiment 5 and compare them to the label (green color). For the purpose of this view we used prediction threshold 0.25 - all pixels above value 0.25 are considered as tumor and enclosed with red line.

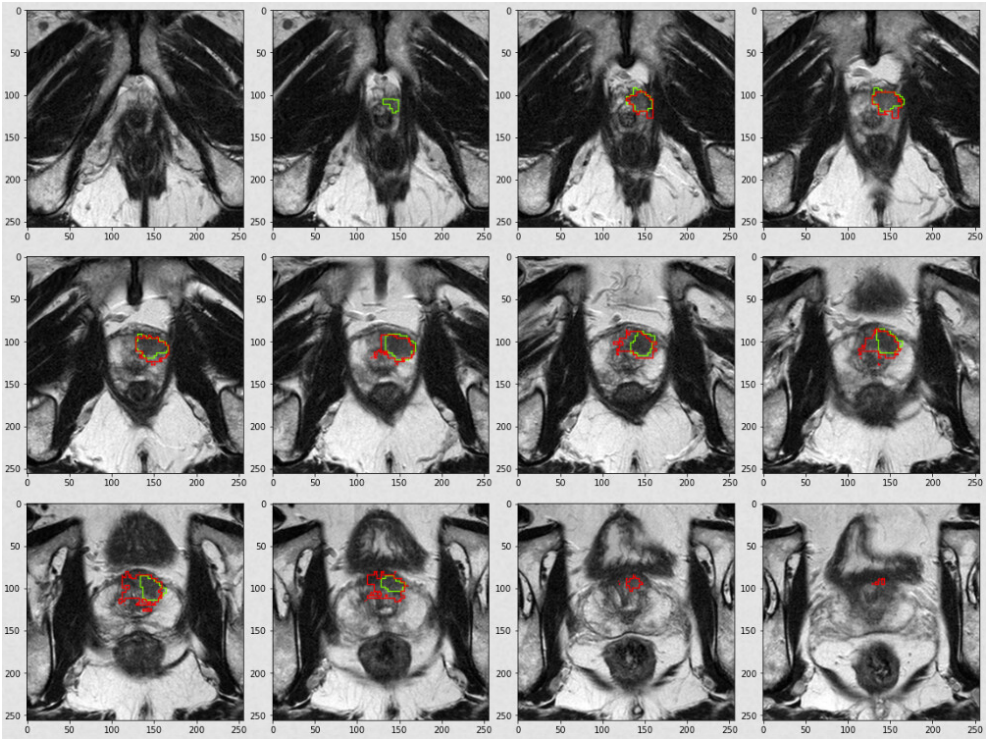


Figure A.1: Example 1

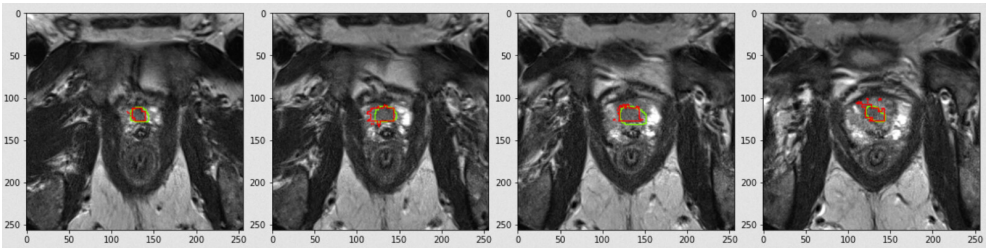


Figure A.2: Example 2

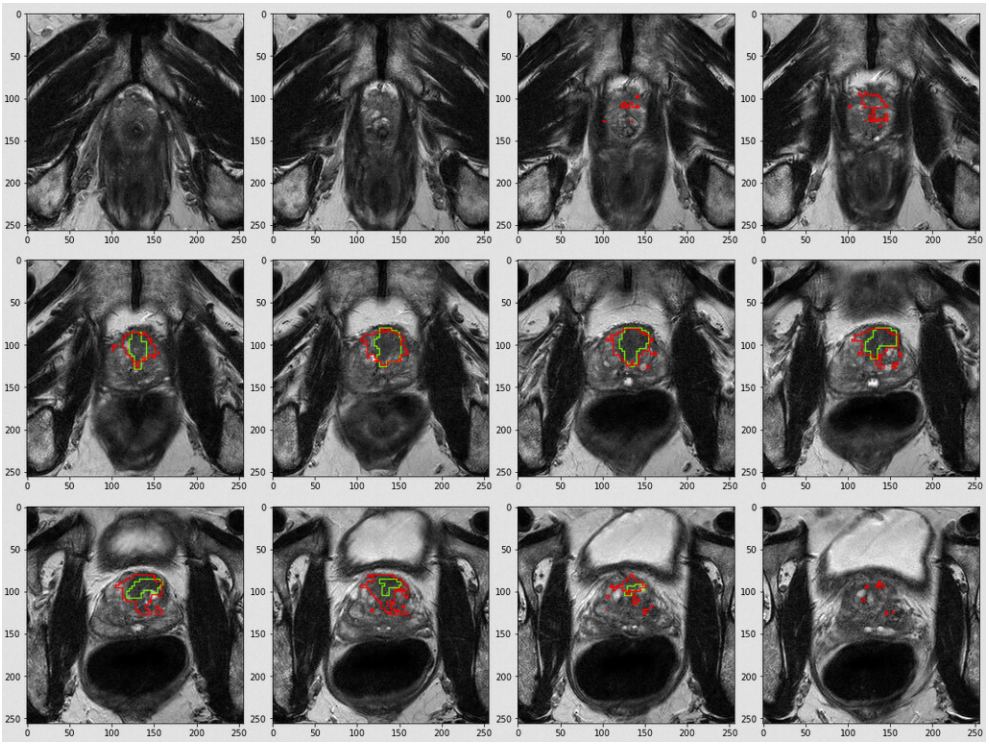


Figure A.3: Example 3

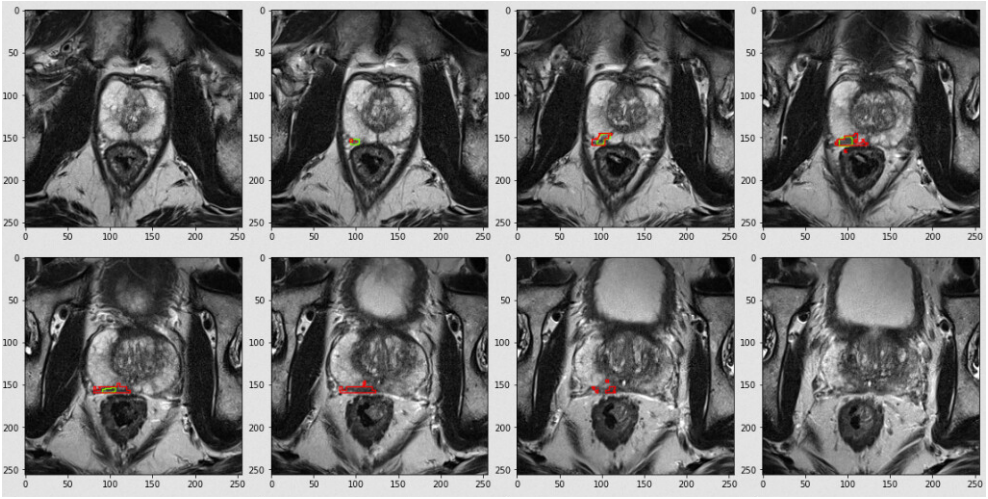


Figure A.4: Example 4

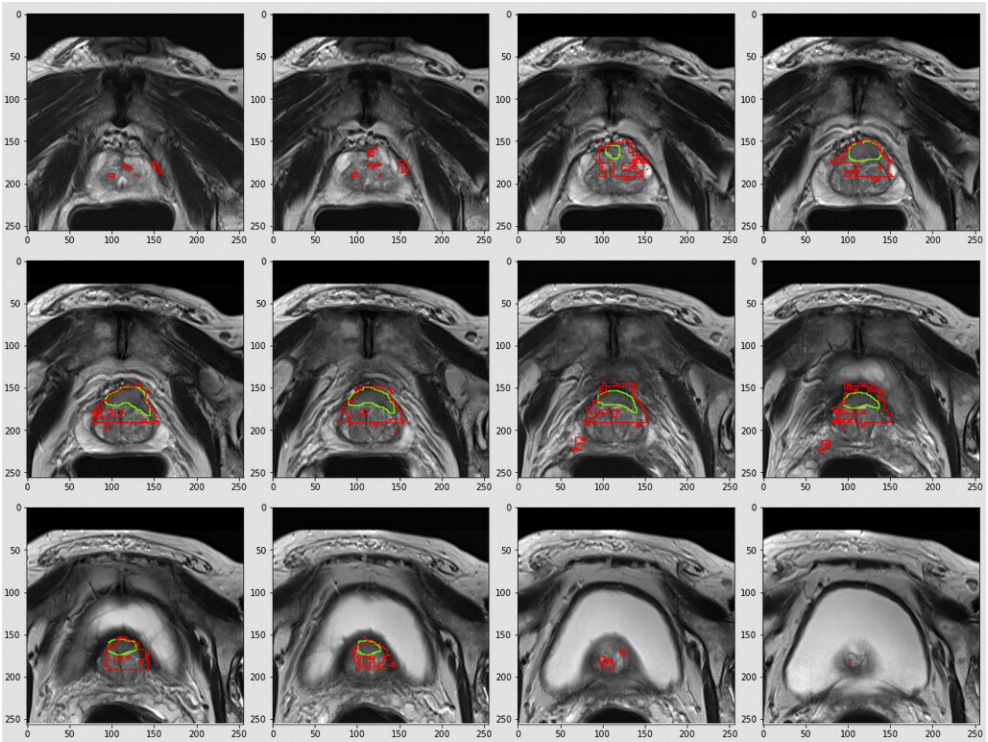


Figure A.5: Example 5

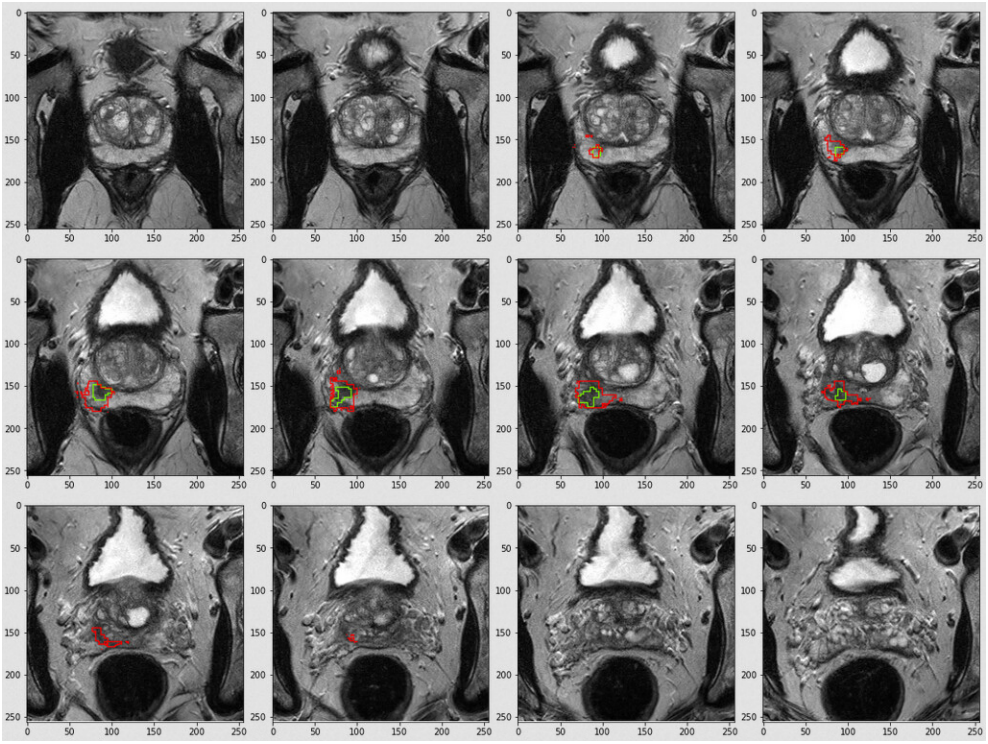


Figure A.6: Example 6

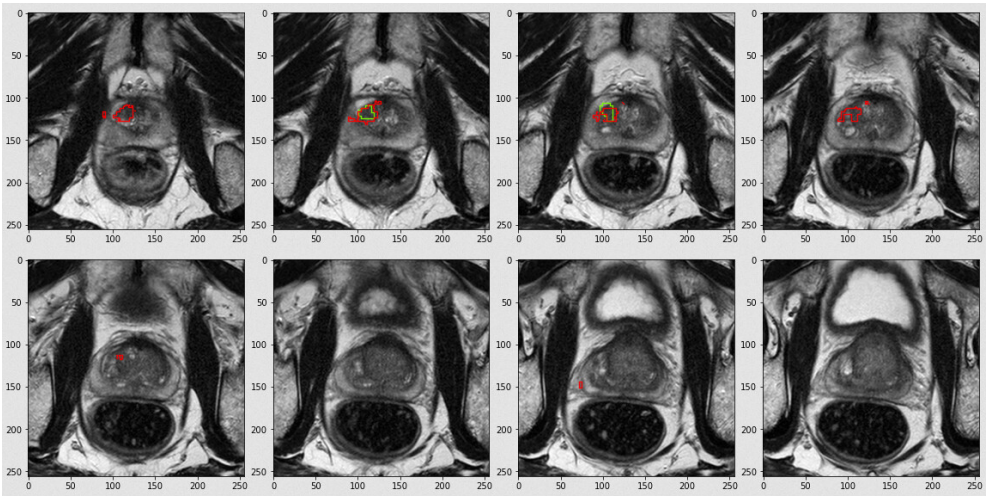
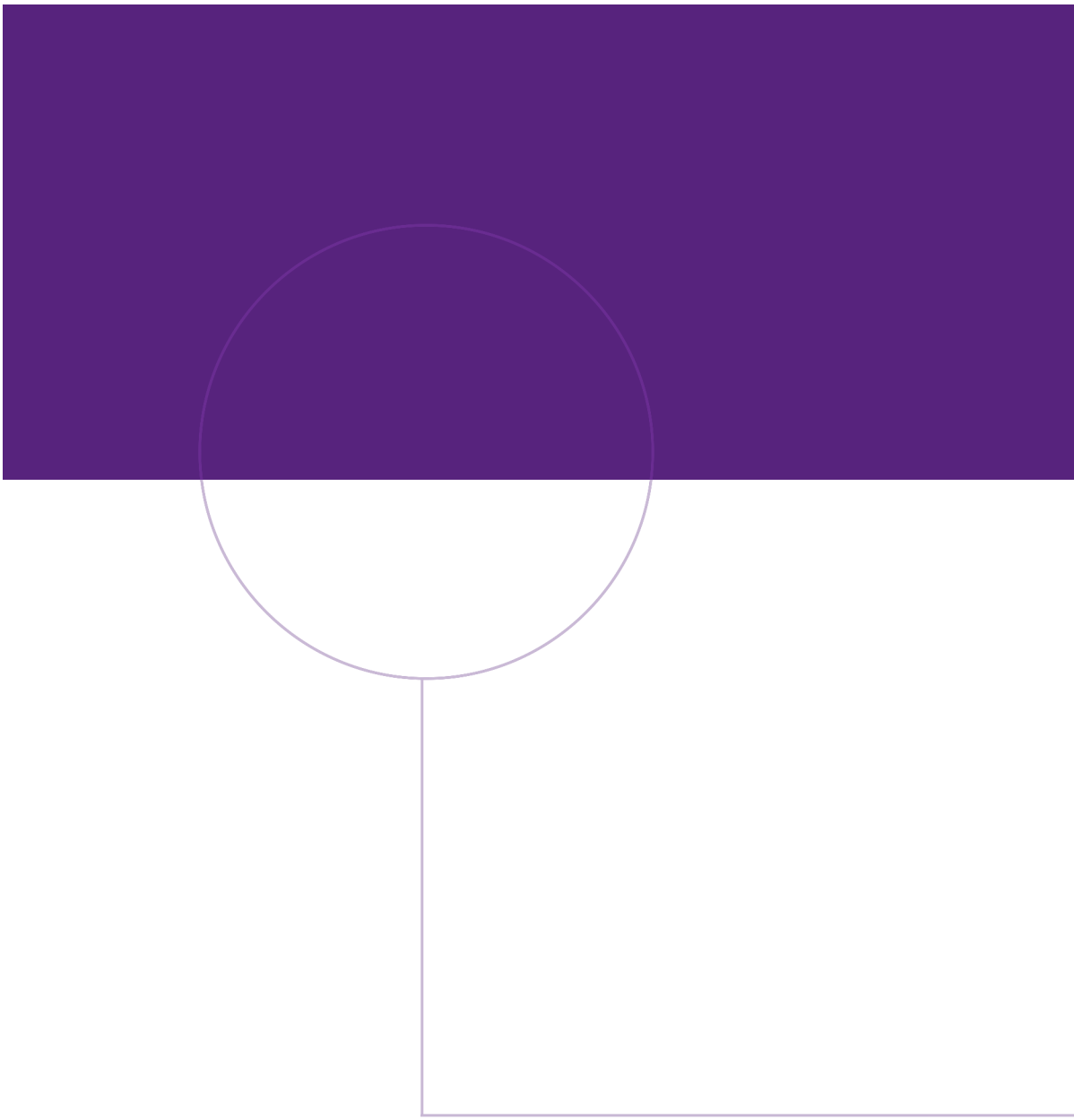


Figure A.7: Example 7



NTNU

Norwegian University of
Science and Technology