Faculty of
Mathematics
and Informatics

VILNIUS UNIVERSITY
FACULTY OF MATHEMATICS AND INFORMATICS
MODELLING AND DATA ANALYSIS
MASTER'S STUDY PROGRAMME

Kidney Tumour Segmentation in Computed Tomography Images using One-Stage and Two-Stage Approaches under Limited Data

Master's thesis

Author: Kamilė Dementavičiūtė
VU email address: kamile.dementaviciute@mif.stud.vu.lt
Supervisor: assoc. prof. dr. Linas Petkevičius
Consultant: sr. rsch. dr. Jonas Venius

Vilnius

2023

Summary

This thesis investigates one-stage and two-stage approaches for kidney tumor segmentation in Computed Tomography (CT) images, when dealing with labeled data scarcity. The rapid development of Deep Learning (DL) has turned medical image analysis into a research hotspot, but accurately segmenting tumors in medical images remains a challenging task, particularly when dealing with limited data, which is a common problem in the medical setting. The thesis proposes a novel approach to two-stage semantic segmentation, employing the latest YOLOv7 object detection model. Having implemented multiple approaches for dealing with labeled data scarcity, such as data augmentation and fine-tuning, the results of the experiments concluded the proposed fine-tuned two-stage approach to achieve an increase of 2.4% overall Dice score, across all patients. Further investigation has found this method to be significantly more successful in the segmentation of a small tumor, which was undetected by the baseline one-stage approach. Although the results come in line with reviewed literature, they should be considered with caution, due to the poor population representability in the training and test set splits of a small dataset.

Keywords: Kidney Tumour Segmentation, Semantic Segmentation, Computed Tomography Images, U-Net, Two-Stage Segmentation

Santrauka

Šiame darbe nagrinėjami vieno ir dviejų pakopų inkstų navikų segmentavimo būdai kompiuterinės tomografijos vaizduose, dirbant su itin mažu sužymėtų duomenų kiekiu. Spartus giliojo mokymosi vystymasis pavertė medicininių vaizdų analizę mokslinių tyrimų centru, tačiau tiksliai segmentuoti navikus medicininiuose vaizduose išlieka sudėtinga užduotis, ypač dirbant su maža imtimi. Šiame baigiamajame darbe yra pasiūlomas naujas dviejų pakopų semantinio segmentavimo metodas, naudojant naujausią YOLOv7 objektų aptikimo modelį. Pritaikius kelis metodus, skirtus sužymėtų duomenų trūkumo problemai spręsti, kaip duomenų augmentaciją bei modelio adaptaciją, eksperimentų rezultatai parodė, jog siūlomas adaptuotas dviejų pakopų metodas, pagerino vienos pakopos segmentacijos rezultatus 2,4% Dice koeficiento metrikos. Tolesnis tyrimas parodė, kad šis metodas yra daug sėkmingesnis segmentuojant mažus navikus, kurie nebuvo aptikti naudojant pradinį vienos pakopos metodą. Nors rezultatai atitinka peržvelgtus eksperimentus literatūros apžvalgoje, į juos reikėtų žiūrėti atsargiai, kadangi mažo duomenų rinkinio paskirstymas į mokymo ir testavimo dalis, dažnai neatspindi tikros populiacijos.

Raktiniai žodžiai: Inkstų Navikų Segmentavimas, Semantinis Segmentavimas, Kompiuterinės Tomografijos Vaizdai, U-Net, Dviejų Pakopų Segmentavimas

Acronyms

3D three-dimensional

AI Artificial Intelligence

AP Average Precision

CNN Convolutional Neural Network

CT Computed Tomography

DCNN Deep Convolutional Neural Network

DL Deep Learning

DNN Deep Neural Network

FCN Fully Convolutional Network

IoU Intersection over Union

mAP Mean Average Precision

ML Machine Learning

MRI Magnetic Resonance Imaging

NMS Non-Maximal Suppression

NN Neural Networks

PN Partial Nephrectomy

RCC Renal Cell Carcinoma

ReLU Rectified Linear Unit

ROI Region Of Interest

SOTA State-Of-The-Art

TNM Tumour Node Metastasis

UI Ultrasound Imaging

Contents

Introduction

Cancer is one of the leading causes of death worldwide, with 18.1 million new cases and nearly 10 million deaths in 2020 alone [43]. Accurate diagnosis and effective treatment often requires a well-defined representation of a patient's anatomy. Medical imaging techniques such as CT or Magnetic Resonance Imaging (MRI) scans can provide this, however manual evaluation by clinicians is time-consuming and subject to human error. To improve both diagnosis and treatment effectiveness, there is a need to develop standardized tools to automate medical image segmentation. Automated segmentation could not only increase the chances of identifying a tumor, but also significantly speed up and standardize the process of dimension and characteristic acquisition, leading to more accurate cancer stage definition. Consequently, this would improve more suitable treatments prescriptions and optimal outcomes.

With the continually advancing technological progress, a search for new ways of aiding medical professionals is at the forefront of Artificial Intelligence (AI) technology implementation in the health-care sector. The rapid development of DL has turned medical image analysis into a research hotspot [21]. Semantic segmentation is one of the fundamental methods employed in medical image analysis. It divides the image into several regions based on feature similarity. The use of this method allows doctors to perform qualitative or even quantitative analysis of regions of interest, as well as to evaluate the effect before and after treatment, this way greatly reducing the workload and significantly improving the accuracy and reliability of medical diagnosis.

The core of a DL-based semantic segmentation network is built on a Convolutional Neural Network (CNN) inspired deep structure like Fully Convolutional Network (FCN) [22] as well as the encoder-decoder architecture. The innovation in the network mainly comes from the continuous optimization of the encoder and decoder structure and the improvement of its efficiency [21]. The described method consist of a single network, hence it is called the one-stage approach. However, there is another common approach, which has been proved to bring better results [2], and that is the two-stage semantic segmentation. In this case the first stage acts as a region detector and in the second, the final segmentation is produced. The two-stage approach decreases the area that needs to be segmented, creating an easier task for the chosen segmentation model.

This thesis investigates the employment of one-stage and introduces a novel approach to two-stage semantic segmentation technique with limited data for the specific case of kidney tumours in CT images, provided by the National Cancer Institute of Lithuania. According to the World Cancer Research Fund, in 2020 kidney cancer was the 14th (9th for men, 14th for women) [15] most commonly occurring cancers worldwide. Looking deeper at the statistics of this condition, Lithuania has some of the most reasons for concern as it had the highest overall rate of kidney cancer that year, with an approximate number of 14.5 diagnoses (age-standardised rate) per 100,000 of the population, which is over three times more than the global average [14]. The numbers of kidney cancer occurrences have been increasing since 1980, according to the estimated projections of Cancer Research UK [36]. The rapid rise of cases is likely to continue and it is expected to be one of the fastest increasing cancers over the next 20 years. This creates an unarguable motivation to increase efforts in ensuring a more accurate and timely diagnosis and successful treatment for the possibly increasing influx of patients.

The results of the thesis concluded the suggested two-stage approach to achieve an increase of up to 2.4% of overall Dice score, across all patients. Further investigation has found the two-stage approach to be more successful in the segmentation of small tumors, which were undetected by the baseline one-stage approach. Although the results come in line with the reviewed literature, they should be considered with caution, due to the poor population representability in the training and test splits of a small dataset.

Challenges

Semantic segmentation problems have a good set of challenges, working with medical data only adds to this. One of the biggest challenges is the scarcity of labelled data. Despite their high performance, even the most advanced segmentation models still require large amounts of high-quality annotated data in order to perform well. Acquiring and annotating high-quality datasets, particularly in the field of medical imaging, can be time-consuming and require specialized expertise, making it an expensive endeavor. The privacy of medical data is also a concern, as strict regulations and guidelines are in place to protect patient privacy and ensure the ethical use of this sensitive information. As a result, the availability of medical datasets is limited, and researchers may need to obtain specific bioethical permissions in order to access and use the data for research purposes.

Intra-variability and inter-variability are important factors to consider when working with labeled medical image data. Tumor labeling can be challenging due to the wide range of sizes, shapes, and appearances that tumors can have, and different annotators or groups of annotators may have different levels of expertise or experience in identifying and labeling tumors. This can cause a variety of problems for DL models, as inaccurate or inconsistent labels can result in poor model performance and make it difficult to evaluate and compare the results of different studies or to reproduce the results of a particular study.

Another challenge is added by the fact that medical image segmentation is a cross-disciplinary field. Pathological and other clinical conditions can be both complex and diverse. Working with them requires in depth understanding of the clinical needs and other subtleties, which AI scientists might struggle with. While clinicians, on the other hand, often lack comprehensive understanding of the specific technology of AI. As a result, AI integration in the clinical setting becomes more burdensome and often cannot meet the specific clinical needs well.

Kidney tumors in particular, can be difficult to detect from medical images, because they have a very heterogeneous texture and can be hard to distinguish from cysts or nearby organs. Furthermore, kidney tumor segmentation in large abdominal CT scans poses problems with balancing the receptive field (the size of the input region that affects the features of a particular layer) of the network with the target spacing for resampling (the size of the output image). For accurate segmentation, it is important for the receptive field to be large enough to capture enough context about the tumor, but not so large that it includes unnecessary or irrelevant information. While if the target spacing is too large, important details may be lost due to downsampling, if it is too small, the algorithm may be slower and more computationally expensive. In order to reduce the memory requirements of the computing devices, it may be necessary to crop or downsample the images, which leads to resolution loss and an increase in class imbalance in the input batches, which can degrade the performance of the segmentation algorithm [41].

Finally, DL-based image segmentation methods have still limited applicability for specific image analysis problems, where higher accuracy is needed. Segmenting lesions or abnormalities in medical images demands a higher level of accuracy than what is desired in natural images [47]. Inaccurate segmentation in medical images can lead to clinical mistakes, so more effective image segmentation architectures are needed to accurately capture the fine details of target objects.

All the above mentioned factors make it difficult for the existing methods to accurately segment kidney tumors. There are many issues to be addressed and a real need for more effective approaches for medical image segmentation. These approaches may involve the development of novel algorithms or improved methods for dealing with labeled data scarcity, which will be investigated in this thesis.

Goal of the Study

The goal of this thesis is to conduct research on kidney tumour segmentation problem in CT images, to compare one-stage and two-stage segmentation approaches and to suggest the best working method for dealing with labeled data scarcity.

Objectives of the Study

1. Review and analyse scientific literature related to semantic segmentation and object detection in medical imaging, as well as dealing with a small dataset.
2. Present the clinical context for kidney pathology.
3. Present and explain the methodology behind the carried out experiments.
4. Employ a baseline one-stage and a novel approach to the two-stage semantic segmentation techniques for CT scans from the National Cancer Institute of Lithuania.
5. Identify and employ the most suitable methods for dealing with labeled data scarcity.
6. Compare the results of the different segmentation approaches, identify the most beneficial techniques and draw a conclusion.

1 Related Work

This chapter contains relevant semantic segmantation and object detection literature review. Additionally, for dealing with labeled data scarcity, data augmentation and fine-tuning approaches have been investigated and are reviewed at the end of this chapter.

1.1 Semantic Segmentation

The state-of-the-art (SOTA) models for image segmentation are primarily variants of the FCN and the encoder-decoder architecture like U-Net [31]. The study of FCN [22] for semantic segmentation was the first article that applied DL to image segmentation and achieved outstanding results. Since then, many image segmentation models have borrowed from it [21]. Convolution based classification models such as VGG [34] and ResNet [11] produced only one-dimensional probability output information. To create a pixel-by-pixel classification, Long et al. [22], added a deconvolution layer at the end, which upsampled the feature map of the last convolution layer and therefore restored the image to the input size, see Figure 1. This was the start of deep learning implementation for the semantic segmentation task.

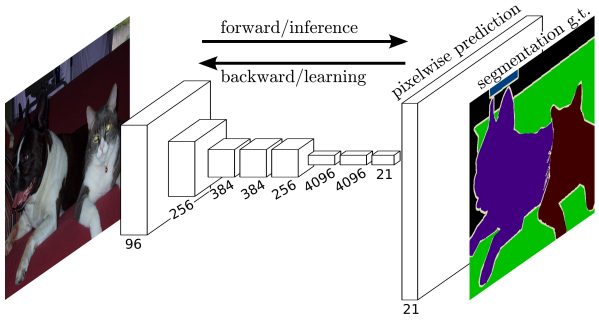


Figure 1: Architecture of FCN, from [22]

One of the first instances of the symmetric encoder-decoder structure can be found in the architecture of SegNet [4], a model that is built on top of FCN. The encoder analyses object information, while the decoder corresponds the parsed information into the final image form. Architecturally similar to the encoder-decoder structure of SegNet, U-Net [31], a groundbreaking model for biomedical image segmentation was presented in 2015. Famous for its architecture resembling a U shape, the model introduces novelty skip connections in between the encoder and decoder that allow the higher resolution features from the encoder to be combined with the upsampled output.

U-Net has been an unparalleled success in medical image segmentation thanks to its ability to incorporate both low and high level information. Free of any fully connected layers, the model relies solely on convolution produced feature maps, for which the full context is available in the input image. This strategy allows comparably easy segmentation of large images. Due to its excellent performance in terms of the small number of necessary annotated data and fast computational speed, various U-Net-based semantic segmentation methods have been introduced.

All the methods discussed above require only a single model for the task and are therefore referred to as one-stage.

Two-stage Methods

Many approaches have been developed to aid one-stage semantic segmentation models, one of them being an additional stage prior to the final segmentation model. The goal of the first stage is to reduce the segmentation area turning it into an easier segmentation task. In the specific case of kidney tumor segmentation, two-stage semantic segmentation can be approached in two ways, one approach could be

to first segment kidney organs and in the second stage to segment the tumors. Another approach would be to first detect the Region Of Interest (ROI) and to segment it in the second stage. The first approach may be more difficult to implement as it requires additional labeling from medical professionals, which can be time-consuming and resource-intensive to obtain.

The ROI detection in two-stage segmentation models also varies. In medical imaging, a common two-stage approach is for similar architectures to be used in both of the stages [2]. Amiri, Mina, et al. implemented a basic U-Net for both of the stages of breast tumor segmentation task from ultrasound images [2]. The implementation is explained in Figure 2. Wang, C., et al. took the use of U-Net for both of the stages even further by proposing a two-stage Deep Convolutional Neural Network (DCNN) framework which was built by concatenating two U-Net-like networks [41]. Inspired by the architecture of image super-resolution CNN (SRCNN), DCNN is able to sufficiently work with original sized medical images, addressing resolution loss in the input data, which is a common cause of performance downgrade in DCNN based segmentation algorithms.

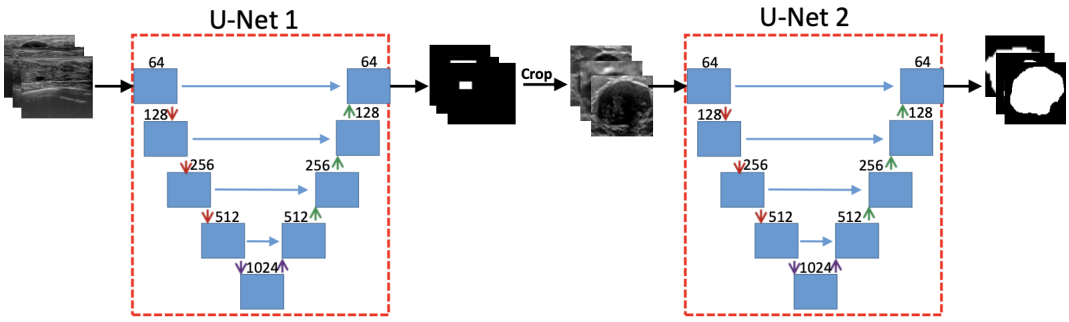


Figure 2: Architecture of a two-stage semantic segmentation approach using one U-Net for detection and one for segmentation, from [2].

Another approach for the two-stage method is implementing an object detection model for the first stage, alongside a semantic segmentation model for the second one. A good example of a successful implementation is the liver detection and segmentation model DSL, by Tang, Wei, et al. [39]. In the first stage, the authors use an improved Faster Regions with CNN features (Faster R-CNN) to detect approximate position of the liver. And in the second stage, the obtained images are processed and input into a DeepLab semantic segmentation model to obtain the final segmentation masks. DSL outperformed the State-Of-The-Art (SOTA) solutions in terms of volume overlap error, average surface distance, relative volume difference, and total score.

Object detection and segmentation approach has been employed in many computer vision applications, with different variations of RCNN, YOLO and SSD as the proposed object detection algorithms [2]. Although it has been suggested that object detection networks may be too complex for the task of detecting objects in medical images and may be limited by the availability of labeled data [2], it is important to continue research and exploration of their potential use.

Object detection is a valuable tool in the medical field for automatically identifying and segmenting objects such as organs, tumors, lesions, and abnormalities. As imaging technology and object detection algorithms continue to improve, this technique is becoming increasingly accurate at capturing subtle differences between different classes of objects, allowing it to detect abnormalities that may be missed by clinical experts [46]. Investigating further how to employ new, more advanced architectures and how to adjust the different object detection models for two-stage segmentation, could shed some light on how to optimize object detection in medical imaging and potentially lead to more accurate and efficient methods for semantic segmentation application.

1.2 Object Detection

DL-based object detection has become increasingly popular in recent years due to the availability of powerful computing hardware and large datasets [46]. Initially, DL-based object detection was a two-stage process, which consisted of object localization and classification tasks. As a result of the increasing development of DL, DCNNs have become more important for object detection (see Figure 3 for the classification of different types of object detection approaches). Most algorithms use a CNN to extract features from the image, to predict the probability of learned classes. The first milestone in applying deep CNNs for object detection was achieved in 2014, when RCNN (Regions with CNN features) [44] was proposed. The advantage of such the two-stage approach was high detection accuracy, however the disadvantage - relative complexity and slow detection speed [44].

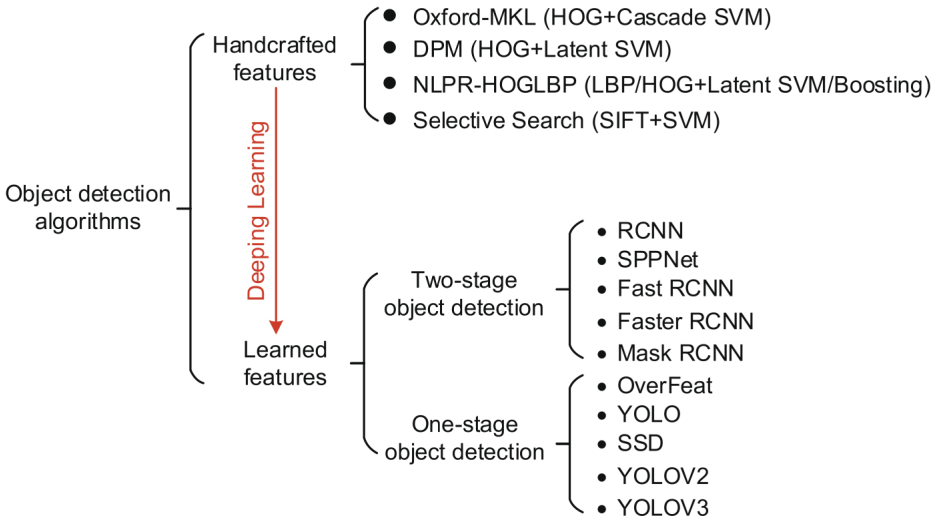


Figure 3: Scheme of different object detection algorithm types with model examples, from [46].

One-stage object detection first came to light in 2013, when OverFeat model was introduced [44]. It utilized feature sharing of CNNs to integrate object classification and localization into one network architecture. Although when compared with RCNN, OverFeat has obvious advantages in speed, it lacks in accuracy, hence up until 2015, RCNN models were the best way to perform object detection [44]. However this soon changed as in 2015 YOLO (You Only Look Once) model came out [30]. It was able to implement class probabilities and bounding-box regression directly from a full-size image, which made it a revolutionary object detection algorithm, fast enough for real-time application. Ever since, YOLO has remained one of the premiere methods for its task for three primary reasons: it's accuracy, relatively low cost, and ease of use. However the model had a number of localization errors and low recall, which had to be addressed and improved by later versions of the model.

Having undergone multiple iterations of development, YOLOv7 [42] is the latest version of the popular algorithm, and improves significantly on its predecessors [42]. YOLOv7 came out in July 2022, and presented relatively better accuracy results as well as a significant improvement in speed. These results are achieved by making a number of changes to the YOLO network architecture and training routines, which will be expanded on in Section 3.3.1.

1.3 Dealing with Labelled Data Scarcity

1.3.1 Image Augmentation

As mentioned previously, one of the biggest challenges for the task of medical image segmentation, employing DL-based algorithms, is the scarcity of labelled data. Image segmentation algorithms based on DCNNs rely heavily on large datasets in order to build a robust and generalizable model and to avoid overfitting [33]. Unfortunately many domains, such as medical image analysis suffer from the lack of big data. This is especially true, when it comes to labelled datasets, as accurately labeling medical images requires considerable skill and effort. Data augmentation is perhaps one of the most popular and straightforward solution techniques, the goal of which is to enlarge the training set by creating new augmented images from the current ones. As a result, this enhances the model's generalisation and consequently performance on new unseen data.

In 2021 Chlap, P. et al. have published an extensive literature review summarising the findings from 149 papers that employ augmentation techniques used in SOTA DL models for segmentation from CT and MRI scans [7]. In their paper, they divide augmentation techniques into four distinct categories: basic, deformable, DL-based and other augmentation techniques. The majority of the reviewed papers (62%) employed what is referred to as the *basic* augmentation techniques. These techniques include simple transformations which map the points of the image to a different position, or manipulates the image intensity values, for the production of a new image. Although the techniques are simple to apply, they have proved to be very effective in improving the trained model's performance [7].

When the basic augmentation techniques do not provide sufficient variability to make the subsequent model generalisable, the remaining techniques are often employed. DL-based augmentation approaches, reported to have been used in around 25% of the reviewed papers, can automatically learn the representations of images and generate realistic new images. According to the review paper, most of the use cases of such methods were conducted to aid image classification or segmentation tasks.

1.3.2 Transfer-Learning

Transfer learning is another valuable approach for training models based on CNNs, to perform medical image segmentation in settings with limited data [48]. It is defined as "the capability of a system to recognize and employ the knowledge learned in a previous source domain to a novel task" [32]. It involves using a model that has already been trained on a larger dataset and adapting it to the specific task at hand.

Transfer learning for medical image segmentation is usually employed using the following two approaches: fine-tuning the network pre-trained on general images or pre-trained on medical images for a different target organ or task [12]. Although transfer learning reduces the training time on the target task, improvements in segmentation accuracy are highly task and data-dependent [17]. Studies have shown that when fine-tuning on a similar domain, even using a small dataset can improve medical segmentation model performance [48]. For tasks related to medical imaging, it is often not advised to use general images as the outsourced data [12]. Better performance has been proved when the tasks of source and target network are more similar, but sometimes even transferring the weights of far distant tasks has been proved to be better than random initialization [12].

Furthermore, [12] has classified transfer learning into three levels:

> **Full network adaption,** where the weights are initialized by a pre-trained network (instead of random initialization) and are all updated during the training of the target task. The weight update is often performed with a very small learning rate.
>
> **Partial network adaption,** where the weights are initialized by a pre-trained network but the weights of certain layers are frozen, while others are updated during the training.

Zero adaption, where the weights are initialized by a pre-trained network and are not updated at all. Generally, this approach from another medical task is not recommended due to the huge variation in the appearance of the segmentation target.

One of the most important considerations that has to be made when fine-tuning is the choice of which layers (if any) should be frozen. From the reviewed literature, the most common approach still seems to be trial-and-error type of testing, by freezing different layers to find the optimal. The common practice in transfer learning is to keep the shallow layers unchanged and to modify deeper (layers closer to the output) layers according to the new dataset [3] [17]. At least for medical image segmentation, this challenges the common belief that the encoder section needs to learn data and task-specific representations [17]. However, recently, when analysing fine-tuning using a U-Net, it was shown that fine-tuning the last layers of the network, which is the common practice for classification networks, is often the worst strategy [3]. Although many different views can be observed on the optimal strategy when it comes to freezing layers, most reviewed studies seem to agree that regardless of the task and the datasets, the full network adoption approach is almost always successful in increasing model performance [38].

Although there are numerous studies reporting dramatic increases in model performance using all levels of transfer learning, there are some experiments that delivered better results when training from scratch. Therefore the specific fine-tuning approach has to be considered carefully, acknowledging both of the datasets at hand, the source data, segmentation model and the task itself.

2 Clinical Context

This chapter gives an overview of Kindey pathology overall, as well as most common current diagnosis and treatment methods. At the end of the chapter, CT imaging technology is introduced.

2.1 Kidney Pathology Overview

Kidney or renal pathology is a complex medical science field that focuses on the diagnosis and treatment of diseases related to the kidneys. It covers a wide range of medical conditions including kidney infections, kidney failure, chronic kidney disease and of course kidney tumours. The diagnosis of renal pathology usually includes a combination of clinical tests, medical image analysis, and biopsy. Treatment options depend on the severity and type of kidney pathology, and can include dialysis, medications, lifestyle changes, and surgery.

This thesis concentrates on the specific type of pathology - kidney tumours. The proliferation of cells in the kidney can lead to the formation of a tumour, commonly referred to as kidney cancer. The most common type of kidney cancer, called Renal Cell Carcinoma (RCC) represents about 90% of all kidney cancers. RCCs begin their growth by forming a cover over the tubules (tiny tubes in the kidneys that return nutrients, fluids, and other substances that have been filtered from the blood, see Figure 4) in the kidney and are grown rapidly if not detected and treated at an early stage.

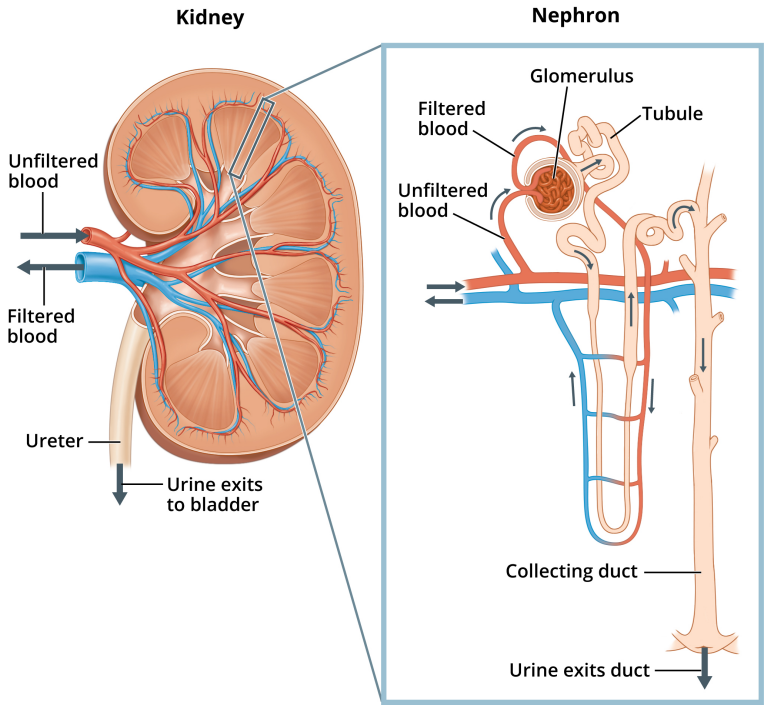


Figure 4: Image of a close up nephron and its place in the kidney. Labels on the kidney cross section show where unfiltered blood enters, filtered blood leaves, and urine exits. On the nephron, the glomerulus, tubule, and collecting duct are labeled along with where unfiltered blood enters, filtered blood exits, and urine exits, from [26].

Annually, over 400,000 new cases of kidney cancer are recorded globally, with over 170,000 deaths [24]. Kidney cancer is most common in people between the ages of 65 and 74 and similarly as in the global context, in Lithuania kidney cancer is twice as common in men as it is in women, with the approximate death rate of 34% and 23% respectively. According to the ECIS - European Cancer Information System, Kidney cancer incidences are expected to rise by 9.3% in Lithuania and by 20.1% in the European Union by 2040 [37]. Unfortunately, the mortality statistics are even more dreary,

showing a 20.5% and 33.6% increase in Lithuania and the EU respectively. Despite being one of top 10 cancer killers [27] and the concerning forecast, research and public awareness surrounding this condition have been relatively low, thus restraining further advancement in detection and treatment [27].

Diagnosis

Renal tumours are usually detected using imaging techniques such as CT and MRI scans. Additionally, ultrasound has been found to be useful in detecting small renal masses that may not show up on CT or MRI scans. Further tests such as intravenous pyelogram (IVP), angiography, and biopsy may also be employed for a more accurate diagnosis.

Early-stage diagnosis of kidney tumors can significantly improve the chances of recovery. If the disease is detected soon enough, the patient can benefit from different treatment options. The determination of the size, shape, location and morphology of a renal mass is important to decide the type of treatment, therefore medical imaging and its research plays a vital role. Currently, the diagnosis and detection are the primary emphases of renal kidney cancer-related research besides recognizing whether the tumor is malignant or benign [40].

The Tumour Node Metastasis (TNM) stage is a way of classifying the extent of the cancer based on the size and location of the tumor, whether the cancer has spread to nearby lymph nodes, and whether it has spread to other parts of the body (metastasized), see Figure 6. Understanding the TNM stage of the disease can help doctors determine the most appropriate treatment options and provide a more accurate prognosis for the patient. The most favorable stage for diagnosis is undoubtedly stage I. Tumors of the first stage have a size smaller than 7cm as can be seen in Figure 5, are organ confined, and do not involve Gerota's fascia (a fibrous envelope of tissue that surrounds the kidney) or have vascular invasion. Patients with stage I disease were reported to have a 5-year survival of approximately 97% [10]. On the other hand, patients who present with or later develop metastatic disease have a universally poorer prognosis as effective treatment options are limited, see Figure 6. All four stages of RCC are visualised in Figure 5.

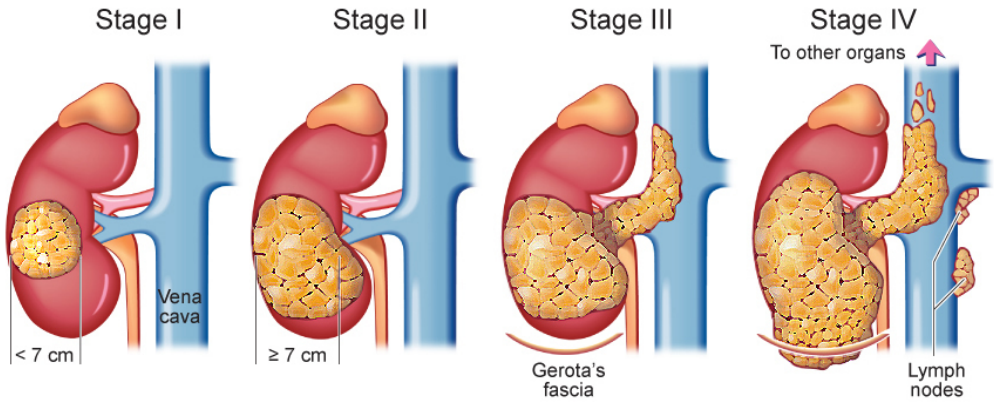


Figure 5: Four stages of renal cancer, from [1]

Treatments

For many years patients with kidney cancer had few treatment options beyond surgery and the survival rates rarely exceeded one year, however over the past 15 years there has been an explosion of promising new treatment methods [27], as minimally invasive therapies started being introduced and developed further. Such treatment methods include Partial Nephrectomy, also cryo, thermo and

		Primary Tumor (T)				Regional Nodal	Distal	5-y
		Size, cm	Venous Involvement	Beyond Gerota's Fascia	Ipsilateral Adrenal	Involvement (N)	Metastasis (M)	Survival
Stage I	A	<4	No	No	No	No	No	97%
	B	>4 but <7	No	No	No	No	No	
Stage II	A	>7 but <10	No	No	No	No	No	84%
	B	>10	No	No	No	No	No	
Stage III	A	Any	Renal vein	No	No	No	No	65%
	B	Any	Infradiaphragmatic IVC	No	No	No	No	
	C	Any	Supradiaphragmatic IVC	No	No	No	No	
		Any	Any	No	No	Yes	No	
Stage IV		Any	Any	Yes	No	Any	No	12%
		Any	Any	No	Yes	Any	No	
		Any	Any	No	No	Any	Yes	

Figure 6: Table kidney tumor patient 5 year survival rates, influenced by the TNM stage at diagnosis, from [10].

microwave ablation as well as stereotaxic radiation therapy. Currently, surgery is considered as the most prevalent treatment for kidney cancer [13]. In the past, the standard procedure used to be Radical Nephrectomy, during which both the removal of the tumour and the affected kidney was performed. The advancements in surgery together with earlier tumour detection has paved way to a more conservative kidney cancer treatment such as nephron sparing procedures, called Partial Nephrectomies, in this case only a small part of the kidney is removed. Therefore accurate and efficient methods of tumour segmentation is essential to decide between the treatment methodologies [28].

Medical imaging techniques can be used not only to detect kidney tumors at an early stage, but also to precisely locate the tumor, which can help to ensure that during a Partial Nephrectomy (PN) procedure, only the cancerous tissue is removed while preserving as much healthy kidney tissue as possible. This approach has become increasingly popular for the treatment of small renal masses. However, proper planning is necessary to minimize the risk of complications, and using 3D reconstruction models from the segmented images can help to optimize treatment [35].

2.2 Computer Tomography Imaging

In most digital images, each pixel states the direct measurement of the color intensity, however there is a number of other techniques to record various properties that can be constructed into an image, many methods can be observed in the area of medical imaging. A good example of this is MRI, where magnetic fields are acquired in Fourier or frequency space and an image is contracted from this information. In the case of CT images, a computerized x-ray imaging procedure is performed.

In CT scanners, a motorized x-ray source rotates around a circular opening in a structure called a gantry (Figure 7). During the scan, the patient lies on a bed that moves slowly through the gantry as the x-ray tube rotates around the patient, shooting narrow beams of x-rays through the body. Instead of using film like traditional tomographic imaging methods, CT scanners use special digital x-ray detectors that are positioned opposite the x-ray source. As the x-rays pass through the patient, they are picked up by the detectors that send the collected information to a computer, which creates cross-sectional images, or "slices." These successive slices can be digitally combined to form a three-dimensional (3D) image of the patient, which makes it easier to identify basic structures as well as any tumors or abnormalities [25].

Currently, alongside X-ray, Ultrasound Imaging (UI), MRI and positron-emission tomography (PET), CT is the most widely used medical imaging technology. The emergence of medical digital image sources, such as computed tomography, has raised new challenges in the field of data processing. Before non-universal data storage formats were used, which made it difficult to share data between devices of

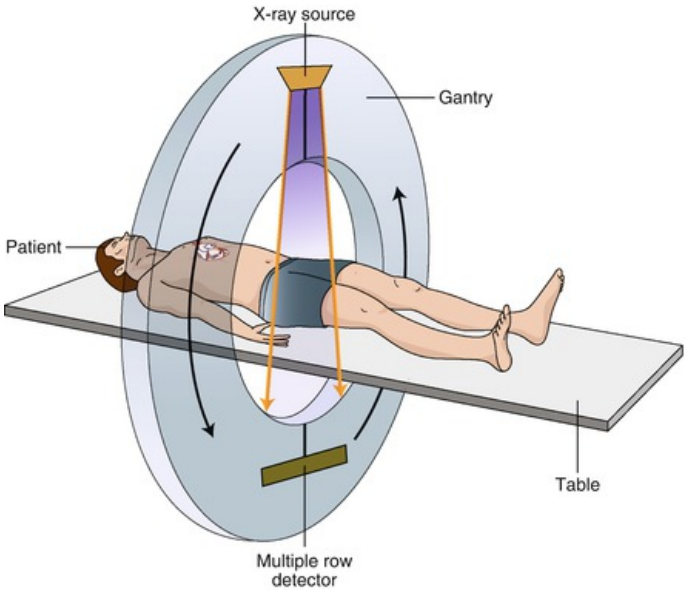


Figure 7: A cartoon depiction of a typical CT scanner, from [9].

different manufacturers. Meanwhile, the amount of medical data has grown rapidly, which raised the need to create a common and standardized method of storing and sharing it. In 1993 Digital Imaging and Communications in Medicine (DICOM) was published, which has remained as the standard ever since.

3 Methodology

This chapter introduces the main methodology behind the carried out experiments. Starting with a very basic introduction of deep learning and digital images, the basic understanding of semantic segmentation and object detection as well as more in depth explanation of the used models (U-Net and YOLOv7), training process and performance evaluation criteria.

3.1 Deep Learning

Till around the early 2000s, the primary segmentation methods were traditional approaches such as edge detection-based, threshold-based and region-based segmentation [45]. These techniques employed knowledge of digital image processing and mathematics, however often struggled to achieve high accuracy in segmentation. With the rapid development of AI, DL has gained a prominent role in image segmentation since the 2010s due to its ability to deliver more accurate results at faster speeds compared to traditional methods. The use of DL in medical image segmentation can effectively help doctors detect tumour regions, confirm the size and location, quantitatively evaluate the effect before and after treatment, greatly reducing the workload of the medical professionals [21].

DL is a Machine Learning (ML) method based on Artificial Neural Networks (ANNs). It uses a Deep Neural Network (DNN) to simulate the learning process of the human brain and extract features from large-scale data (images, audio, text, etc.) [21]. When a neural network contains more than one hidden layer, it is considered to be a *deep* neural network. With increasing depth, the network is able to extract more complex features and consequently solve more difficult problems.

3.1.1 Artificial Neural Networks

As DNNs are inspired by the structure of the human brain, they also consist of thousands or even millions neuron like mathematical models, called artificial neurons. Similar to the synapses in the brain, these simple processing nodes are densely interconnected. In addition to neurons and synapses, Neural Networks (NN)s also have weights, biases, and activation functions. The following section will briefly explains the role of each of these components in the functioning of a perceptron.

Artificial Neurons

A single neuron can be effectively demonstrated using a perceptron model. A perceptron is a single layer NN. It takes inputs $x_1, x_2, ...x_n$, which are multiplied with weights $w_1, w_2, ...w_n$ and adds a bias term b, then computes a linear function z, on which an activation function f (e.g. sigmoid) is applied and output y is produced. The linear function, followed by the activation are written in equation (1).

$$z = \sum_{i=1}^{n} w_i x_i + b, \qquad y = f(z) \tag{1}$$

This is called the weighted sum of the inputs. A neuron can get its input directly from data (input layer neurons), as in the case of a perceptron, or from the outputs of other neurons (hidden layers neurons). The higher the weight for the specific input, the higher the importance of that specific input. The weights and biases are called the unknown parameters of the model. The (sub)optimal values for those parameters are found during the parameter estimation (training) process.

Activation Functions

Choosing an activation function is critical for the performance of a NN model. The function determines how the weighted sum of a neuron's inputs is transformed into the output, and can be broadly classified into two categories: linear and non-linear. Non-linear activation functions are the

most widely used, as they enable the network to model non-linear relationships in the data, which are typically found in real-world scenarios. Without non-linear activation functions, a neural network would be unable to capture these complex patterns, making them an essential element in the design of effective models.

Typically, the same activation function is used in all the hidden layers, while the output layer uses a different one, depending on the type of prediction needed. For example in the case of this experiments the activation function for the output is sigmoid (logistic) (2). The advantage of the sigmoid function for probability prediction stems from the fact that the output exists in the range $[0, 1]$, which can be interpreted as a probability. It is often used for binary classification tasks, which in the specific case of this experiment means classifying a pixel to either: *tumour* or *not tumour*.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

Among the most popular activation functions for the hidden layers, that is also used in the segmentation model of this experiment, is Rectified Linear Unit (ReLU) function (3). For any real number x, it is defined as

$$ReLU(x) = \begin{cases} x & \text{if} \quad x > 0 \\ 0 & \text{if} \quad x \leq 0 \end{cases} \tag{3}$$

The main advantages of using ReLU over other activation functions is computational efficiency, reduced likelihood of vanishing gradient and faster training speeds.

The visualisations of the mentioned activation functions are presented in Figure 8.

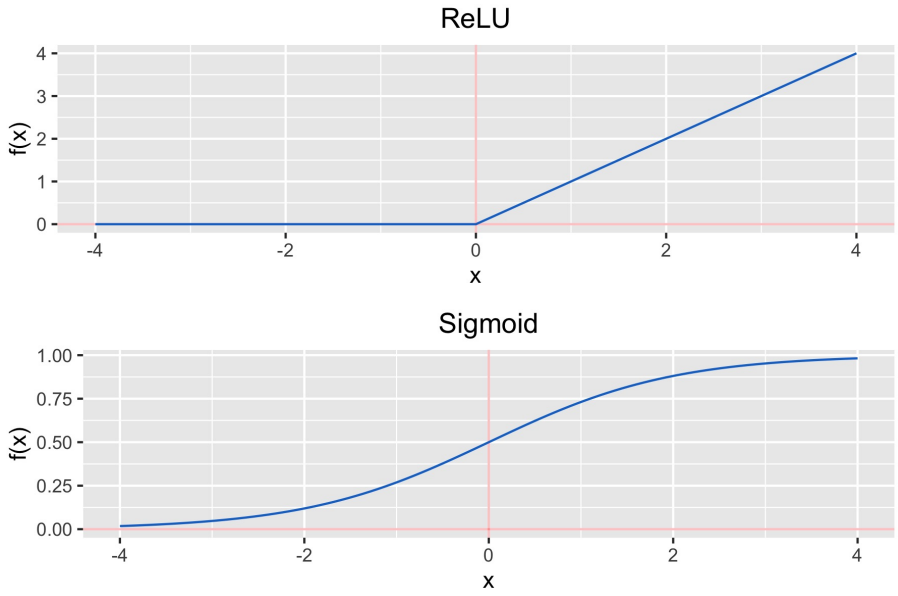


Figure 8: Activation functions

3.1.2 Network Training

The experiments of this thesis will approach kidney tumor semantic segmentation as a supervised learning task. Supervised learning is a type of ML where a model is trained to make predictions using labeled (ground truth) data.

For this task, the available data is split into three sets: train, validation and test. The train set is used to train the model. During training, the model is presented with the input-output pairs and the

algorithm adjusts the parameters to minimize the difference between the predicted output and the true output. The validation set is used to evaluate the model's performance on a dataset that is separate from the training set and to tune the model's hyperparameters (settings that are not learned during training and are used to control the learning process, e.g. learning rate, batch size etc.). And finally, the test set is used to evaluate the model's performance on unseen data and to provide an unbiased estimate of the model's generalization error (difference between the model's performance on the training set and its performance on the test set).

The training of a DL model includes optimizing a loss function using an optimization algorithm. The loss function measures the difference between the model's predictions and the ground truth labels, and the optimization algorithm adjusts the model's parameters to minimize this difference. The process of adjusting the model's parameters is called backpropagation, which involves calculating the gradient of the loss function with respect to the model's parameters and using this gradient to update the parameters in the direction that reduces the loss. The training process continues until a predefined stopping condition has been met.

3.1.3 Image Data

When working with image data, it is important to understand the structure of digital images. A digital image is an image made of pixels, or in other words - picture elements. Every such image is stored in a form of a matrix, where every value is numerical and represents the corresponding pixel (or voxel in 3D images) intensity. Pixel intensities are finite and discreet values in the range $[0, 255]$. Values closer to 0 represent darker shades and vice versa. In the case of CT images, darker shared correspond to low-density objects (air in the lungs), while lighter ones - to high-density objects (bones). Greyscale images have a single channel, while coloured images consist of three channels, one for red, green and blue (RGB) tones.

In DL tasks, image data is often preprocessed and transformed in order to make it more suitable for use in a specific model or to ease the learning process. This can involve resizing the images to a consistent size, normalizing pixel/voxel values, and possibly augmenting the dataset by applying various transformations such as cropping and flipping.

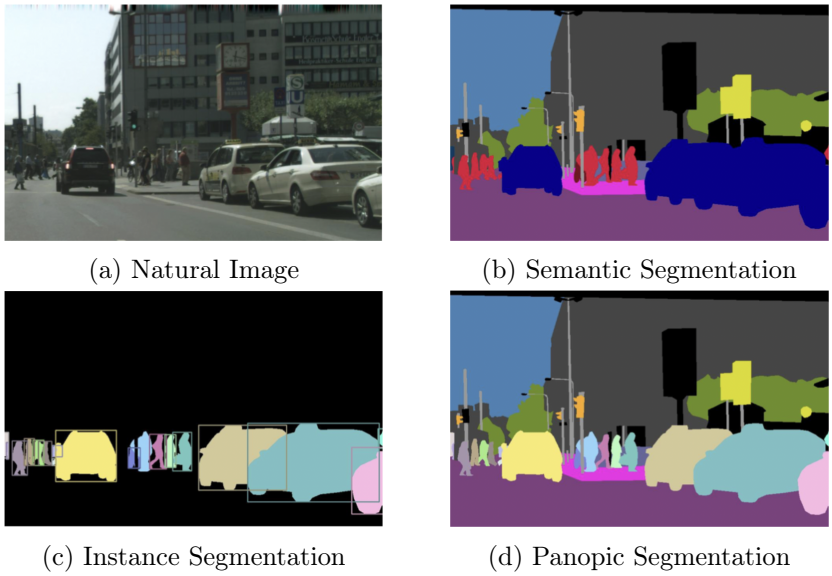
3.2 Image Segmentation

Image segmentation is a computer vision technique that divides an image into multiple regions or segments based on the characteristics of the pixels within that image. Image segmentation is a low-level (pixel-level) vision task, as it focuses on the spatial information within an image and aims to label pixels in a way that reflects shared characteristics such as color, intensity, and texture. Image segmentation is an important and difficult part of image processing. In the field of medical imaging, this process is used to extract meaningful information for easier diagnosis and analysis, surgical planning, and treatment monitoring.

Image segmentation can be divided into three categories: semantic, instance and panopic segmentation, see Figure 9.

3.2.1 Semantic Segmentation

The type of image segmentation used in the experiments of this thesis is semantic segmentation. It is a classification task, where every pixel in the image is assigned a class label. The specific task at hand is a binary classification, as there are only two classes that the pixels can be assigned to: *tumour* and *not tumour* or in other words *background*.



(a) Natural Image (b) Semantic Segmentation

(c) Instance Segmentation (d) Panopic Segmentation

Figure 9: Examples of different image segmentation methods, from [18]

.

3.2.2 Mathematical Problem Formulation

All the necessary mathematical notations for the task formulation are listed below.

2D input data $X \in \mathbb{R}^{1\times d_w \times d_h}$, here 1 stands for a single color channel, as the images are greyscale, d_{x_w} and d_{x_h} are the input image width and height dimension respectively

Ground truth labels $Y \in \mathbb{R}^{1\times d_w \times d_h}$

Dataset $\mathcal{D} = \{X^{(i)}, Y^{(i)}\}_{i=1}^{N}$, where N is the number of samples

Mathematical model $f_\theta(\mathbf{X})$ or f_θ

Unknown model parameters $\theta \subset \mathbb{R}^{d_\theta}$, to be estimated from the input data

Parameter estimate $\hat{\theta} \subset \mathbb{R}^{d_\theta}$

Loss function $\mathcal{L}_\theta$, to evaluate the parameters θ

The goal of the task is for the chosen objective function to be minimised. For the problem of semantic segmentation, lets consider a parametric function f with unknown parameters $\theta \in \Theta \subset \mathbb{R}^{d_\theta}$, in which an optimization task of $\mathcal{L}_\theta(f_\theta(\mathbf{X}); \mathcal{D})$ will be considered. Now the formulation can be rewritten as:

$$\mathcal{L}_\theta(f_\theta(\mathbf{X}); \mathcal{D}) = \mathcal{L}_\theta(\hat{\mathbf{Y}}, \mathbf{Y}) \rightarrow \min_\theta,\ \theta \in \Theta \subset \mathbb{R}^{d_\theta}$$

The desired output image can be noted as follows

$$\hat{Y} \in (0,1)^{1\times d_w \times d_h} \subset \mathbb{R}^{1\times d_w \times d_h}$$

here $\hat{Y} = f_{\hat{\theta}}(X)$.

3.2.3 U-Net

The first model that was implemented in the experiments of this thesis is the classical U-Net. It is an architecture developed in 2015 by Olaf Ronneberger et al. [31] for biomedical image segmentation. It is a fully convolutional neural network that is designed to learn from fewer training samples. Its architecture utilizes an encoder-decoder structure, see Figure 10, which consists of four encoder and four decoder blocks that are connected via a bottleneck.

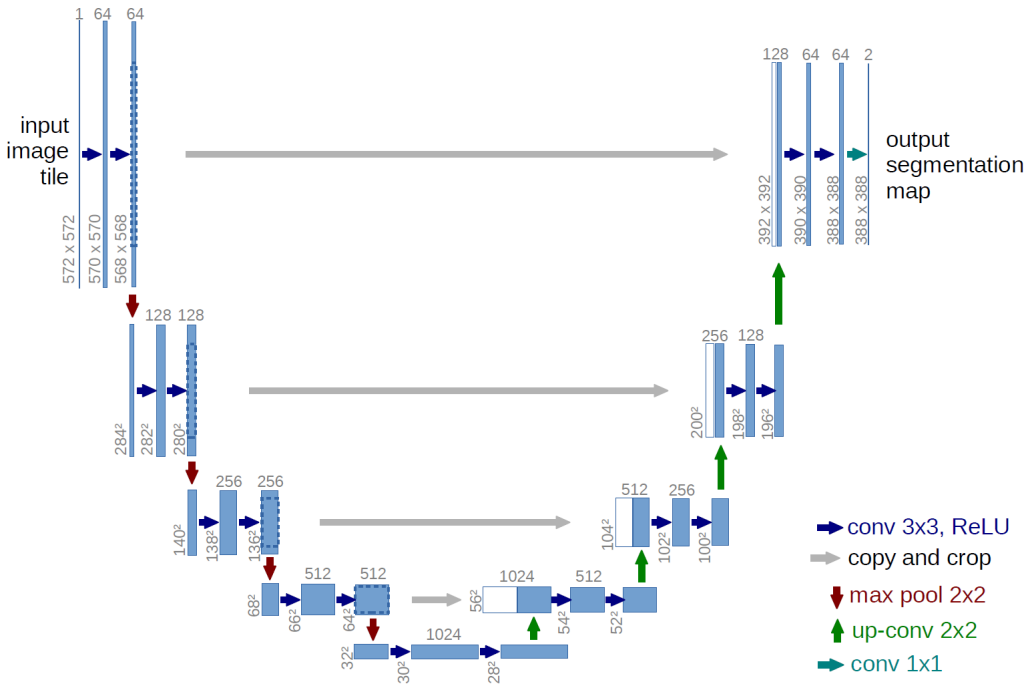


Figure 10: U-Net architecture. The blue boxes correspond to multi-channel feature maps, with the number of channels denoted at the top. The image dimensions are at the lower left edge of each box. The white boxes represent the copied feature maps.

Encoder Network extracts features from the input image and passes them to the decoder network. The encoder consists of a sequence of blocks, each composed of two 3×3 convolutions followed by ReLU activation functions. After each block, a 2×2 max-pooling is applied to reduce spatial dimensions by half while decreasing computational cost due to the reduced number of trainable parameters. The output of each encoder block additionally passes through a skip connection to the corresponding block in the decoder, which helps to reconstruct the original input image from its abstract representation.

Skip Connections allow information to be passed from encoder layers of the network directly to the decoder. This helps the decoder generate more accurate semantic features, while also creating a shortcut for gradients flowing back through the network without being diminished. Skip connections also help to optimize the gradient flow during backpropagation, allowing the model to learn better representations.

Bottleneck (the horizontal part) contains a compressed representation of the input data. It connects the encoder and the decoder network and completes the flow of information. This compressed view should only contain abstract and high-level information to be able to construct the output. The bottleneck consists of two 3×3 convolutions, where each convolution is followed by a ReLU activation function.

Decoder Network takes an abstract representation and generates a semantic segmentation mask. Each block starts with a 2×2 transpose convolution, followed by the concatenation of corresponding features from earlier layers (via skip connections) from the encoder block. Two 3×3 convolutions are then used, each followed by a ReLU activation function. The output passes through a 1×1 convolution with sigmoid activation, which produces the pixel-wise classification segmentation mask.

3.2.4 Hyperparameters

Hyperparameters are variables that determine the network structure and how the network is trained. In the following experiments, only the training-related hyperparameters were adjusted. All the relevant hyperparameters are listed below:

Epoch indicates the number of times all of the training data has been passed through the training process of the model. The number of epochs equals the number of iterations if the batch size is the entire training dataset.

Batch size (in the case of a computer vision task) is the number of images that go through the network in one iteration, before the internal parameters are updated. One epoch iterates N/B (here N is the number of samples, B - the batch size) times through the entire train set, until all the training samples go through the training process. The batch size can be equal to the entire train set size, be equal to one, or be somewhere in between. A larger batch size can result in faster training, but can also make the model more computationally expensive. Additionally small batch sizes add more regularization.

Learning rate defines how quickly a network updates its parameters. The used U-Net models use gradient descent, which is a first-order iterative optimization algorithm for finding a local minimum of a differentiable function (loss function). Selecting the appropriate value for this parameter can be challenging because a learning rate that is too low may cause the algorithm to take an excessive amount of time to reach the minimum, while a learning rate that is too high may cause the model to diverge as it may skip over the minimum.

Optimization algorithm. Optimization involves training a model iteratively to find the minimum value of the loss function. One common optimization technique is gradient descent, which uses the gradient of the objective function to navigate the search space and find the minimum value. It follows the negative gradient of an objective function in order to locate the minimum. One limitation of gradient descent is that a single step size (learning rate) is used for all input variables. Extensions to gradient descent like AdaGrad and RMSProp update the algorithm to use a separate step size for each input variable but may result in a step size that rapidly decreases to very small values. An additional extension, to the mentioned methods is the Adaptive Movement Estimation algorithm (Adam), which automatically adapts a learning rate for each input variable for the objective function and further smooths the search by using an exponentially decreasing moving average of the gradient to make updates.

Loss functions is an essential component of the optimization process, as it defines the goal that the model is trying to achieve. In simple terms, the loss function is a method of evaluating how well your algorithm is modeling the dataset. It is a mathematical function of the parameters of the machine learning algorithm. All the employed objective functions are listed and explained in Section 4.2.1.

3.2.5 Evaluation Metrics

Evaluating the quality of an algorithm requires a correct objective indicator. It is important to carefully consider which evaluation metric is most relevant for a given task, as different metrics may emphasize different aspects of model performance. The main evaluation metric for medical image segmentation is described below.

Dice-Sørensen Coefficient (often referred to as simply Dice) is an overlap-based method, which is used to calculate the similarity or overlap between two samples. It is the most frequently used metric for semantic segmentation model evaluation, the values of which range from 0 to 1. The closer the value is to 1, the better the segmentation effect. Given two image labels Y and $\hat{Y}$ (ground truth and predicted), the metric is defined as

$$Dice(Y, \hat{Y}) = 2\frac{|Y \cap \hat{Y}|}{|Y| + |\hat{Y}|}$$

In the experiments of this thesis, a few additional metrics have been calculated. First of all Intersection over Union (IoU), which is another popular choice for semantic segmentation, as well as precision and recall. Accuracy metric has not been included as due to the high class imbalance, it would not reflect the models' ability to segment tumors. All the three metrics are explained in detail in Section 3.3.2.

3.3 Object Detection

Object detection is an improved combination of image classification and object localization tasks, see Figure 11. Image classification involves predicting the class of one object in an image, while object localization refers to identifying the location of one or more objects in an image and drawing a bounding box around them. Object detection combines these two tasks which means it both localizes and classifies one or more objects inside an image.

The detection of lesions or objects of interest is a key part of medical diagnosis, which typically consists of localizing and classifying small areas, such as organs or lesions in the full image space [20]. The input in object detection tasks is an image usually with at least one object in it, the output is a file with bounding boxes that contain the objects coordinates, a class label for each object as well as the confidence score.

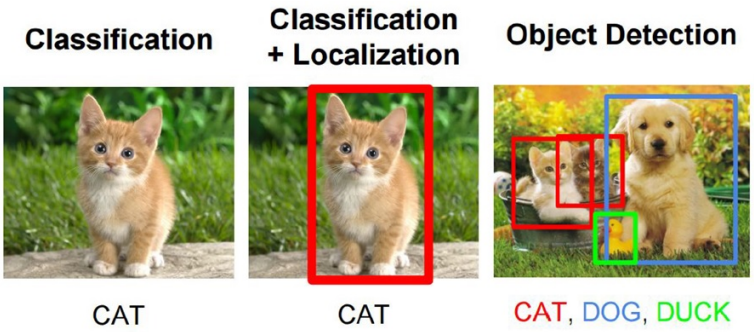


Figure 11: Overview of object recognition computer vision tasks, from [8]

DL-based object detection has become increasingly popular in recent years due to the availability of powerful computing hardware and large datasets [46]. These techniques have the ability to learn complex, high-level features from data, which allows them to accurately detect objects that may have subtle differences in appearance or be at different scales. In contrast, traditional object detection methods rely on hand-crafted feature engineering techniques and shallow trainable architectures. By

leveraging the capabilities of deep learning, object detection algorithms can overcome the limitations of traditional approaches and provide more accurate and robust results [46].

As mentioned in the literature review chapter, object detection algorithms can be split into two categories: one-stage detectors and two stage detectors. Two-stage algorithms follow the traditional object detection pipeline, generating region proposals at first and then classifying each proposal into different object categories. While one-stage detectors adopt a unified framework to make the predictions for object localization and classification at the same time. Since the experiments in this thesis will be using a one-stage object detection model YOLOv7 [42], further explanations will focus on the specific details of this type of approach.

YOLO Architecture

The original YOLO algorithm and most of its variations consist of three key components: the head, neck, and backbone. The backbone is the part of the network made of convolutional layers to detect key features of an image and process them. The neck connects the backbone and the head. It concatenates the feature maps from different layers of the backbone network and sends them as inputs to the head. The head processes the aggregated features and makes predictions on probabilities and bounding box coordinates. The YOLOv7 variation used in the experiments actually contains two heads. The head responsible for the final output is called the lead head, and the head used to assist training in the middle layers is named auxiliary head. All the described parts of the model work together to extract key visual features from the image then classify and bound them.

How It Works

First a YOLO model takes an input image and divides it into $S \times S$ grid cells (see the first image in Figure 12), which are used to detect and localize any objects they may contain. Each cell will predict B bounding boxes, relative to their cell coordinates, alongside a class label and confidence score for the potential object.

In YOLO models the object coordinates are normalised and therefore alongside the confidence score, the values lie within the interval $[0, 1]$. The confidence levels capture the model's certainty that there exists an object in that cell and that the bounding box is accurate. It is calculated by finding the IoU between the predicted bounding box and the ground truth bounding box. While each cell may predict any number of bounding boxes and confidence scores for those boxes, it only predicts one class.

The grid approach often leads to a significant overlap of predicted objects from the cumulative predictions of the grids. To handle this redundancy and reduce the predicted objects down to those of interest, the Non-Maximal Suppression (NMS) algorithm is used, to suppress all the bounding boxes with comparatively lower probability scores. Once the detector outputs a large number of bounding boxes with the desired probability scores, it removes the bounding boxes with the largest IoU and lower probability score. This step is then repeated until only the desired final bounding box remains. This process is visualised in Figure 12.

In order to be able to update our parameters, of course, a loss function is implemented. YOLO loss function can be divided into three parts: one responsible for finding the bounding-box coordinates, one for the bounding-box score prediction, and the class score prediction. The final loss consists of a weighted sum of all three components.

3.3.1 YOLOv7

YOLOv7 is the latest variation that outperforms all previous YOLO versions in terms of both speed and accuracy [6]. Compared to the previous versions, YOLOv7 has a faster and stronger network architecture that provides a more effective feature integration method, more accurate object detection

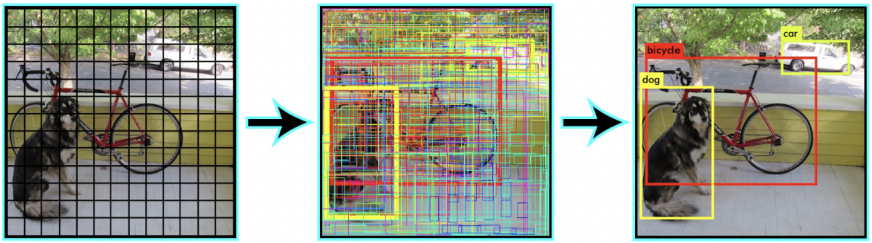


Figure 12: Non-Maximal Suppression algorithm visualization, from [5]

performance, a more robust loss function, and an increased label assignment and model training efficiency. It is also much more cost-effective to implement, requiring significantly cheaper hardware than other neural networks, and can be trained quickly on small datasets without the need for pre-trained weights. Considering this, it is expected that YOLOv7 will become the industry standard for object detection in the near future, surpassing the previous SOTA model, YOLO v4, in real-time applications [6]. This improved model efficiency is achieved by making a number of changes to both the YOLO network architecture and training routines.

On the architectural level, YOLOv7 integrates a computational block in the YOLOv7 backbone called Extended Efficient Layer Aggregation Network (E-ELAN). It enables the model to achieve the ability to continuously improve the learning ability of the network without destroying the original gradient path [42]. In addition, YOLOv7 scales its architecture by concatenating concatenating layers together. This allows the model to meet the needs of different inference speeds.

Besides the architectural improvements, YOLOv7 benefits from an updated trainable bag-of-freebies. The term refers to a set of modifications or enhancements that can be applied to the basic model's algorithm in order to improve its accuracy without increasing the training cost, hence they are called *freebies*. These modifications can help to improve the accuracy and robustness of the YOLO algorithm, and can be applied to a variety of different object detection tasks.

3.3.2 Evaluation Metrics

An important metric in object detection is IoU, which evaluates the overlap between two bounding boxes (definition is described in Image Segmentation Evaluation Metric section above). By applying the IoU and setting a threshold (usually set to 50%, 75% or 95%) the detections can be classified into the following groups:

True Positive (TP) is a correct detection, detected using IoU threshold.

False Positive (FP) is a wrong detection, detected using IoU $<$ threshold.

False Negative (FN) is a ground truth that was not detected.

True Negative (TN) would represent a corrected misdetection, not relevant in the tasks of this experiment.

These groups of detections make up the confusion matrix and are used to calculate key evaluation metrics in object detection, such as Average Precision (AP) and Mean Average Precision (mAP).

First of all the formula for precision is

$$P = \frac{TP}{TP + FP}$$

or

$$P = \frac{TP}{total\ all\ detections}$$

While recall is

$$R = \frac{TP}{TP + FN}$$

or

$$R = \frac{TP}{all\ ground\ truths}$$

AP is the area under the precision-recall curve. The precision-recall curve shows the tradeoff between precision and recall for different IoU thresholds. AP summarised the precision-recall curve into a single value representing the average of all precisions. The general definition for AP is finding the area under the precision-recall curve:

$$\int_{R=0}^{1} P(R)dR$$

The mAP is simply the average AP over all classes. It can be calculated as follows:

$$mAP = \frac{1}{C}\sum_{c=1}^{C} AP_c$$

where C is the number of classes.

It is common to look out for a good value of mAP50 metric, to evaluate object detection performance. The number 50 means that it is the mAP calculated at IoU threshold of 0.5. One reason it may be considered more important than other metrics is because it provides a more complete picture of the model's performance. A model with high precision but low recall may have a high mAP score if it is able to accurately detect a large number of the objects it predicts, even if it misses some. On the other hand, a model with low precision but high recall may have a low mAP if it generates a large number of false positive predictions.

4 Experiments

This chapter consists of four main parts: dataset introduction, one-stage approach, two-stage approach and result comparison. The first section introduces the used datasets. The second two sections first of all explain the experimentation process and then the results. And the final section compares the results among the different segmentation approaches.

The experiments were conducted using Google Colab Pro+ and high performance computing (hpc) resources, provided by the Information Technology Research Center of Vilnius University. Python was the used programming language, alongside Tensorflow library, for building the segmentation model. The code for the YOLOv7 detection model [42] and the inter-slice image generation [29] has been outsourced. The code, used in the experiments can be found in the following GitHub repository: https://github.com/kamidem/kidney-tumor-segmentation.

4.1 The Dataset

The following experiments were conducted primarily using kidney region CT scans from the National Cancer Institute of Lithuania. The initial dataset contained scans of 32 patients, however scans of one patient had to be dismissed, because of incorrect labeling. The final dataset of the 31 remaining patient came to a total of 6675 images (934 of which contained a tumor), with a maximum of one tumor per slice.

The CT images are originally 3D and were saved in DICOM (Digital Imaging and Communications in Medicine) format, which is a standardized format that stores medical images, such as X-rays, CT scans, and MRIs. DICOM images contain both the pixel data of the image and a set of metadata, such as patient information, image acquisition parameters, and image annotations. This allows DICOM images to be shared, stored, and processed by different medical systems and software applications, enabling data exchange between different medical institutions and practitioners.

As mentioned earlier, one of the most important applications of kidney tumor segmentation is the reconstruction of the tumor for accurate diagnosis and surgical treatment. Accurate reconstructions require working with 3D data, however training a model on 3D data requires a significantly larger dataset than methods for 2D segmentation. Although the accuracy often suffers, it is still possible to produce a 3D reconstruction of the segmented areas, employing 2D segmentation techniques from CT images, as the DICOM files contain information on slice location and thickness. Since the main dataset for the experiments of this thesis consists of very few cases, it was decided to use 2D semantic segmentation models. Therefore the images were converted to Portable Network Graphics (PNG) image format, with each image being of dimension 512×512. Image intensity values were normalized to fit into the interval $[0, 1]$.

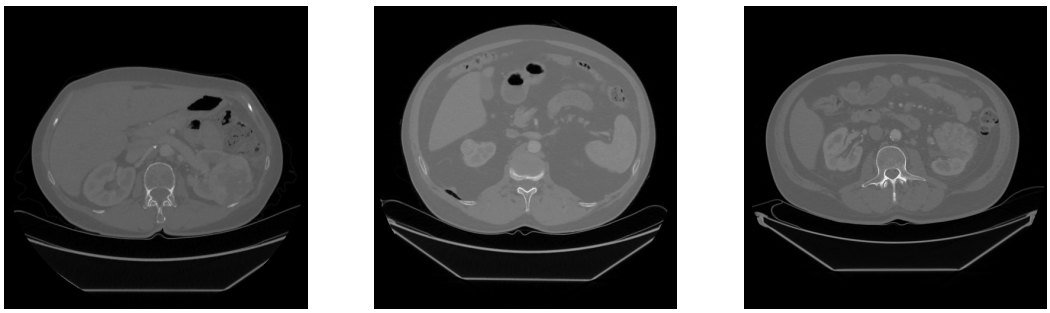
The original 2D CT images can be seen in Figure 13.

Data for Object Detection

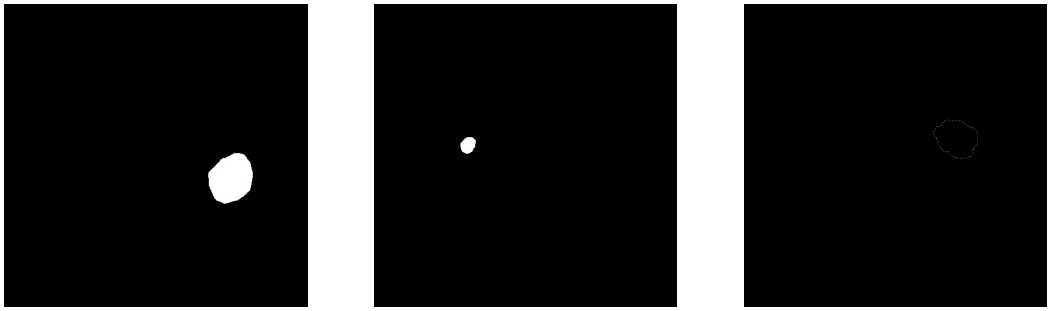
The semantic segmentation labels from the original dataset were converted to bounding boxes for the object detection task in the two-stage approach. The segmentation masks were converted to the format, used in YOLO models, $(class, x_c, y_c, w, h)$. Here x_c and y_c give the central x and y coordinate of the bounding box respectively, while w and h give the bounding box width and height. Both of the coordinates as well as the width and height values are normalized, to fit into the interval $[0, 1]$.

4.1.1 K-cross fold validation

Model evaluation is often performed with a hold-out split, which means that the majority of the data is used for training (usually 70% - 90%) and the rest for evaluating the model. However this is a very naïve approach, as it assumes that the data is representative across the splits and that there are no



(a) image 1 (b) image 2 (c) image 3



(d) mask 1 (e) mask 2 (f) mask 3

Figure 13: Examples of the images in the Kindey Tumour Dataset from the National Cancer Institute of Lithuania

redundant samples within the datasets. K-fold Cross Validation is a more robust evaluation technique. It splits the dataset in k-1 training batches and 1 testing batch across k folds. This approach allows to train the model for multiple times with different dataset splits, which increases the confidence in the model evaluation results, in comparison to the simple hold-out split technique. It is especially useful when dealing with data scarcity problem.

The kidney tumor dataset was split into 6 folds by patient ID. The number was chosen to ensure that more patients make it to the test set. The folds were split into training, testing and validation set, ensuring that each patient ends up exactly once in the test set. The training set is used to train the network model, validation set - to adjust the hyperparameters, and the test set - to evaluate the final performance of the model on unseen data.

As the original patient IDs are quite long, for the ease of view, in the upcoming tables and plots they were replaced by shorter versions p1-31 (patient 1 to patient 31). Assignment of patients to each fold is given in Table 1.

Fold	1	2	3	4	5	6
Patients	p1-5	p6-10	p11-15	p16-20	p21-25	p26-31

Table 1: Patients assigned to the test set in each fold.

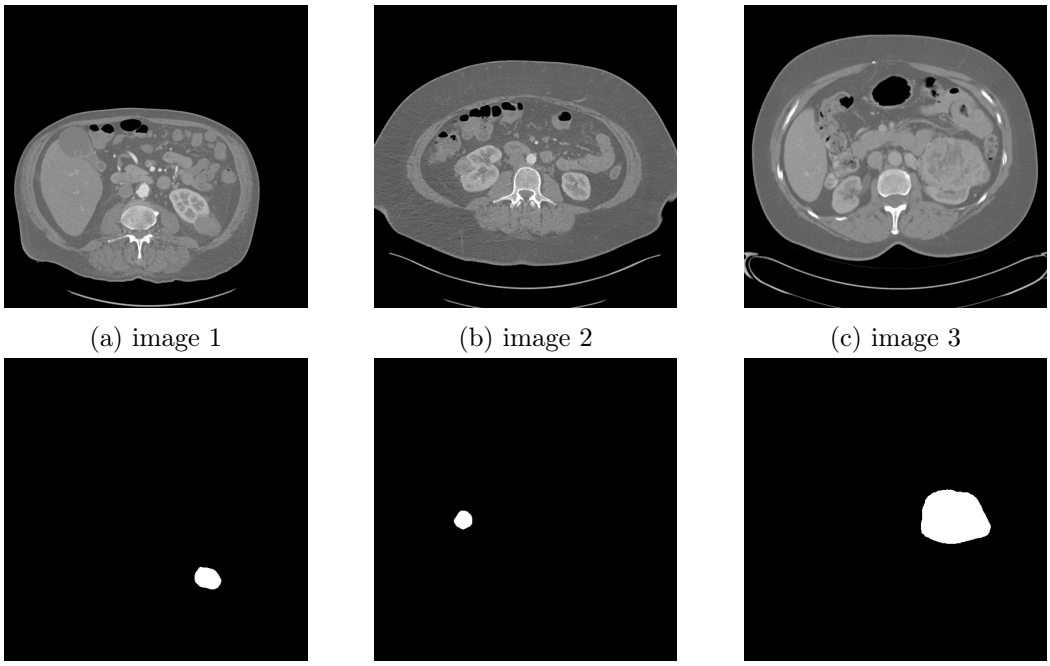
4.1.2 Additional Dataset

An additional dataset was outsourced from the 2021 Kidney and Kidney Tumor Segmentation Challenge (KiTS21) [19]. KiTS21 is the second edition of the competition in which teams compete to develop the best system for automatic semantic segmentation of renal tumors and the surrounding

anatomy. The dataset includes patients who underwent partial or radical nephrectomy for suspected renal malignancy between 2010 and 2020 in the US.

Each patient's contrast-enhanced preoperative scan was independently segmented three times for each instance of the following semantic classes: kidney, tumor, cyst. A total of 300 patient scans have been labelled. To avoid any human error in labeling, the dataset provides the average labels, which were averaged from the segmentations from three different labeling by different individuals. Only the averaged labels of kidney tumor were used in the experiment of this thesis. A few examples of images and masks are presented in Figure 14.

This dataset was used for fine-tuning, which, as mentioned before, is an effective approach to deal with labeled data scarcity. It can be an effective way to improve the performance of a model on a new task, especially when both of the tasks are related.



(a) image 1 (b) image 2 (c) image 3

(d) mask 1 (e) mask 2 (f) mask 3

Figure 14: Examples of the images from the KiTS21 dataset.

4.2 One-Stage Approach

The one-stage semantic segmentation approach consists of a single segmentation model. Being one of the most popularly used approaches in any semantic segmentation task today, a basic U-Net model was chosen. This model will stand as a baseline for performance comparison.

4.2.1 Training

The chosen U-Net model almost completely resembles the original [31], with only a few minor adjustments. First of all, batch normalization layers were added before each ReLU activation. It serves to speed up the training and allows the use of higher learning rates, making the learning process easier. It also stands as a regularization technique, to avoid overfitting. The second change to the original U-Net was the implementation of different loss functions, the process is described in detail below.

The experiments of one-stage approach can be split into the following categories:

1. Dataset Experiments

2. Objective Function Experiments

3. Fine-tuning Experiment

Dataset Experiments

With the aim of increasing the performance and generalization ability of the U-Net model, a number of the original train set variations have been tested. All the augmented datasets were generated offline. All of them are described in detail below.

Original. In this case, no adjustments were made to the train set, it was trained as it is.

Tumor region train set is a smaller version of the original that reduced the original train set by removing all instances where a tumor has not been found. This way the train set was reduced from 5047 to 692 images, which made the training significantly faster. By only leaving the image and mask pairs with a segmented instance of a tumor, an attempt is made to reduce the class imbalance, which is a common problem in tumor segmentation. Training a semantic segmentation model with class imbalance can be challenging as the model may become biased towards the more frequently occurring classes, leading to poor predictions for the underrepresented classes.

Tumor region + basic augmentation is an attempt at dealing with labeled data scarcity by generating a larger train set to improve model performance and generalizing ability. As the Tumor region performed significantly better than the Original, the following experimentations with data augmentation were performed only on the Tumor region train set variation. Many various basic augmentation techniques have been tested out, which included geometric transformations (rotation, translation, scaling), intensity transformations (adjusting brightness, contrast), noise injection (Gaussian noise) etc. The final augmentations consist of horizontal flip(with probability 0.2), rotation (with 15 degree limit, 0.5 probability), random size crop (with height remaining between 450 and 490, with 0.5 probability), random brightness contrast (with brightness limit 0.1, contrast limit 0.1, and 0.5 probability), hue saturation value shift (with hue shift limit 4, saturation shift limit 4, value shift limit 4 and 0.5 probability), CLAHE - enhanced local contrast (with upper threshold value for contrast limiting of 1, probability 0.5) and Gaussian noise (with probability 0.5). The applied augmentations doubled the train set size, which in the end was 1382. A few examples of the augmented images that were included in the best performing dataset can be observed in Figure 23.

Tumor region + basic augmentation ×2 train set included the same type of augmentations as the previous one, except that instead of doubling the train set size, it tripled it - with double the amount of augmented images. A general rule with data augmentation, is to use as much as possible without overfitting the model. The goal of testing with different amounts of augmentations, was to find the optimal amount for the dataset at hand.

Tumor region + inter-slice augmentation train set consisted of additionally generated inter-slice images (see Figure 24). There images were generated using an outsoursed code for frame interpolation neural network model called FILM: Frame Interpolation for Large Motion [29]. Inter-slice image generation was applied only to adjacent slices, which was checked before the generation by scan names. A total of 661 sliced were generated, which increased the train set to 1353 image and mask pairs.

Tumor region + basic augmentation + inter-slice augmentation train set is the combination of the Tumor region trains set and the two already described augmentations. The final train set size consisted of 2043 image and mask pairs.

Objective Function Experiments

One of the most important decisions that has to be made on a DL task is the choice of a loss function. Considering the reviewed papers [16] [23] and others of similar tasks, the U-Net model has been trained using the following objective functions: Dice Loss, Log-Cosh Dice Loss, Tversky Loss and Focal Tversky Loss.

Dice Loss is a region-based loss and it can directly optimize the Dice Similarity Coefficient (DSC) which is the most commonly used segmentation evaluation metric. There are two variants for Dice loss, with and without squared terms in the denominator. In the following experiments, the non-squared version is used and it is defined as

$$\mathcal{L}_{Dice} = 1 - \frac{2\sum_{i=1}^{N} y_i \ \hat{y}_i + b}{\sum_{i=1}^{N} y_i + \sum_{i=1}^{N} \hat{y}_i + b}$$

The equation includes an additional smooth factor b added to the numerator and denominator to avoid undefined edge cases when $y = \hat{y} = 0$, preventing division by zero and smoothing gradients of the loss function. This version, often called soft Dice, is a commonly used similarity metric for single-class segmentation tasks [41]. Overall, the smooth factor in the Dice Loss serves as a regularization term that can help to improve the stability and generalization of the model, and can be an important factor to consider when designing and evaluating models for tasks such as medical image segmentation.

Log-Cosh Dice Loss is a variation of Dice Loss that has been proved to bring improved semantic segmentation results [16]. This loss function is proposed for its tractable nature while encapsulating the features of dice coefficient. It is defined as

$$\mathcal{L}_{lc-Dice} = \log(\cosh(\mathcal{L}_{Dice}))$$

Tversky Loss achieves a better trade-off between precision and recall. It adapts the Dice loss to emphasize false negatives and is defined by

$$\mathcal{L}_{Tversky} = 1 - T(\alpha, \beta) = 1 - \left(\sum_{i=1}^{N} y_i \hat{y}_i\right) / \left(\sum_{i=1}^{N} y_i \hat{y}_i + \alpha \sum_{i=1}^{N} (1 - y_i)\, \hat{y}_i + \beta \sum_{i=1}^{N} y_i\, (1 - \hat{y}_i)\right)$$

where α and β are hyper-parameters that control the trade-off between false negatives and false positives.

It was designed to optimise segmentation on imbalanced medical datasets by utilising constants that can adjust how harshly different types of error are penalised in the loss function. To summarise, this loss function is weighted by the constants α and β that penalise false positives and false negatives respectively.

Focal Tversky Loss is a variant on the Tversky loss that also includes the gamma modifier from the Focal Loss. It applies the concept of focal loss to focus on hard cases with low probabilities and is defined by

$$\mathcal{L}_{FTL} = (\mathcal{L}_{Tversky})^{\frac{1}{\gamma}}$$

However, highly imbalanced segmentation works better with focus based loss functions. Similarly, binary-cross entropy works best with balanced data-sets, whereas mildly skewed data-sets can work around smoothed or generalized dice coefficient. [16]

Fine-tuning Experiment

Considering the reviewed literature, the full network adaption fine-tuning approach was implemented. First, having tuned the model's hyperparameters with keras tuner, the one-stage model was trained on KiTS21 dataset. Full network adaption was tested out with loading the saved model and fine-tuning it on the target data. A few different learning rates were implemented, to find the optimal. U-Net was trained for around 60 epochs on the KiTS21 dataset. And fine-tuned with the original dataset for up to 30 epochs.

Hyperparameters

For all of the training experiments the training stopping criteria was either 25 epochs or no improvement in validation set loss for 10 consecutive epochs. The optimal number of epochs was under 20. For assurance purposes, the most successful model was run for 50 epochs with a stopping condition of 15, to ensure the optimal performance was achieved. The optimal batch size was 4. For optimization, Adam optimizer was used, together with the initial learning rate of 0.0001.

The final consideration was the threshold value in the evaluation. The predicted mask values come in the range $[0, 1]$. To get the final tumor label output, the predicted mask has to be binarized using a chosen threshold. Pixels below the threshold are classified as the background, while pixels above the threshold are classified as tumors. The optimal value of 0.75 was chosen for the threshold.

4.2.2 Results

In order to save time, the following experiments were only conducted on a single fold and once the optimal was obtained, training, using the rest of the folds was conducted. To optimise this benchmark model, variations of the dataset as well as the objective loss have been considered. Results are presented below.

Dataset Experiments

The results of using different variations of the original dataset are presented in Table 2.

After experimenting with a variety of augmentations, their intensities as well as different amounts of augmented images, the dataset that achieved the highest model performance consisted of $1:1$ ratio of augmented and original images, with only basic augmentation applied. Although by itself, interslice agmentation did increased the model's performance, it did not outperform basic augmentations and did not increase model performance when added to the basic augmentations. It can be observed from Table 2 that the applied basic augmentation techniques showed over 10% of increase in Dice in both Original and Tumor region datasets.

Objective Function Experiments

The results of employing the aforementioned objective functions can be seen in Table 3.

The optimal objective function was Focal Tversky loss, with optimal hyperparameters $\alpha = 0.5$, $\beta = 1$, $\gamma = 2$. Using a larger β weighs recall higher than precision (by placing more emphasis on penalizing false negatives than false positives).

Dataset	Dice	IoU	Recall	Precision
Original	0.16064	0.22779	0.24005	0.57676
Original + basic augmentation	0.26779	0.35260	0.41638	0.64620
Tumor region	0.27357	0.33856	0.32132	**0.74686**
Tumor region + basic augmentation	**0.39369**	**0.48002**	0.58265	0.72659
Tumor region + inter-slice augmentation	0.32505	0.43407	**0.63560**	0.39799
Tumor region + basic augmentation + interslice augmentation	0.34977	0.45428	0.57013	0.52910

Table 2: Results of the one-stage segmentation U-Net model using different variations of the original dataset.

Loss Function	Dice	IoU	Recall	Precision
Dice	0.32090	0.38315	0.42711	0.75862
Log Cosh Dice	0.36412	0.42655	0.42858	0.81789
Tversky	0.38915	0.46268	0.50700	0.79023
Focal Tversky	**0.39369**	0.48002	0.58265	0.72659

Table 3: Results of the one-stage segmentation U-Net model employing different objective functions.

It is apparent that in all of the loss function cases, the model exhibits low recall relative to high precision. This means that the model is able to accurately identify positive samples, but it tends to miss a significant number of positive ones.

K-Cross Fold Validation

The evaluation results across all six folds can be seen in Table 4.

Fold	Dice	IoU	Recall	Precision
1	0.39369	0.48002	0.58265	0.72659
2	0.36363	0.45740	0.49756	0.54065
3	0.23712	0.28236	0.35312	0.46883
4	0.24978	0.31787	0.42441	0.31531
5	0.07245	0.10604	0.09807	0.45410
6	0.29402	0.37144	0.31406	0.75951

Table 4: Final one-stage approach results across all 6 folds.

As the Dice score varies significantly, the results suggest that with this small amount of data, it is difficult to build a model that could be generalizable. The results of each fold depend highly on the successful distribution of patients between train, validation and test sets. Because of such results, k-

cross fold validation was not implemented in the two-stage approach. All of the remaining results were obtained using the most successful fold, numbered 1. It is important to note, that the success of this fold is likely to be based on the fact that the model's hyperparameters were tuned using this specific fold. Hyperparameters highly depend on the data, since the dataset consists of very few patients, naturally the chosen hyperparameters might not be optimal for the rest of the folds.

Regardless of the fact that the produced models cannot generalize well, further experiments have been carried out, to find out whether the two-stage approach can achieve better performance using fold 1 as input.

Examples of the one-stage segmentation for each patient in fold 1 are displayed in Figure 25.

Fine-tuning Results

The chosen fine-tuning approach under-performed in comparison to the baseline model. Optimal results are presented in Table 5.

Dice	IoU	Recall	Precision
0.23451	0.35328	0.58237	0.34032

Table 5: Optimal results from fine tuning the one-stage approach.

These results were achieved after fine-tuning the pre-trained model for 22 epochs with the learning rate of 0.0001, as a smaller learning rate, as some literature suggested, did not seem to tune the model well for the new task.

Looking at the results of the baseline one-stage approach 2 3 4, it can be seen that in almost all of the experiments precision is significantly higher than recall. However in the case of this fine-tuned mode, recall is the higher metric. This indicates that the model is quite sensitive as it produces many more false positives.

4.2.3 Per Patient Results

The Dice score results for each patient in the test set can be observed in Figure 15, while full evaluation results can be found in Table 6.

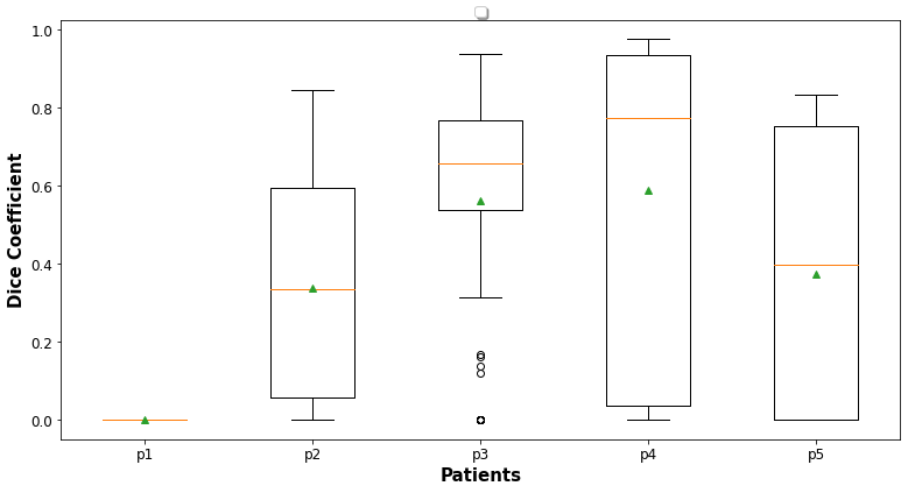


Figure 15: One-stage Dice score box plot results per patient. The green triangles indicate the mean Dice score across all the images of a single patient.

The highest level of segmentation accuracy was achieved on the p4 scans of the patient, with a Dice score of 0.590678. Examination of the ground truth labels for each patient in the test set reveals that this patient has one of the larger tumors in the dataset, as is evident by the large number of ground truth labels (56 slices) and the size of the tumor label on the axial slices. In contrast, the lowest level of segmentation accuracy was observed for the p1 patient, with a Dice score of 0, indicating that the tumor was not detected at all. Visual inspection of the remaining test patients suggests that p1 patient has the smallest tumor among the group.

Patient	**Dice**	**IoU**	**Recall**	**Precision**
p1	0	0	0	1
p2	0.338484	0.240036	0.486053	0.670360
p3	0.563112	0.441908	0.821892	0.485046
p4	0.590678	0.523460	0.573659	0.912262
p5	0.375406	.291819	0.456944	0.718508

Table 6: One-stage segmentation results per patient.

Examples of the segmentation for each patient are displayed in Figure 25.

4.3 Two-Stage Approach

The two-stage semantic segmentation approach, includes using two models, one for object detection and one for the segmentation of the detected areas. For the object detection task, the latest YOLOv7 model was implemented. And for the purposes of fair comparison, the detected images were segmented using the same U-Net architecture, used in the one-step approach.

The two-stage approach is not as straight forward as the previously discussed one. It requires additional processing steps in between the two models and additional processing for the segmented areas before evaluation. All the steps are described below.

Stage 1 An object detection model is trained. A small IoU threshold for the NMS algorithm is set, in order to produce more bounding boxes.

1.1 Bounding box sizes are unified to the same size and predictions are cropped according to the central coordinates of the bounding boxes. The cropped image names store the crop's coordinate information as well as the confidence score.

1.2 A train dataset that resembles the predicted cropped patches is generated. The images of all the train, validation and test sets are upscaled to pixel dimensions of 512×512.

Stage 2 The segmentation model is trained.

2.1 The test set from the first stage is used to generate predictions from the trained segmentation model.

2.2 An overlay algorithm takes the information from the filenames of the predicted test set labels and overlays all predicted patches into a single original size image (with dimensions 512×512). The algorithm considers the confidence score of each patch and the final threshold value binarizes the pixel values to generate the final prediction mask.

2.3 Segmentation evaluation.

4.3.1 Stage 1: Object Detection

For the first stage, an open-sourced code for YOLOv7 was used from the original paper authors [42]. For the training stage no adjustments were made to the model itself, the model was trained from scratch on the full original dataset. Some experimentation was performed with the hyperparameters and the data augmentations that are integrated into the model.

YOLOv7 has integrated *wandb*, which is an experiment tracking tool. It allowed to track how well the training was going in real time, showing the plots of the desired metrics, such as training and validation objective loss, box loss, , recall, precision etc. Additionally it displayed predicted bounding boxes on the input image after each epoch.

The optimal hyperparameters for YOLOv7 were the following: batch size 8, learning rate 0.001, the model was set to run for a total of 300 epochs, but has an early-stop after around 250 epochs as the desired metrics started declining. The best models for all five folds were achieved after around 200 epochs.

Once the detection model was trained, to maximize the chances of detecting the entire tumor, a few experiments were conducted to find the optimal IoU threshold value for the NMS algorithm before running predictions on the test set. This is supposed to result in an increase in the number of predicted bounding boxes. The optimal values were concluded to be 0.1 for the IoU threshold and 0.0009 for the confidence score threshold.

Step 1.1

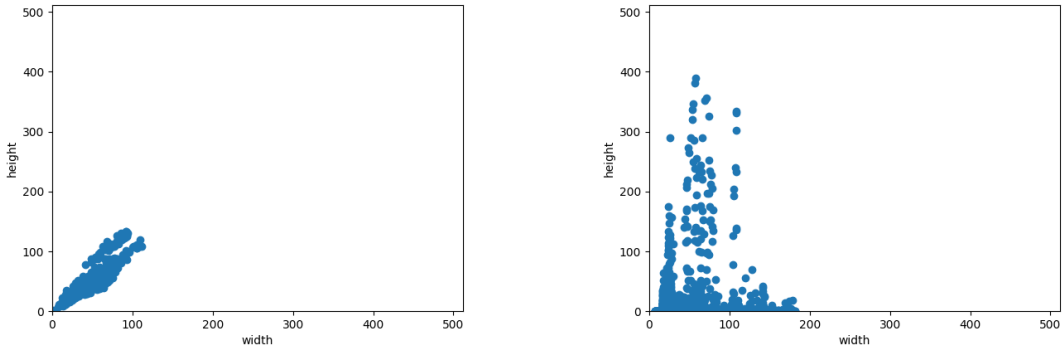
Since the segmentation model requires equal size input data, the train set images as well as all the detected patches had to be cropped to unified dimensions. If the chosen unified size is too large, this means that the segmentation model still deals with a relatively difficult task, having a larger region to segment and might produce more false predictions. If the crops are too small, the two step approach might fail to segment the entire size of large tumors. Therefore a few variations of image sizes have been experimented with to ensure the optimal size was chosen.

First the summary statistics for both the detected bounding boxes as well as the train set ground truth bounding boxes were calculated (see Table 7). Additionally scatter plots that show the distribution of different bounding box width and height dimensions were generated (see Figure 16). From these plots it can be seen that the predicted label dimensions are a lot more varied, especially in terms of the height dimension. In terms of the width dimension, the vast majority of the bounding boxes are plotted within the same region as in the ground truth plot, which indicates at least somewhat good performance of the model. The experimentation started testing with the maximum of the average ground truth dimension, creating crops of size 54×54. Additionally larger crop dimensions were tested, such as 64×64 and 128.

Bounding boxes	**Average width**	**Average height**	**Minimum width**	**Minimum height**	**Maximum width**	**Maximum height**
Ground Truth	51	54	1	1	111	133
Predicted	59	11	7	1	182	389

Table 7: Bounding box statistics, measurements in pixels.

Once the detected images were cropped, they were saved using the following name format: *patient_scan_prob_x_y.png*. Here *patient* indicates the name of the patient, *scan* - the number of the scan, *prob* - the confidence score of the bounding box, x - the upper left corner x coordinate of the cropped patch, y the upper left corner y coordinate of the cropped patch. This information will be



(a) Ground truth bounding boxes.

(b) Predicted bounding boxes.

Figure 16: Scatter plots displaying the distribution of bounding box width and height dimensions.

used in Step 2.2, when all the segmentation masks from the same images will have to be overlayed to create the final predicted mask for evaluation.

Step 1.2

Using the same unified crop size, a train set is generated for the segmentation model. A few variations of the optimal augmented train set have been tested:

1. First of all, to better model the possible output of YOLOv7, a few random train images were chosen for additional crop generation. Here the crops were shifted by a random number of pixels in the range [-10,10]. This was done to replicate the various bounding boxes that might include a tumor, but are not optimal as they were not excluded by the NMS algorithm. It is also helpful in the cases where the detected tumors might not be in the exact center for the crop.

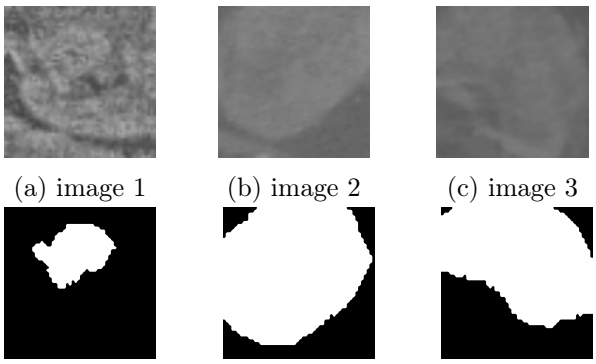
2. And secondly, to deal with the labeled data scarcity further, the segmentation was also tested with an augmented dataset, which included the same basic augmentations as in the one-stage approach (see a few examples in Figure 17). However in this case, additional crops shifted by a number of pixels in the range [-10,10], were not produced, as this could potentially produce four different versions of the same area and might therefore lead to overfiting. A few examples of the training set images with an added augmentation, can be observed in Figure 17.

4.3.2 Stage 2: Semantic Segmentation

First of all, since the segmentation model requires equal size input data, the cropped data splits were upscaled to dimensions 512×512. In addition to the experimentation with data augmentation, a few variations of train set and test set cropped image sizes was experimented with. And also, similarly as in the one-stage approach, the same experimentations with the objective function were carried out.

The training stopping criteria was once again either 50 epochs or no improvement in validation set loss for 10 consecutive epochs. The optimal number of epochs was around 10. The optimal batch size was 8. For optimization, Adam optimizer was used, together with the initial learning rate of 0.0001.

Finally, shallower versions of the U-Net model were tested. A shallow U-Net is a variant of the U-Net architecture that has fewer layers. As a result, it has fewer parameters and requires less computation to train and run, making it potentially faster and more efficient than the original model. However, it may also be less effective at extracting complex features and may produce less accurate segmentation results, particularly on tasks with highly varied or complex data. Since the cropped data is less complex



(a) image 1 (b) image 2 (c) image 3

(d) mask 1 (e) mask 2 (f) mask 3

Figure 17: Example images and masks from an augmented and cropped train set for U-Net training in the 2-Step approach. The displayed images are of 54 pixel length and with

than the full CT images, two shallow U-Net versions have been tested. One with a single block removed (U-Net 512) from both the encoder and decoder and another with two blocks removed (U-Net 256).

Step 2.1

Once the U-Net model has been trained, the detected patches are also upscaled to the image dimensions 512×512 and are run through the model to get a mask prediction. Here various threshold values have been tested, the optimal value of 0.9 was chosen. The predicted masks are saved with the same file name that contains the coordinates and confidence score information.

Step 2.2

The predicted masks are run through an overlay algorithm. An overlay algorithm first downscales the patches to the original size. Then it generates a blank image (consisting of only zero pixel values, which makes it completely black) of size 512×512 and using the coordinates in the name of the crop, sums the pixel values of each detected patch to it, multiplied by the confidence score of that detected area. This process is repeated for all the detected patches, any overlapping areas will have an added pixel value score, increasing their likelihood of being included in the final predicted mask after it passes a final threshold. The optimal value for the final threshold was found to be 0.001. Once the overlay process is finished, a final threshold value binarizes the pixel values to generate the final prediction mask. This process is visualised in Figure 18.

Step 2.3

The final predicted masks are run through the trained model for the final two-stage method evaluation. The final consideration was the threshold value in the evaluation step. The predicted mask values come in the range $[0, 1]$. To get the final tumor label output, the predicted mask once again has to be binarized using a chosen threshold. The optimal value of 0.5 was chosen.

Fine-tuning Experiment

The same full network adaption fine-tuning approach was implemented in the two-stage case. First of all the data splits, used in the one-stage task were converted into the necessary format for the two-stage approach. Then, after finding the optimal hyperparameters, both of the models were pre-trained and finally fine-tuned using the original dataset. Once again, few different learning rates were

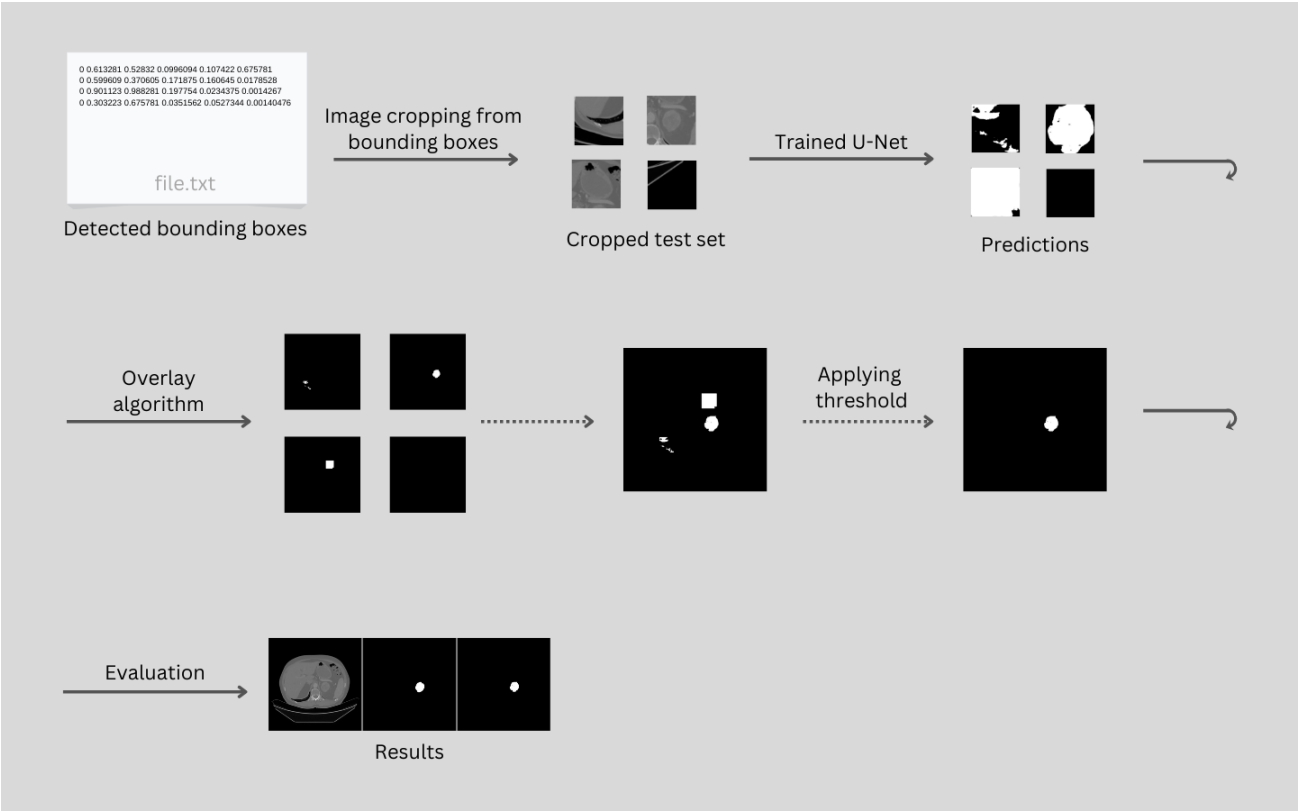


Figure 18: First the detected images are cropped using a chosen unified crop size and the coordinates from the bounding boxes. Then the detected patches are taken through the trained U-Net model, which generates predicted masks. The filenames of the predicted masks contain crop location and confidence score information, which is used in the overlay algorithm. In this algorithm each pixel value of the predicted patches is multiplied with the probability of that mask, then a threshold is applied and a final mask is produced. The final predicted mask is then used for model evaluation, which produces evaluation metrics and images that compare the original CT image, ground truth mask and the predicted mask.

implemented, to find the optimal. Since the U-Net in the one-stage approach did not benefit from fine-tuning, the two-stage approach additionally implemented only fine-tuning the detection model.

4.3.3 Results

Since in the one-stage approach, k-fold cross validation did not work, the two-stage approach was performed only on fold 1, for comparison purposed.

The results for the optimal object detection model are presented in Table 8.

Data split	**mAP50**	**mAP**	**Precision**	**Recall**
Test	0.343	0.226	0.497	0.397
Validation	0.404	0.291	0.653	0.354

Table 8: Results of the optimal YOLOv7 model. The displayed validation set metrics are taken from the epoch where the optimal model, used on the test set, was obtained.

$mAP50^{val}$ of 0.404 for such a small dataset is quite good, especially considering the $mAP50^{val}$ of 0.697, which was achieved by the COCO dataset (consisting of 328,000 images, having 80 categories),

as reported in the original paper [42]. Having chosen the right IoU threshold for the NMS algorithm as well as confidence score threshold ensured a good amount of the the right bounding boxes were selected at inference. In the end 984 labels were produced, with a total of 2595 bounding boxes. Although there are quite a few incorrect detections, the segmentation algorithm will fix many of these mistakes.

Regarding the optimal dataset for the U-Net model, after experimenting with various crop sizes for both training and detection sets, size of 64×64 in the end brought the best results. The dataset, that achieved the best results was the one containing the same type of augmentations as in the one-stage approach (Tumor region + basic augmentations).

As the segmentation approach consist of many different stages, that have not been combines into a single process, the optimal values of the dataset, objective function and the remaining hyperparameters had to be found through manual search. The results of each training, that explains the choice of the final hyperparameters is displayed in Table 13.

The optimal results of the two-stage approach are presented in Table 9. And examples of segmentation results of each patient in the test set can be observed in Figure 26.

Dice	IoU	Recall	Precision
0.40249	0.31170	0.47326	0.58168

Table 9: Optimal results from the two-stage approach.

4.3.4 Per Patient Results

The Dice score results for each patient in the test set can be observed in Figure 19, while full evaluation results can be found in Table 10.

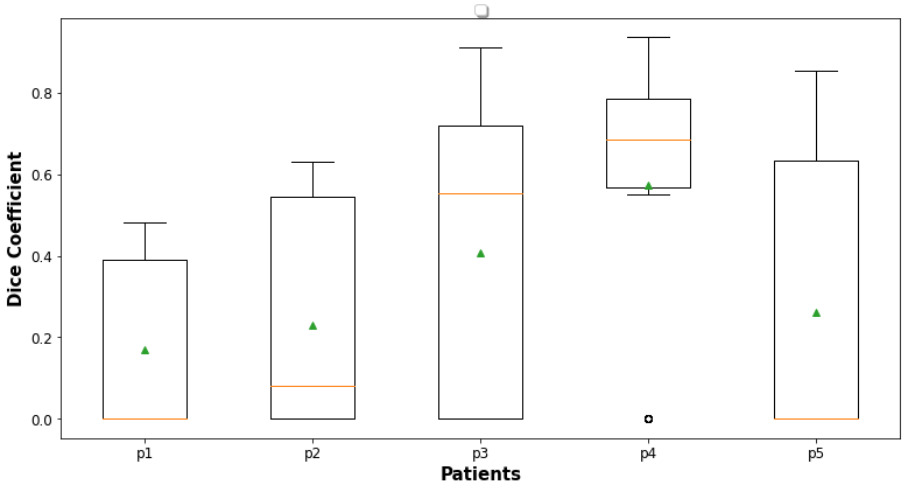


Figure 19: One-stage Dice score box plot results per patient. The green triangles indicate the mean Dice score across all the images of a single patient.

Same as in the one-stage approach, the best segmentation was achieved on patient's p4 scans, achieving 0.573628 Dice score. And similarly as in the one-stage approach, patient p1 had the worst segmentation results, however in the case of this method, the model managed to segment at least a bit of the existing tumor.

Patient	Dice	IoU	Recall	Precision
p1	0.169194	0.107580	0.391857	0.408271
p2	0.229559	0.156147	0.497076	0.323017
p3	0.406694	0.315192	0.423142	0.581142
p4	0.573628	0.459416	0.602057	0.776382
p5	0.261451	0.195003	0.322435	0.444586

Table 10: Full Two-Stage per patient results.

Fine-Tuning Results

The detection model produced 990 labels with a total of 3077 bounding boxes. The number of the produced bounding boxes increased by almost 500, in comparison to the original two-stage approach.

Similarly as in the one-stage approach, fine-tuning both the detection and the segmentation models gave worse model performance than the initial approach. However, better results were achieved, when fine-tuning only the detection model.

The optimal results of the fine-tuned two-stage approach are presented in Table 11. Examples of segmentation results of each patient in the test set can be observed in Figure 27.

Dice	IoU	Recall	Precision
0.41661	0.32970	0.51975	0.62092

Table 11: Optimal results from the fine-tuned two-stage approach.

4.4 Result Comparison

The best results for one-stage, two-stage and fine-tuned two-stage methods are presented in Table 12.

Model	Dice	IoU	Recall	Precision
One-Stage	0.39369	0.48002	0.58265	0.72659
Two-Stage	0.40249	0.31170	0.47326	0.58168
Fine-Tuned Two-Stage	**0.41661**	0.32970	0.51975	0.62092

Table 12: Summary of One-Stage, Two-Stage and Fine-Tuned Two-Stage approach evaluation metrics.

Considering the main metric - Dice score, the two-stage approach outperformed one-stage in both original and the fine-tuned model cases. Considering the rest of the metrics, one-stage approach had the advantage. Two-stage approach had an improved Dice score of nearly 1%. Fine-tuning of the two-stage approach detection model improved the Dice score even further by an additional 1.4%. Looking at individual patient results provides further information about the capabilities of each model.

Figure 20 displays achieved Dice score for each patient in all three approaches. It can be observed that in cases p2-4 one-stage approach outperformed both of the two-stage approaches. t is clear that the improvement in overall Dice score of the two-stage approach comes from the ability to detect extra

small tumor in the case of patient p1. This capability is increased more than twice by the fine-tuned version of the model. In the case of p5, one-stage approach outperforms two-stage approach, however the fine-tuned version has the best Dice score.

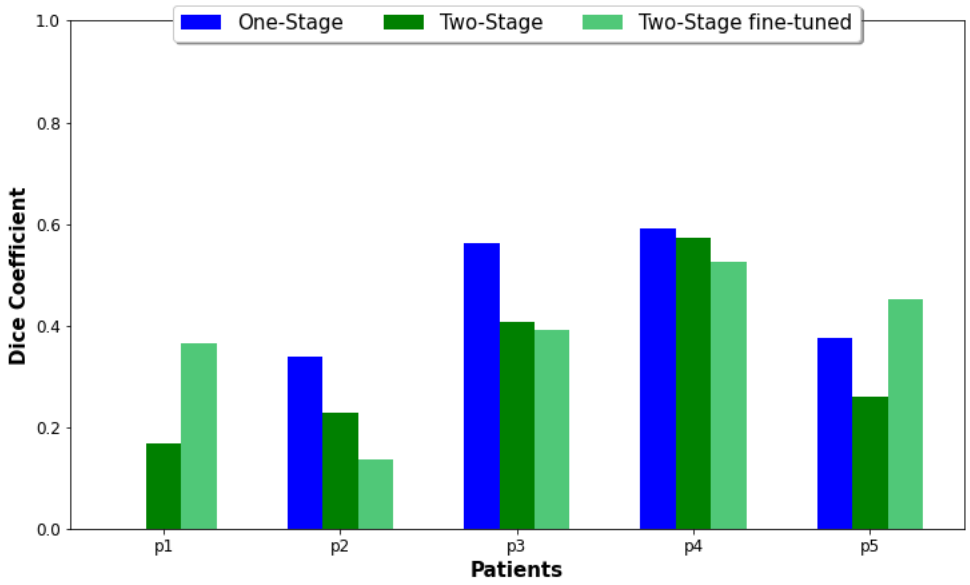
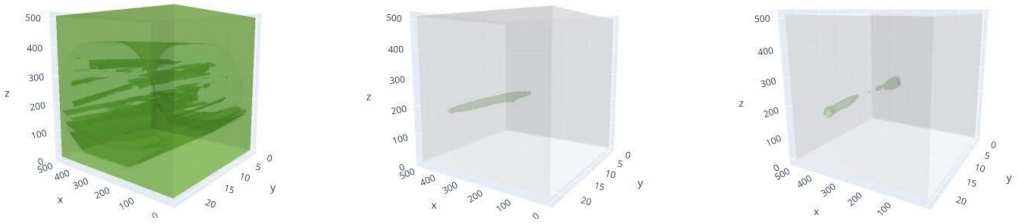


Figure 20: Per patient Dice coefficient result comparison.

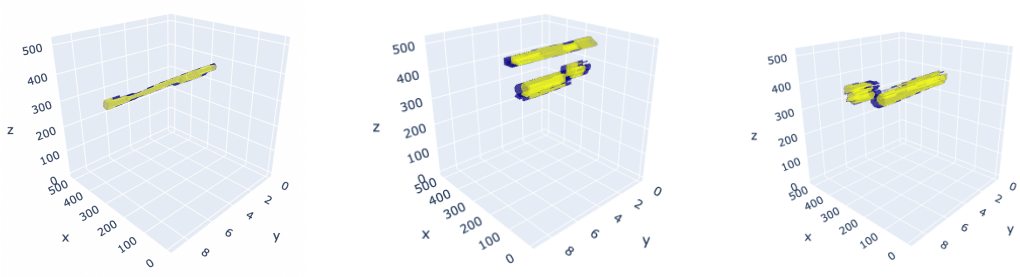
For better understanding of the segmentation of all three methods, 3D visualization of the segmented tumor has been generated. Figure 21 shows the comparison between the mask segmented by one-stage approach and the ground truth. It can be seen, that although not all the slices have been segmented, the model managed to capture the approximate size of the tumor well, and of course the location.



(a) CT scan (b) Ground truth label (c) Predicted label

Figure 21: A recreated 3D visualization of p2, from labels generated using the one-stage approach.

Figure 22 displayed the 3D representation of p1 segmentation by two-stage and fine-tuned two-stage approaches, which was completely missed by the baseline one-stage method. A significant improvement can be observed between the two-stage and the fine-tuned version, with far less false positives. Although the location of the tumor is mostly captured well in image (c), the size of it is still overcompensated. All the representation show that the predictions at least roughly reflect the ground truth. Although some labels have better segmentation, than others, the predictions themselves are not random.



(a) Ground truth label

(b) Predicted label (two stage)

(c) Predicted label (fine-tuned two stage)

Figure 22: A recreated 3D visualization of p1, from labels generated using both versions of the two-stage approach.

4.4.1 Computational Costs

Inference time for the two-stage and fine-tuned two-stage approaches was around three times longer than for the one-stage approach, taking around 60 and 20 minutes respectively.

Results and Conclusions

Results

1. K-cross fold validation approach returned results of high Dice score variation across all folds (difference of up to 0.32 in Dice score), see Table 4.

2. Experiments conducted with different versions of the original dataset and various data augmentation techniques concluded the best results to come from using a train set, which only included slices with a labeled tumor and doubling the train set size, by additionally applying basic augmentation techniques, see Table 2. The model performance improvement of up to 0.1 in Dice score was achieved.

3. Fine-tuning reduced the performance of the U-Net in both of the approaches. However a 1.4% increase in Dice score was achieved in the two-stage approach, when only the object detection model was fine-tuned.

4. Considering the final Dice score results across all the patients, the proposed two-stage approach outperformed one-stage by 1%, with an additional increase in 1.4% by the fine-tuned two-stage segmentation method, see Table 12. Looking at the model performance for each test set patient individually, one-stage approach has the advantage in most patient cases. However, it was unable to segment a small size tumor, which was where the two-stage approach gained its advantage, increased further by the fine-tuned model.

5. The proposed two-stage model evaluation time is around 60 minutes, while the baseline one-stage approach required only 20 minutes.

Conclusions

1. As the Dice score varies significantly, the results suggest that with this small amount of data (31 patient), it is difficult to build a model that could be generalizable. The results of each fold depend highly on the successful distribution of patients between train, validation and test sets.

2. The implemented experiments demonstrate that augmentation techniques must be implemented with caution and adjusted specifically for the dataset and task at hand.

3. Fine-tuning has been reported not to always bring improved results, which could be due to the variability in CT scanners, protocols, or perhaps, the use of contrast agents or even data distribution. However, given the comparable nature of the tasks, it is unexpected to observe such a drastic decrease in the Dice score. A more appropriate conclusion would be to state that the amount of fine-tuning-related experimentation, carried out for the purpose of this thesis, was insignificant. It is fair to say that fine-tuning should receive further application investigation for this specific task.

4. One-stage model may be more successful at capturing larger scale objects, however it may miss smaller ones. The two-stage approach, on the other hand, is less successful at capturing larger tumors, but manages to segment small sized tumor slices, missed by the baseline model. Accurate small tumor segmentation is of key importance, as this indicates earlier stages of the cancer, when more treatment options are available. The results of the proposed two-stage approach have shown potential and should be further investigated for possible application in the clinical setting.

5. Considering the increased complexity of the two-stage approach, the required time for inference is understandable. However, the two-stage approach, presented in this thesis, could undoubtedly benefit from further implementation optimization, which could make the process simpler and faster.

Discussion

Having carried out the experiments on such a small dataset, the conclusions should be considered with caution. Further experiments, addressing the scarcity of labeled data, should be carried out, to confirm the validity of the results. Having said that, the discovery of the two-stage segmentation approach's increased ability to segment small objects comes in line with another study [39]. This suggests that the proposed method for two-stage segmentation has potential for application and should be investigated and optimized further.

Additional investigation is required, to have a better understanding of the upcomings and failings of the suggested two-stage approach. Using a YOLOv7 model for detection requires all the detected areas to be cropped to a unified size, so they could be segmented by the U-Net. The produced labels often do not resemble a clear shape of the tumor, which is much more accurate when using the one-stage method. This would be a big problem for correct stage diagnosis and assigning the right treatment. Although the experiments of this thesis, showed the chosen crop size to bring the optimal final results, further investigation could shed some more light on this specific aspect on the algorithm and potentially increase the segmentation ability for larger tumors.

One-stage approaches have the advantage of being fast and efficient, but may struggle to capture the fine details concerning small tumors. While two-stage approaches can better capture these details, but may be slower and more complex, additionally having slightly lower segmentation ability for large tumors. Ultimately, for the moment a combination of one-stage and two-stage methods may be the most effective approach, depending on the specific needs of the medical professionals. Further research is needed to identify the best strategies for integrating the two-stage approach using a complex object detection model, and perhaps more advanced segmentation architecture and to develop more robust and generalizable methods for working with labelled data scarcity.

Acknowledgements

I would like to express my sincere gratitude to Dr. Linas Petkevičius, for his time and effort guiding and supporting me throughout the process of writing this master thesis. I would also like to thank Dr. Jonas Venius and the National Cancer Institute of Lithuania for providing the kidney tumor dataset and for the helpful consultations. And finally, I would like to thank the Information Technology Research Center of Vilnius University for providing high performance computing resources, which greatly aided in my research.

References

[1] Stages of kidney cancer treatment. https://www.uptodate.com/contents/epidemiology-pathology-and-pathogenesis-of-renal-cell-carcinoma/print, 2022.

[2] Mina Amiri, Rupert Brooks, Bahareh Behboodi, and Hassan Rivaz. Two-stage ultrasound image segmentation using u-net and test time augmentation. *International journal of computer assisted radiology and surgery*, 15(6):981–988, 2020.

[3] Mina Amiri, Rupert Brooks, and Hassan Rivaz. Fine-tuning u-net for ultrasound image segmentation: different layers, different outcomes. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 67(12):2510–2518, 2020.

[4] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[5] Hmrishav Bandyopadhyay. Yolo: Real-time object detection explained. https://www.v7labs.com/blog/yolo-object-detection, 2023.

[6] Gaudenz Boesch. Yolov7: The most powerful object detection algorithm (2022 guide). https://viso.ai/deep-learning/yolov7-guide/, 2022.

[7] Phillip Chlap, Hang Min, Nym Vandenberg, Jason Dowling, Lois Holloway, and Annette Haworth. A review of medical image data augmentation techniques for deep learning applications. *Journal of Medical Imaging and Radiation Oncology*, 65(5):545–563, 2021.

[8] Cognethi. Evolution of object detection networks.

[9] The Lindner Center for Research, The Christ Hospital Physicians Ohio Heart Education The Christ Hospital, and OH USA Vascular Center, Cincinnati. Basic principles in computed tomography (ct).

[10] S Brandon Hancock and Christos S Georgiades. Kidney cancer. *The Cancer Journal*, 22(6):387–392, 2016.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Mohammad Hesam Hesamian, Wenjing Jia, Xiangjian He, and Paul Kennedy. Deep learning techniques for medical image segmentation: achievements and challenges. *Journal of digital imaging*, 32(4):582–596, 2019.

[13] National Health Institute. Kidney cancer treatment. https://www.nhs.uk/conditions/kidney-cancer/treatment/, 2019.

[14] World Cancer Research Fund International. Kidney cancer data. https://www.wcrf.org/cancer-trends/kidney-cancer-statistics/, 2022.

[15] World Cancer Research Fund International. Worldwide cancer data. https://www.wcrf.org/cancer-trends/worldwide-cancer-data/, 2022.

[16] Shruti Jadon. A survey of loss functions for semantic segmentation. In *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–7. IEEE, 2020.

[17] Davood Karimi, Simon K Warfield, and Ali Gholipour. Transfer learning in medical image segmentation: New insights from analysis of the dynamics of model parameters and learned representations. *Artificial Intelligence in Medicine*, 116:102078, 2021.

[18] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019.

[19] KiTS21. The 2021 kidney and kidney tumor segmentation challenge. https://kits21.kits-challenge.org/, 2021.

[20] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[21] Xiangbin Liu, Liping Song, Shuai Liu, and Yudong Zhang. A review of deep-learning-based medical image segmentation methods. *Sustainability*, 13(3):1224, 2021.

[22] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[23] Jun Ma, Jianan Chen, Matthew Ng, Rui Huang, Yu Li, Chen Li, Xiaoping Yang, and Anne L Martel. Loss odyssey in medical image segmentation. *Medical Image Analysis*, 71:102035, 2021.

[24] Toni K. Choueiri Michael B. Atkins, Ziad Bakouny. Epidemiology, pathology, and pathogenesis of renal cell carcinoma. https://www.urolife.in/stages-of-kidney-cancer-treatment/, 2020.

[25] National Institute of Biomedical Imaging and Bioengineering. Computed tomography (ct). https://www.nibib.nih.gov/science-education/science-topics/computed-tomography-ct, 2022.

[26] National Institute of Diabetes, Digestive, and National Institutes of Health. Kidney Diseases. Kidney and nephron.

[27] Brian Owens. Kidney cancer. *Nature*, 537(7620):S97–S97, 2016.

[28] Prabod Rathnayaka, Vinoj Jayasundara, Rashmika Nawaratne, Daswin De Silva, Weranja Ranasinghe, and Damminda Alahakoon. Kidney tumor detection using attention based u-net. 2019.

[29] Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. Tensorflow 2 implementation of "film: Frame interpolation for large motion". https://github.com/google-research/frame-interpolation, 2022.

[30] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[32] Chuen-Kai Shie, Chung-Hisang Chuang, Chun-Nan Chou, Meng-Hsi Wu, and Edward Y Chang. Transfer representation learning for medical image analysis. In *2015 37th annual international conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 711–714. IEEE, 2015.

[33] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[35] Andrzej Skalski, Jacek Jakubowski, and Tomasz Drewniak. Kidney tumor segmentation and detection on computed tomography data. In *2016 IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 238–242. IEEE, 2016.

[36] Cancer Research UK Stephanie McClellan. Kidney cancer rates are increasing, so what's fuelling the surge? https://news.cancerresearchuk.org/2017/04/24/kidney-cancer-rates-are-increasing-so-whats-fuelling-the-surge/, 2017.

[37] ECIS European Cancer Information System. Long term estimates of cancer incidence and mortality.

[38] Satoshi Takahashi, Masamichi Takahashi, Manabu Kinoshita, Mototaka Miyake, Risa Kawaguchi, Naoki Shinojima, Akitake Mukasa, Kuniaki Saito, Motoo Nagane, Ryohei Otani, et al. Fine-tuning approach for segmentation of gliomas in brain magnetic resonance images with a machine learning method to normalize image differences among facilities. *Cancers*, 13(6):1415, 2021.

[39] Wei Tang, Dongsheng Zou, Su Yang, Jing Shi, Jingpei Dan, and Guowu Song. A two-stage approach for automatic liver segmentation with faster r-cnn and deeplab. *Neural Computing and Applications*, 32(11):6769–6778, 2020.

[40] Neethu Rose Thomas and J Anitha. An automated kidney tumour detection technique from computer tomography images. In *2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS)*, pages 1–6. IEEE, 2022.

[41] Chengjia Wang, Tom MacGillivray, Gillian Macnaught, Guang Yang, and David Newby. A two-stage 3d unet framework for multi-class segmentation on full resolution image. *arXiv preprint arXiv:1804.04341*, 2018.

[42] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*, 2022.

[43] World Health Organization (WHO). Cancer. https://www.who.int/news-room/fact-sheets/detail/cancer, 2022.

[44] Youzi Xiao, Zhiqiang Tian, Jiachen Yu, Yinshu Zhang, Shuai Liu, Shaoyi Du, and Xuguang Lan. A review of object detection based on deep learning. *Multimedia Tools and Applications*, 79(33):23729–23791, 2020.

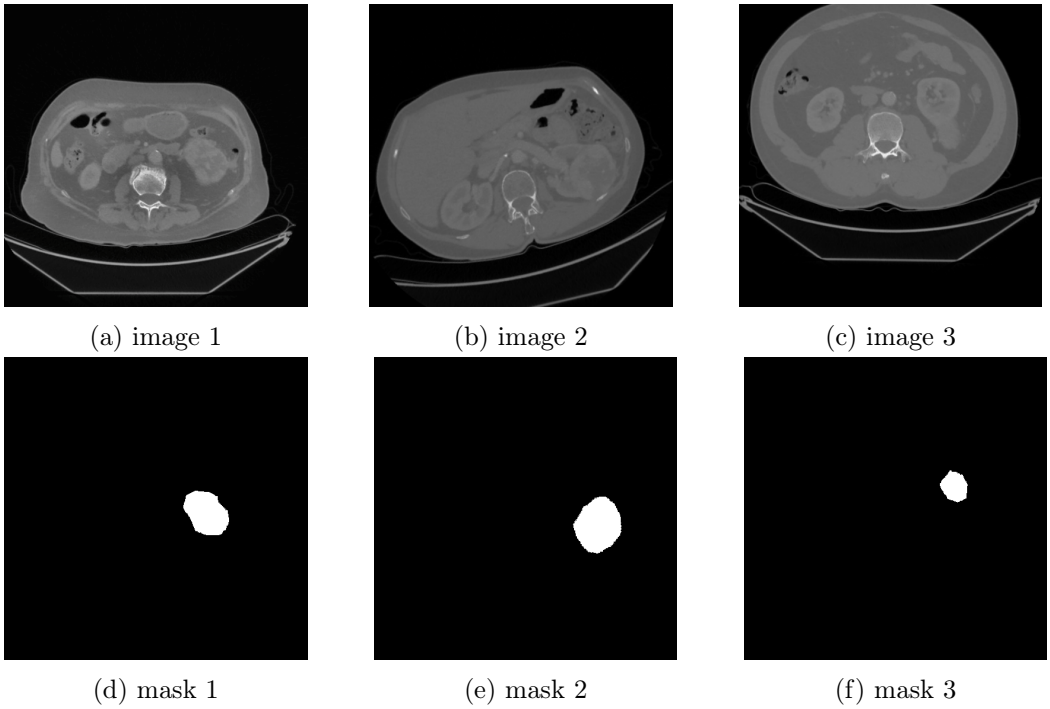
[45] Yu-Jin Zhang. Image segmentation in the last 40 years. In *Encyclopedia of Information Science and Technology, Second Edition*, pages 1818–1823. IGI Global, 2009.

[46] Zhong-Qiu Zhao, Peng Zheng, Shou-Tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11):3212–3232, 2019.

[47] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 3–11. Springer, 2018.

[48] Riaan Zoetmulder, Efstratios Gavves, Matthan Caan, and Henk Marquering. Domain-and task-specific transfer learning for medical segmentation tasks. *Computer Methods and Programs in Biomedicine*, 214:106539, 2022.

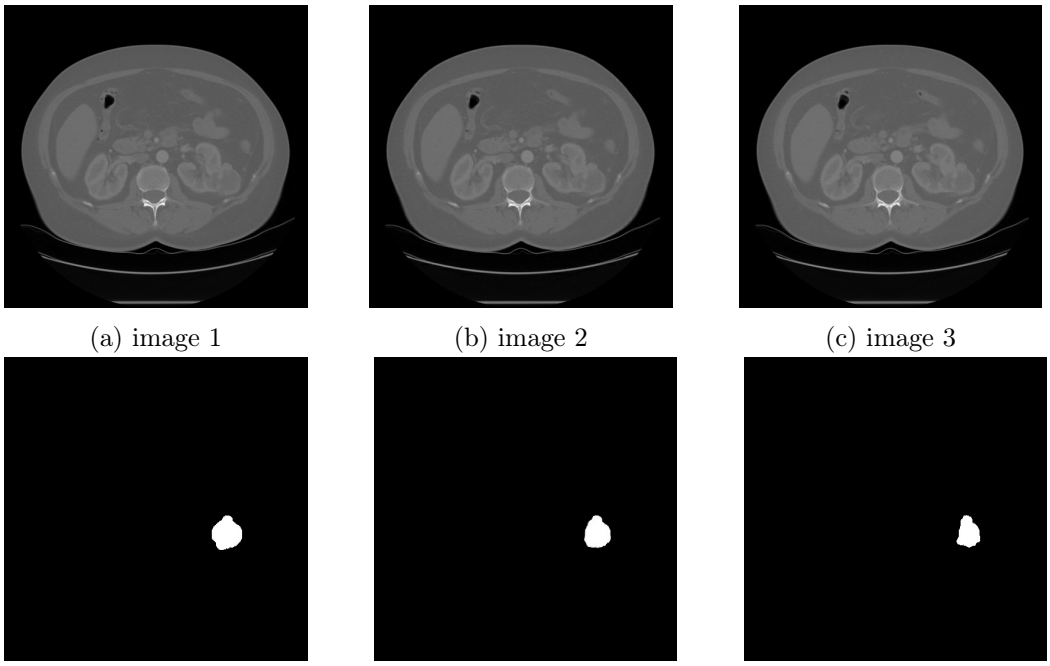
Appendix A



(a) image 1 (b) image 2 (c) image 3

(d) mask 1 (e) mask 2 (f) mask 3

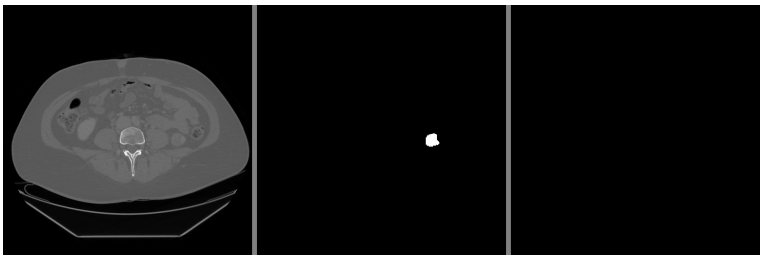
Figure 23: Examples of augmented images from the updated training set.



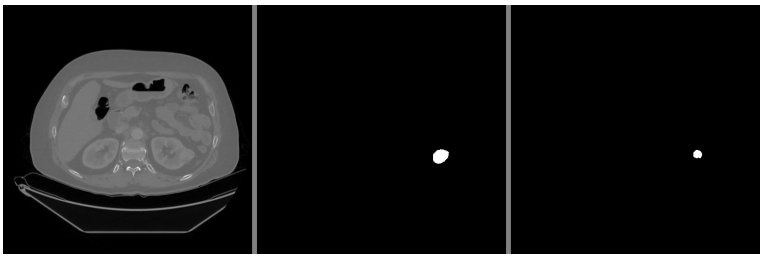
(a) image 1 (b) image 2 (c) image 3

(d) mask 1 (e) mask 2 (f) mask 3

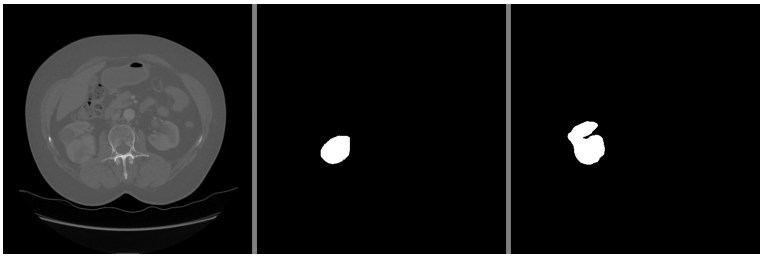
Figure 24: Example of inter-slice augmentation. Image and mask 1 and 3 are from the original dataset, while the image and mask in the middle (number 2) is a generated inter-slice.



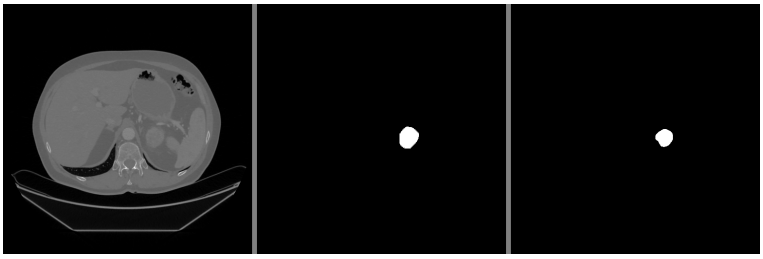
(a) p1, slice 9



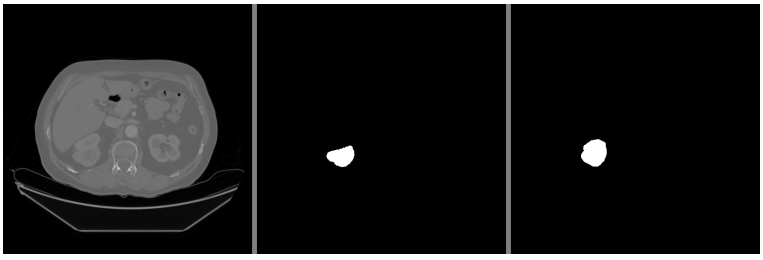
(b) p2, slice 120



(c) p3, slice 90

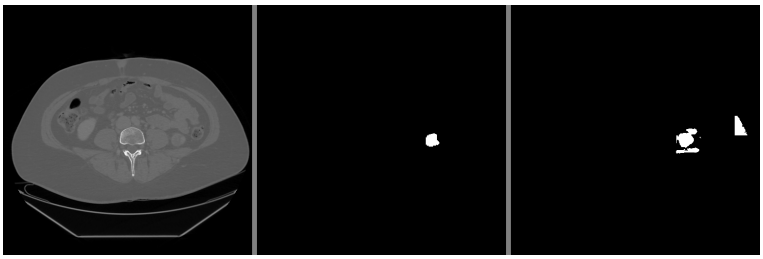


(d) p4, slice 166

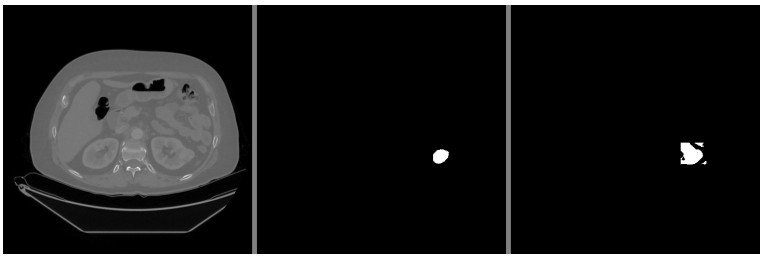


(e) p5, slice 115

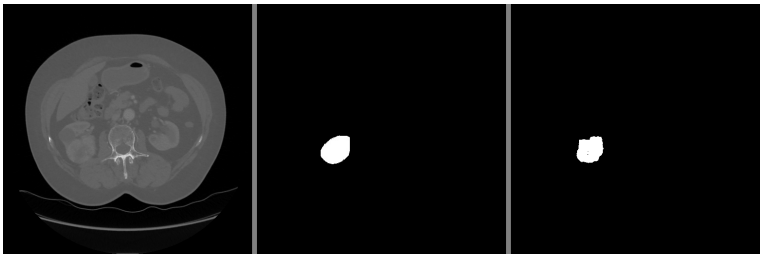
Figure 25: One-Stage results: CT Image - Ground Truth Label - Predicted Label



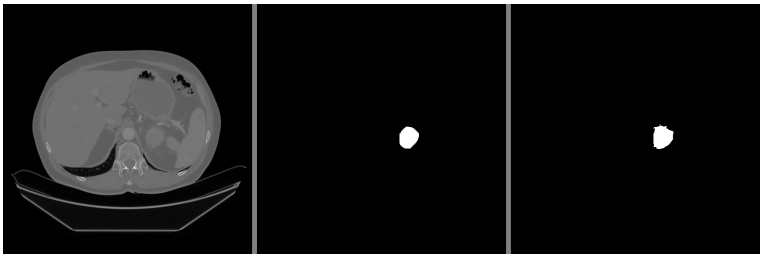
(a) p1, slice 9



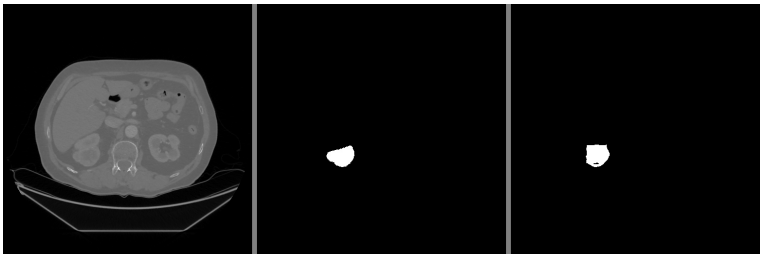
(b) p2, slice 120



(c) p3, slice 90

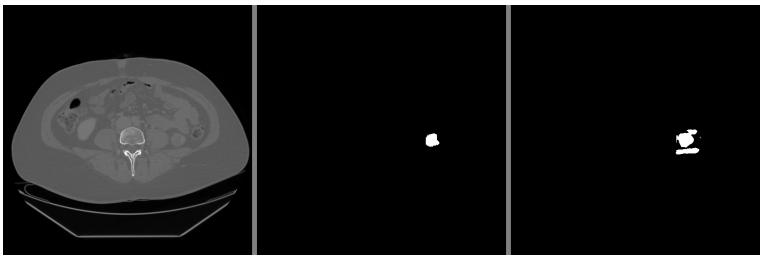


(d) p4, slice 166

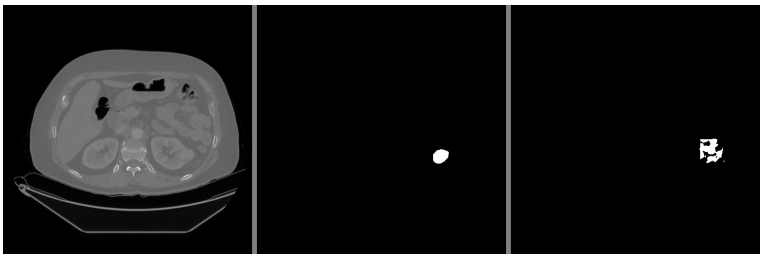


(e) p5, slice 115

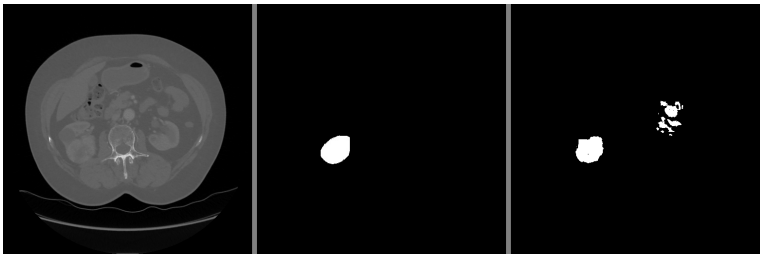
Figure 26: Two-Stage method results: CT Image - Ground Truth Label - Predicted Label



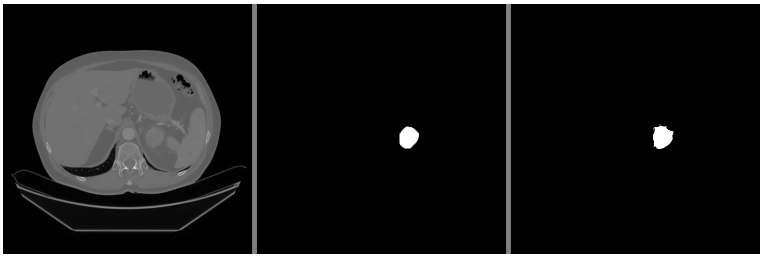
(a) p1, slice 9



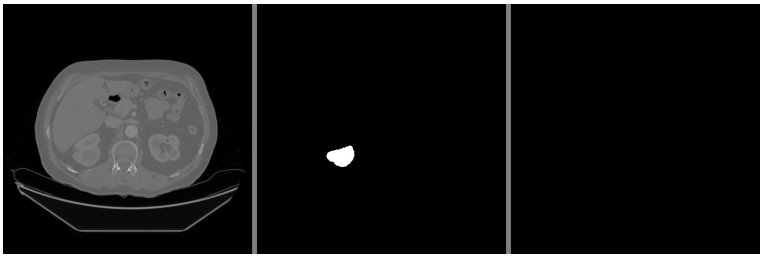
(b) p2, slice 120



(c) p3, slice 90



(d) p4, slice 166



(e) p5, slice 115

Figure 27: Tuned Two-Stage method results: CT Image - Ground Truth Label - Predicted Label

Crop Size	**Dice**
54	0.26158
64	0.26633
128	0.24896
Dataset	**Dice**
Tumor region + basic augmentation	0.28863
Tumor region + multiple crops	0.26633
Model	**Dice**
U-Net	0.28863
U-Net 512	0.28580
U-Net 256	0.23752
Loss function	**Dice**
Dice	0.26633
BCE	0.22885
Tversky	0.25519
Focal Tversky	0.25129
Log Cosh Dice	0.25668
Overlay threshold	**Dice**
0.1	0.26633
0.2	0.22621
0.001	0.38204
0.0005	0.38203

Table 13: Results of trial and error-type optimal hyperparameter search for the two-stage model. Inside each group, the rest of the hyperparameters remained the same, however there is some variation between the groups. The optimal hyperparameters that were used in the best performing model are colored in teal.