UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Computer Science Curriculum

Donatas Vaiciukevičius

Every Click Counts: Deep Learning Models for Interactive Segmentation in Biomedical Imaging

Master's Thesis (30 ECTS)

Supervisor: Dmytro Fishman, PhD

Tartu 2024

Every Click Counts: Deep Learning Models for Interactive Segmentation in Biomedical Imaging

Abstract:
The medical field of radiology is experiencing an unprecedented surge in demand, largely driven by an aging population and consequent pressures on an already understaffed workforce. These pressures have resulted in an increased demand for technological advancements to manage the escalating workload. Although various machine learning solutions have been implemented in some areas to alleviate this strain, significant challenges remain. Notably, manually measuring tumours detected in computed tomography images, a key task in the oncological diagnostic workflow, continues to be labour-intensive and presents an opportunity for improvement. To this end, in this thesis, we explored the potential of employing interactive deep-learning models to assist radiologists, thereby improving diagnostic workflow. We experimented with techniques, such as RITM and FocalClick, for the analysis of computed tomography images. This analysis facilitated our primary contribution - the introduction of dynamic radius disk encoding, which adds a new dimension to the click-based user input and substantially boosts model performance. This innovation reduces the need for repeated interactions and enhances segmentation quality with fewer clicks. Additionally, we present an improved augmentation strategy and introduce a novel metric for evaluating interactive segmentation models. Our results demonstrate the effectiveness of these deep learning methods, particularly when augmented with dynamic radius disk encoding, in streamlining radiological diagnostics. The findings represent a promising avenue for further research aimed at further optimising these techniques for clinical application.

Keywords:
machine learning, deep learning, interactive segmentation models, RITM, FocalClick, dynamic radius disk encoding, computed tomography segmentation, biomedical imaging, tumour analysis, artificial intelligence in radiology, clinical decision support systems, neural networks in healthcare, CT image processing, medical imaging AI, diagnostic efficiency improvements, healthcare automation.

CERCS: P176 – Artificial intelligence; T111 – Imaging, image processing; B110 - Bioinformatics, medical informatics, biomathematics, biometrics;

Iga Klikk Loeb: Süvaõppemudelid Interaktiivseks Segmenteerimiseks Biomeditsiinilises Pildistamises

Lühikokkuvõte:
Radioloogia valdkonnas valitseb erakordne nõudluse kasv, mis on suuresti tingitud elanikkonna vananemisest ja sellest tulenevast survest niigi puudulikule tööjõule. Suurenenud on vajadus tehnoloogiliste uuenduste järele, mis aitaks tulla toime aina suureneva töökoormusega. Mõnes valdkonnas on selle koormuse leevendamiseks rakendatud masi-

nõpet, kuid mitmed kitsaskohad on siiski jäänud lahenduseta. Tähelepanuväärselt tööja ajamahukas ülesanne on üks onkoloogilise diagnostika möödapääsmatu osa - kompuutertomograafiapiltidel tuvastatud kasvajate käsitsi mõõtmine. Käesolevas töös uuriti võimalusi interaktiivsete süvaõppemudelite kasutamiseks radioloogide abistamiseks ning seeläbi diagnostilise töövoo parandamiseks. Kompuutertomograafiapiltide analüüsiks katsetati erinevaid tehnikaid, näiteks RITM ja FocalClick. Nende meetodide uurimisel jõuti dünaamilise raadioplaadi kodeerimise kasutuselevõtuni, mis lisas klõpsupõhisele kasutajasisendile uue mõõtme ja suurendas oluliselt mudeli jõudlust. See uuendus vähendab vajadust korduvateks interaktsioonideks ja parandab segmenteerimise kvaliteeti vähemate klõpsude abil. Lisaks pakutakse välja täiustatud augmentatsioonistrateegia ja tutvustatakse uudset mõõtemeetodit interaktiivsete segmentatsioonimudelite hindamiseks. Meie saadud tulemused demonstreerivad interaktiivsete segmenteerimismeetodite ja nende dünaamilise raadoplaadi kodeerimisega kombineerimise tõhusust radioloogilise diagnostika täiustamisel. Tulemused kujutavad endast paljulubavat suunda edasisteks uuringuteks nende meetodite jätkuvaks optimeerimiseks kliinilise rakendamise eesmärgil.

Võtmesõnad:
masinõpe, süvaõpe, interaktiivsed segmenteerimismudelid, RITM, FocalClick, dünaamiline raadiusketta kodeerimine, kompuutertomograafia segmenteerimine, biomeditsiiniline kujutamine, kasvajate analüüs, tehisintellekt radioloogias, kliiniliste otsuste tugisüsteemid, neurovõrgud tervishoius, CT-pildi töötlemine, meditsiinilise kujutamise AI, diagnostilise efektiivsuse parandamine, tervishoiu automatiseerimine.

CERCS: P176 – Tehisintellekt; T111 – Pilditehnika; B110 – Bioinformaatika, meditsiiniinformaatika, biomatemaatika, biomeetrika;

Contents

1 Introduction **5**

2 Background **7**
2.1 Computed Tomography Imaging 7
2.2 Computational Methods for Biomedical Segmentation 8
2.2.1 Classical Algorithms 8
2.2.2 Deep Neural Networks in Computer Vision 9

3 Methods **19**
3.1 Datasets and Preprocessing 19
3.2 Evaluation Metrics and Strategy 23
3.3 Interactive Segmentation Methods 24
3.3.1 Classical Algorithms 24
3.3.2 Deep Learning Methods 25
3.4 Writing Assistance 27

4 Results **28**
4.1 Experimental Setup 28
4.2 Evaluating Classical Methods on CT Images 28
4.3 One-Click Segmentation with Deep Convolutional Networks 30
4.4 Assessment of Dynamic Click Encoding 31
4.5 Prediction Refinement Capability Evaluation 31
4.6 Examining Generalization Across Datasets 33
4.7 Augmentation Strategy Evaluation 35

5 Discussion **37**

6 Conclusion **39**

7 Acknowledgments **40**

References **41**

Appendix **46**
I. Licence 46

1 Introduction

In the late 20th and early 21st centuries, developed nations witnessed significant medical advancements that contributed to increased life expectancy. Concurrently, urbanisation, along with economic and environmental concerns, led to an increase in childless families, therefore reducing average birth rates. These factors have resulted in an ageing population, increasing pressure on healthcare systems [1, 2]. Furthermore, The growth of the workforce capable of supporting this ageing population has not been able to keep up the pace in recent years. In particular, radiology has been significantly affected. An increasing number of patients, combined with a broader range of therapeutic options available for treating most cancer forms, resulted in further increasing demand for specialists who can analyse medical imaging, such as computed tomography (CT) scans.

Given the increased demand for radiology services, one of the most time-consuming tasks is the visual analysis of the CT scans, which is crucial for most cancer treatment decisions. The treatment of most tumours is often determined by the stage of the disease, assessed using the TNM staging system. This system classifies cancer based on tumour size (T), the extent of spread to lymph nodes (N) and the presence of metastasis (M). Quick visual analysis of these parameters is challenging due to the detailed and precise assessment required to accurately stage the cancer. Therefore, applying computer vision techniques has the potential to streamline the visual inspection procedure, possibly at least partially speeding up the entire diagnostic process.

Early-to-mid 2010s have seen significant advancements in computer vision with the developments in convolutional deep neural networks [3, 4]. Initially, these innovations led to concerns about machine learning methods replacing medical specialists. However, it has become evident that rather than replacing, computer vision tools could augment the work of radiologists instead. Studies as recent as 2023 [5] assure that the most effective use of this technology is as supportive tools. Equipping radiologists with the best equipment available can address the current crisis in this part of the healthcare system.

Based on this, we have defined the goal of this master thesis - to explore the use of machine learning tools to assist radiologists in accurately determining the size and structure of tumours, thereby streamlining the diagnostic workflow. Typically, radiologists spend a significant amount of time detecting, measuring, and reporting all lesions present in any CT scan they examine [6]. To tackle this issue, we focus on speeding up tumour segmentation, leaving the critical task of their detection to radiologists. Recent works have shown that it is possible to prompt a deep neural network to segment lesions in CT imaging slices [7]. However, machine learning solutions, in general, are commonly not robust enough to perform consistently well across diverse and unseen data. Therefore, we aim to enhance the precision of tumour segmentation through interactive segmentation techniques. Applying methods like RITM [8] and FocalClick [9], proven in other visual domains, has the potential of not only generating but also refining tumour segmentations.

We do not only apply these techniques on CT images but also propose adjusting how the user input is encoded in these architectures, leading to measurable improvements in performance.

The primary motivation behind our research stems from the challenges in lung cancer screening initiatives, faced by institutions such as the Radiology Clinic of Tartu University Hospital. Therefore, our main focus is on the segmentation of lung nodules. However, to ensure the generalizability of our methods and explore their potential applications, we also test these approaches on various other types of cancerous tissue.

In the following chapters, we give an overview of prior work in computer vision, followed by a description of our datasets and experiments. Finally, we describe our results, discuss their significance, and ponder future work.

2 Background

Non-invasive imaging methods like computed tomography are one of the main tools in radiologists' arsenal when it comes to detecting and diagnosing tumours [10, 11]. Therefore, many machine learning efforts [12, 7], including this work, have focused on this modality. However, most computer vision techniques are designed with colour images in mind. Due to the technical differences between computed tomography and colour images, adapting methods commonly used with one format to another often poses challenges. We begin this chapter by discussing the main features of CT scans that must be accounted for, then follow up by describing various methods that can be adapted and used for interactive medical imaging segmentation. We start with classical algorithms, which leverage topographic features of input images. That is followed up with descriptions of more complex deep learning based methods.

2.1 Computed Tomography Imaging

Computed tomography (CT) is a technique that produces images of internal organs in a non-invasive fashion. CT scan is a three-dimensional image consisting of 3D pixels, also known as voxels - values with corresponding x, y and z coordinates. A combination of specialised CT scanners and tomographic reconstruction software is used to produce such scans.

A CT scanner (Fig. 1) uses X-ray tubes and corresponding ray detectors arranged on the opposite sides of a ring surrounding a patient. Emitted rays pass through different organs of the human body, losing intensity in the process. The remaining signal is picked up by the detector to measure how much of the ray was absorbed within the patient. The absorption rate varies based on the tissue - for example, soft and relatively coarse lung tissue material absorbs less of the rays than dense bone structures. The ring spins around its axis, measuring the change in ray intensities from various angles of the same plane. The measurements are processed using tomographic reconstruction to compute a cross-section image. The image represents the difference in the emitted and detected ray intensities rather than the detected signal strength, causing more ray-absorbing tissues to appear brighter in the constructed image. During patient examination, contrast material is injected into the patient's body. Contrast strengthens the absorption of X-rays, making blood vessels and organs that receive a lot of blood flow appear brighter and clearer on the scan. The spinning ring is moved along its axis to take measurements of different parts of the human body, and the produced cross-sections are stitched together into a 3D volume, as seen in Figure 1.

As a 3D structure, a computed tomography scan has to be projected into a 2D space to be viewed on a monitor or other two-dimensional medium. While there are various methods to produce such projections, when there is a need to inspect a specific area of the scan, commonly viewing it slice by slice is the preferred approach. This slice tends to be

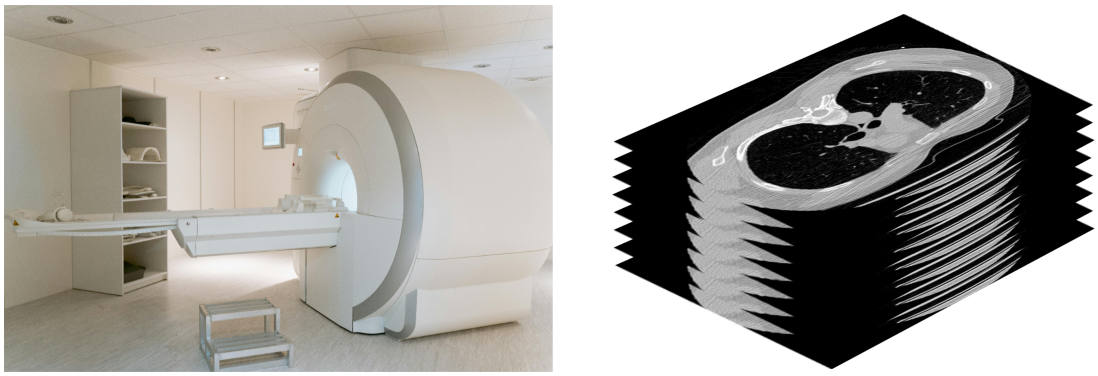


Figure 1. Constructing a CT scan. On the left, a computed tomography scanner is displayed (image from Pexels.com). The patient is laid on the table which moves to the ring that houses X-ray emitting hardware. The signals measured within the ring are reconstructed into cross-section images stacked to create a 3D scan volume.

visualised as a grayscale image, as the measurements are stored in a single channel rather than 3 channels in RGB images. The values of a CT scan are referred to as Hounsfield Units (HU) and fit into a range of $[-1024, 3071]$ (Hounsfield scale) [13]. The higher the value, the more captured substance reduces X-ray intensity. For example, air has an HU of -1000, and water - HU of 0.

2.2 Computational Methods for Biomedical Segmentation

Given the complexity and heterogeneity of CT scans, various computational methods have been considered for their analysis. These methods vary in their complexity and effectiveness. Classical computer vision algorithms tend to be relatively lightweight and simple, while deep learning-based methods trade this simplicity for a potential increase in performance and robustness.

2.2.1 Classical Algorithms

Classical segmentation algorithms are one of the approaches popular for their simplicity and adaptability. Their behaviour is clear and interpretable in contrast to the black-box behaviour of neural networks. When solving an engineering problem, the most simple, efficient solution with sufficient performance is the one to choose. Classical algorithms fulfil the criteria of simplicity, therefore, evaluating their performance in the task of tumour segmentation in CT scans is crucial before considering the real-world application.

One of the approaches examined is the flooding algorithm. It's an iterative process where the segmentation mask spreads based on signal similarity. First, a starting point (pixel) is chosen on the object that should be segmented. Then the algorithm considers

each surrounding pixel - the value of every neighbour is compared to the initial value. The new neighbour is added to the mask if the difference between the two does not exceed a pre-defined threshold. Neighbours of the new pixel are added to the queue for consideration, and the algorithm continues until there are no more pixels to consider. It performs well in homogeneous regions, works fast and is not too sensitive to initial conditions. However, when considering cancerous tissue, it might leave enclosed parts of tumours unsegmented, as there can be significant differences in signal strength.

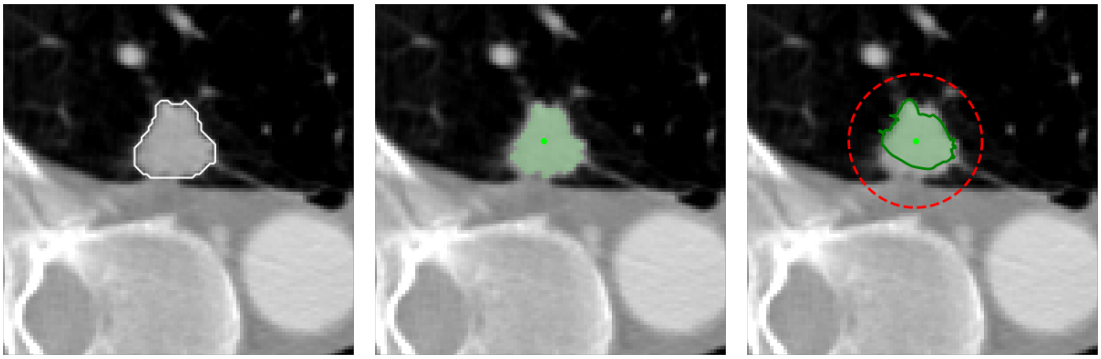
Active Contours algorithm (also referred to as snakes algorithm) [14] focuses on boundaries rather than topography, making it more robust to noisy patterns within the objects. A so-called "snake" is an energy-minimizing spline formed of independent points. The "snake" is iteratively guided by its own energy function, a combination of internal and external forces affecting the spline. Afterwards, the area enclosed within the spline can be turned into a segmentation mask. The internal energy of the curve attempts to keep the "snake" smooth (E_{curv}) and continuous (E_{cont}), while external forces pull the "snake" towards image features like edges (E_{edge}), lines (E_{line}) and terminations (E_{term}). External energies are also given a weight (w), influencing which features should affect the spline more. Given an initial arrangement of points, the algorithm attempts to minimise this energy function through gradient descent.

$$E_{snake} = E_{curv} + E_{cont} + w_{edge}E_{edge} + w_{line}E_{line} + w_{term}E_{term} \tag{1}$$

There are various possible modifications to the definition of E_{snake}, each with their own side effects. For example, incorporating a "balloon" force [15] was suggested to better define the behaviour of the "snakes" in homogeneous areas. We evaluate the original algorithm proposed by Kass *et al.*[14]. While active contour algorithms perform better with difficult topography than flooding, they have their own flaws. The algorithm is very sensitive to hyperparameter values and requires a good initial guess of the object's location. It also has trouble defining edges in noisy images. Therefore, images are commonly blurred in a preprocessing step, partially losing detail. The algorithm is commonly used for contouring, not segmentation. Therefore, its performance suffers when applied to segment hollow objects. Lastly, the approach is a lot slower when compared to flooding and does not guarantee that the fitted spline will only settle around the intended object.

2.2.2 Deep Neural Networks in Computer Vision

While classical methods have their uses and can be applied for certain tasks in computer vision, neural networks have been shown to outperform them in the segmentation of biomedical imaging [16]. The inspiration for neural networks stems from the processes in the human brain researched by neuroscientists. Neurons, the nerve cells that form pathways in the brain, are interconnected and send signals to each other given a particular



(a) (b) (c)

Figure 2. Classical algorithms used for segmenting a lung nodule. Panel (a) illustrates a lung nodule outlined in white. Panels (b) and (c) show masks created by the flooding and active contour algorithms, accordingly. A green dot marks the nodule's centre, serving as the starting point for the flooding algorithm and the initial centre for the active contour ("snake") algorithm. The initial "snake" is depicted by a red dashed line, while the green solid line shows its final shape after completion of the algorithm.

stimulus. Similarly, artificial neural networks are built from smaller building blocks called artificial neurons. These neurons function by combining an array of input values into a single output. That is achieved by applying different weights to each input, aggregating the inputs into a single value, adding a bias and running the result through a non-linear activation function. In artificial neural networks, neurons are arranged in layers, where a later layer receives outputs of an earlier layer as an input. The weights and biases, which are initially set at random, are crucial adjustable parameters. During training, the model initially generates predictions that significantly deviate from the desired outcome. A loss function is used to calculate the error between these predictions and the expected outcomes. This error is then used to adjust the weights and biases through an optimisation process, typically involving gradient descent, to minimise the loss and enhance the model's accuracy over time.

The most basic type of artificial neural network is the fully connected network, in which each neuron is linked to every neuron of the preceding layer. While this structure enables broad information transfer across the network, it is not effective universally. Fully connected networks tend to struggle with imaging due to poor scalability.

Consider a fully connected network designed to classify 128×128 pixel images, a tiny resolution by modern standards. Typically, pixels consist of 3 color channels leading to a total of 49152 inputs. In a fully connected network each neuron must process all of the inputs, consequently requiring 49152 weights per neuron. Given a relatively low size of 512 neurons in the first layer, the network would already contain over 25 million learnable parameters from the input layer alone. Usually, more than one layer is

necessary for effective predictions.

The number of parameters increases even more drastically when increasing image resolution. Doubling it to 256×256 not only quadruples the number of input neurons but necessitates a larger and deeper network to handle the increased amount of raw data. Due to this weakness, researchers often use alternative architectures for image analysis.

One of the most widespread approaches when it comes to neural networks in computer vision is using convolutional filters. A convolutional filter contains a grid of input weights (commonly referred to as a kernel) that can vary in size, with a 3×3 grid being the most common. Inputs to this filter are multiplied by the kernel weights and then summed (along with a possible bias term) to produce a single output value. This output is then typically used as an input for an activation function. The filter covers all input channels at the same time. Therefore, the depth of the kernel has to match the number of input channels. This filter moves across the image in a sliding window manner, producing a 2D grid of output values called a feature map. A single layer contains multiple filters, producing multiple feature map channels. Typically, the network contains consecutive convolutional layers, where feature maps from an earlier layer become inputs for the next one. The early layers extract low-level features from the image, such as edges or diagonal lines, but the deeper we go into the network, the more complex the representations become [17]. Commonly networks have a shrinking structure where the feature maps reduce in resolution, giving the convolutional filters in lower layers more and more global context [18, 4]. The convolutional architecture has demonstrated greater efficiency in visual workloads compared to fully-connected networks.

Commonly, the architecture of a convolutional neural network (CNN) is tailored to its specific task. For instance, when dealing with a classification problem, it is common for a CNN to be comprised of a series of convolutional layers, followed by a classification head constructed from several fully-connected layers. Convolutions are primarily responsible for doing the heavy lifting of feature extraction, while the classification head condenses these features down to a vector of final predictions.

In contrast, a segmentation task requires the network to classify each pixel individually rather than the image as a whole. Using the above-described approach here would be impractical, as the classification head would have to scale up to the resolution of the input image. To address this, Long *et al.* introduced the Fully Convolutional Network (FCN), which employs convolutional layers to produce prediction as a classification map, eliminating the need for fully connected layers. This architecture not only outperformed prior approaches in semantic segmentation but was also groundbreaking in being input-resolution independent. This kind of model could be trained on images of one resolution and then generate predictions on images of a different resolution. This idea quickly took off, and various fully convolutional architectures were soon proposed, such as the U-Net, proposed by Ronneberger *et al.* [19] and shown to be effective in biomedical imaging. This network has been further refined with subsequent ideas, such as U-Net++

and nnU-Net [20, 12].

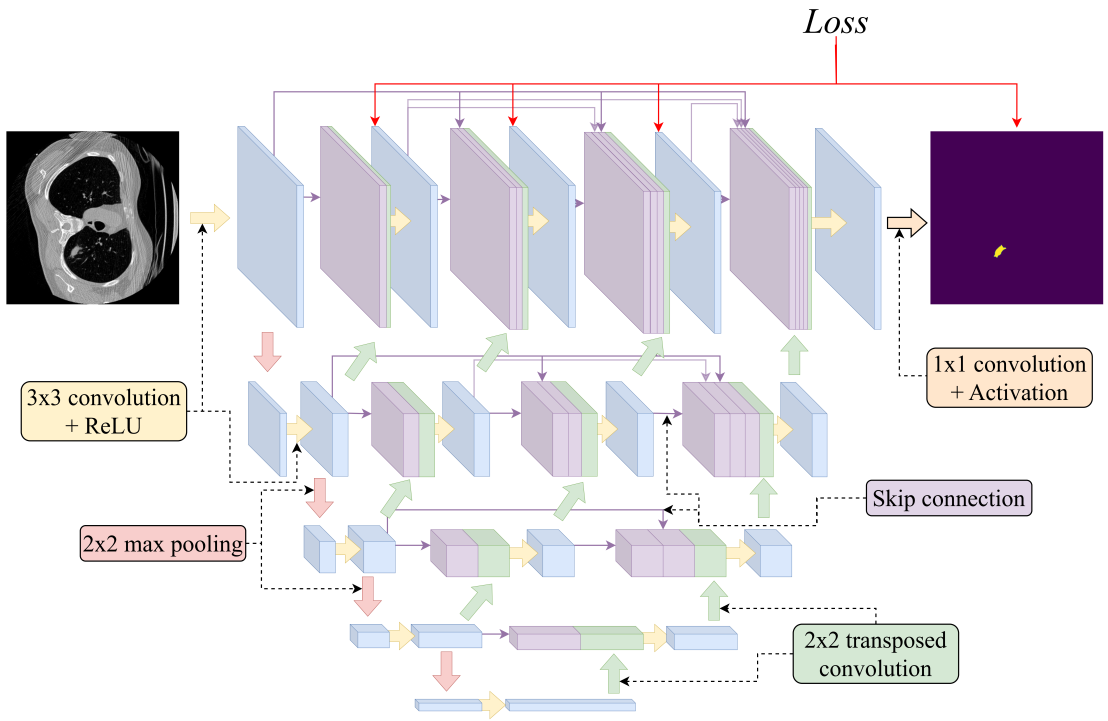


Figure 3. U-Net++. The semantic segmentation model consists of a contracting path (comprised of 3×3 convolutions and max pooling) and an expanding path (2×2 transposed convolutions). The paths are connected by intermediate convolutional blocks and skip connections, deep supervision is employed for the highest resolution layer.

The architecture of U-Net consists of two primary components: a contracting encoder and an expanding decoder. When put together, they form a structure roughly resembling the letter "U", which is the inspiration for the model's title. The encoder has a series of convolutional layers stacked with max-pooling in-between to down-sample the feature maps until a high-level representation is reached. However, instead of being used for prediction, the high-level representations are up-sampled back to the original size and gradually squeezed into fewer channels, finishing with a channel per predicted class. Transposed convolutions are employed for up-sampling, rather than simpler techniques like bilinear or nearest neighbour interpolation, allowing the network to learn more optimal upscaling of feature maps. Additionally, for each level of the decoder, the corresponding resolution feature maps from the encoder are concatenated to the outputs from the lower resolution block, providing additional information to help with the final prediction. The idea for skip connections was introduced with the ResNet [4], where their primary purpose was to ease the training of deep convolutional models. However,

in U-Net, they primarily serve to reintroduce early context to the decoder mechanism.

U-Net++ [20] is an evolution of the U-Net, redesigning the skip connections between encoder and decoder branches. Rather than passing the encoder feature maps directly to the corresponding decoder block, they undergo transformations through multiple convolutional blocks. Additional blocks are added to every pathway. Each block combines features from the earlier blocks of the same path and up-sampled features of a corresponding lower-resolution block. It is hypothesised that adding in-between convolutions allows for a more gradual transformation of features between the encoder and the decoder, allowing for easier optimisation. Additionally, deep supervision [21] is used for each top-level convolutional block to enhance training effectiveness.

However, convolutional segmentation models do not have to be limited to an encoder-decoder architecture. A good example is the HRNetV2 [22] semantic segmentation model. Instead of a single signal path, it employs a multi-stage, multi-scale architecture. The first stage of the network contains high-resolution bottleneck convolution blocks. However, consecutive stages add additional convolution streams where the feature resolution is halved. In between every stage, feature fusion modules exchange information across representations of different resolutions. Higher resolution features are down-sampled via 3×3 convolutions with a stride of 2, while lower resolution maps are squashed with a 1×1 convolution and up-sampled through bilinear interpolation. At the end of a network, a feature aggregation module up-samples features from lower-resolution streams via bilinear interpolation, and a final prediction map is produced by a 1×1 convolution and an activation function.

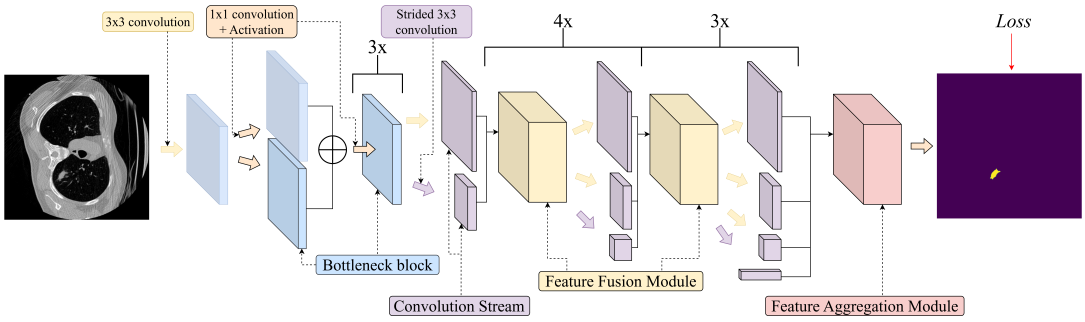


Figure 4. HRNetV2. The model consists of multiple parallel convolution streams, which generate feature maps at different scales. Feature fusion modules are used to share information between streams of different resolutions and final prediction is produced from aggregated features of all resolutions.

While the aforementioned architecture is capable of generating effective segmentations, it is best employed as a feature extractor, as demonstrated with HRNetV2+OCR [23]. The principle behind the OCR (Object-Contextual Representations) module is that a pixel's class is determined by the label of the object to which the pixel belongs. Thus,

the prediction can be enhanced by considering the region surrounding the pixel. Unlike methods relying on dilated convolutions [24], OCR leverages the object's context to enhance pixel-level predictions. The feature maps generated by the HRNetV2 backbone are considered to be soft object regions, where each pixel is associated with every region to some degree. An object region context vector is derived from these regions. Subsequently, the initial feature maps are adjusted by weighting each pixel's association with each object region, refining the final prediction.

This is achieved by a specialised form of the self-attention mechanism. The attention mechanism [25] identifies and prioritises data segments most relevant within a specific context. It operates with three inputs: a key (input data to be analysed), a query (representation of what should be concentrated on) and value (data from which the information is extracted). This mechanism calculates an attention score by correlating the key with the query, indicating the amount of focus each value should receive. Then, a dot product of the attention score and the value is produced to extract the relevant information. In the case of OCR, object region context functions both as a key and value for the self-attention mechanism, while features from the backbone serve as the query.

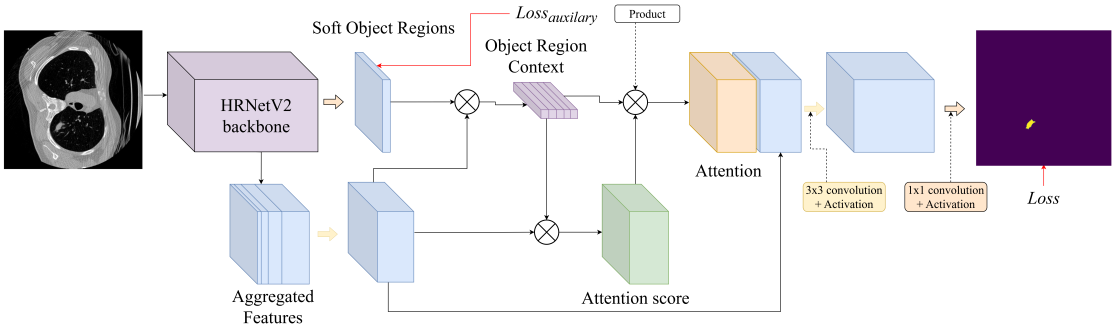


Figure 5. HRNetV2+OCR. The approach employs HRNetV2 as a backbone. Its predictions and features are used in a specialized version of self-attention, further enhancing the predicted segmentation masks.

While semantic segmentation models have their use, they lack a particular functionality we're looking for - influencing the prediction by user guidance. This is achievable with a specialised branch of computer vision known as interactive segmentation. Interactive segmentation researchers commonly re-purpose semantic segmentation models to encode the user's input into their pipeline and guide the prediction accordingly [8, 26]. Numerous interactive segmentation models and architectures exist, yet their application in medical imaging is still emerging. Notably, many of these models are yet to be tested on imaging types such as CT scans.

At the moment of writing, no single approach dominates interactive segmentation benchmarks. Methods such as FocalClick, SimpleClick, and ICL CFR demonstrate varying performance levels across different datasets [9, 26, 27]. However, these methods

are built upon concepts originally proposed by Sofiiuk *et al.* [8] with RITM. This method facilitates the integration of user input into a feed-forward network without major changes to the backbone while also making use of existing incomplete segmentation masks. Additionally, together with RITM a now commonly used sampling strategy was proposed to generate artificial user input for training purposes.

The concept of RITM is clear and intuitive yet robust. First, user input has to be integrated into an existing model. Commonly, it is denoted as click coordinates in an image. A click can be positive, indicating that the marked underlying object should be segmented, or negative, suggesting otherwise. For use within the model, clicks must be transformed into a spatial form. In RITM, they are turned into small constant radius disks in a binary map. The map is split into two channels for positive and negative inputs and serves as additional input for the model. Interactive segmentation is commonly an iterative process, producing intermediate predictions during inference. RITM uses these imperfect segmentations by incorporating the most recent prediction as another input channel, further enhancing the model's precision.

One of the main design philosophies with RITM is that a backbone should not be modified to incorporate additional information into the model. Instead, three channels—comprising click maps and a pre-existing segmentation mask—are processed through an independent convolutional block. This block generates guidance feature maps that align with the dimensions of the outputs from the backbone's first convolutional block, in this case, HRNetV2+OCR. These guidance feature maps are then element-wise summed with the outputs from the initial convolutional block. This method ensures that the core architecture of the backbone remains unchanged, resulting in easier fine-tuning.

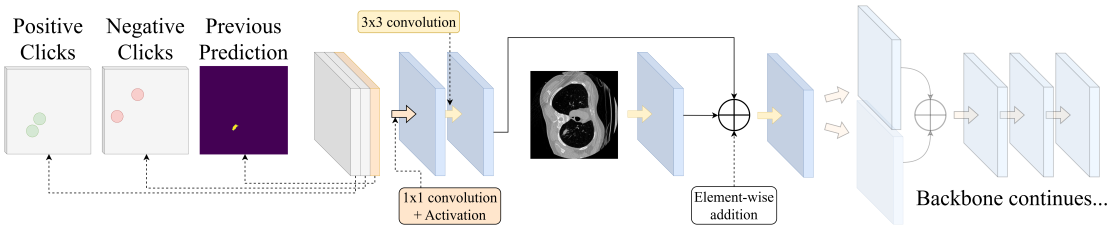


Figure 6. RITM. Click features and a previous mask are passed through a convolutional block, reshaping the tensor to match the dimensions of backbone features. This tensor is then incorporated into the backbone via element-wise addition.

During the inference phase, users interact with the model — typically through a bespoke user interface — to specify which object should be segmented. Based on the input, the model predicts a segmentation mask, and if improvements are required, the user adjusts the segmentation by providing an additional click. Then, both clicks and the prior segmentation are fed into the model. In such cases, all inputs are cropped to focus on the region of interest. This region is based on the bounding box of the existing

prediction, which is expanded with a ratio of 1.4 to ensure parts of the object of interest are not cropped away. The crop is then interpolated up to at least 400 pixels per axis. After inference, the produced segmentation is readjusted to fit into the original image's dimensions.

However, training an interactive segmentation model is less straightforward than a semantic one. The guidance information, usually produced by the user during inference, has to be simulated. An iterative sampling strategy where each subsequent click would be based on a previously predicted mask best resembles an interaction with a real user. However, it is computationally expensive. Random sampling is cheaper but does not correlate with real-world use. Therefore, a combination of both strategies is used. Some points are sampled randomly for each training batch, and up to 3 points are sampled iteratively during training. Standard image augmentations such as random cropping, flipping, color-jitter, etc., can and are used with this approach during training.

FocalClick [9], an evolution of RITM, aims to increase the efficiency of its predecessor and improve the perfection of existing masks with an additional refinement stage. It is argued that RITM is inefficient when refining an existing prediction, as the entire area around the mask is re-segmented, although the new click only suggests changes in a smaller area. This slows inference down and allows for unintended changes in the new prediction.

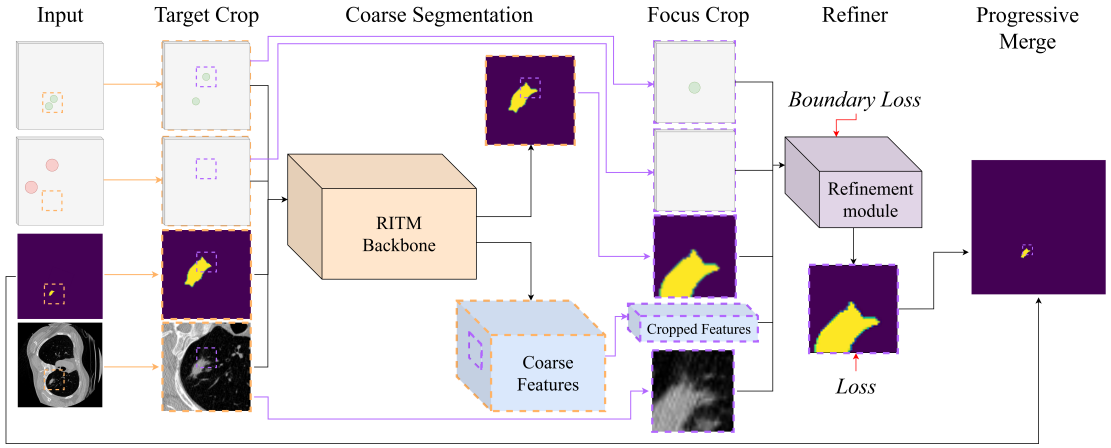


Figure 7. FocalClick. The architecture relies on two prediction stages. In the first stage, Target Crop is made around the existing mask and the latest click, and prediction is produced by the RITM backbone. In the second stage, a tighter Focus Crop is produced around the area, which changed after the coarse prediction. This crop is then passed through a refiner block, further enhancing the prediction. Lastly, the predicted mask is merged with the previous prediction in a Progressive Merge step.

Instead, the FocalClick pipeline employs two separate crops. The initial "Target Crop" encompasses the existing mask and the latest click with a bounding box, which

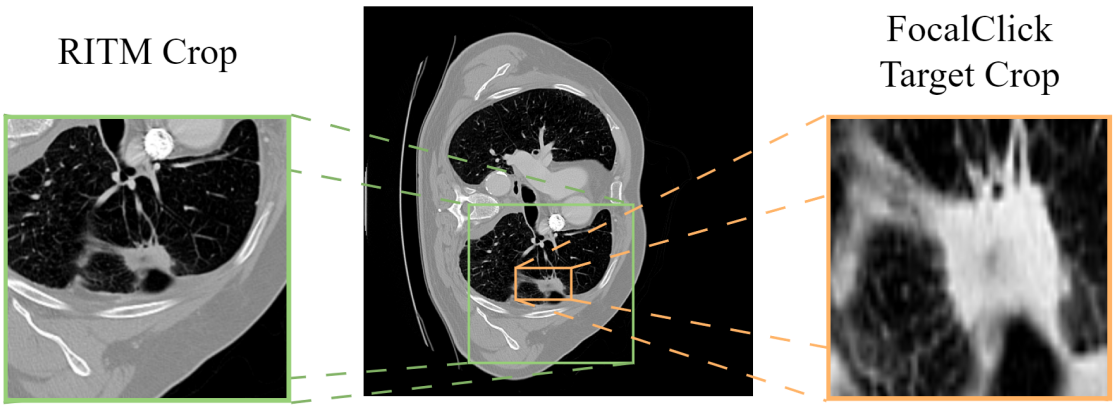


Figure 8. Comparison of image cropping between RITM and FocalClick. Target Crop encloses the region of interest tighter and reshapes it to a square aspect ratio.

is then expanded, similarly to RITM. However, the crop can be much smaller, with the minimum resolution limited to 10×10 pixels. However, unlike RITM, where original proportions of the crop are maintained during up-sampling, FocalClick scales the crop to a fixed size of 256×256, regardless of the original aspect ratio. This ensures consistent inference speed but causes distortion to the input image. If no prior segmentation mask exists, "Target Crop" is bypassed, and the full image is resized to the square 256×256 dimensions. A prediction is then performed using a backbone network, employing the same model guidance strategy as RITM. This is considered a coarse segmentation.

Subsequently, a "Focus Crop" is performed, targeting the specific region intended to be modified. It is computed by calculating a difference map between previous and new segmentations, identifying regions with changes and extracting a bounding box around the connected region containing the latest click. The cropped image, clicks, coarse segmentation and feature maps are used in a refiner block composed of standard and Xception [28] convolutions. The block contains two heads that predict a detail map M_d and a boundary map M_b. Then the refined prediction M_r is calculated by updating the coarse prediction logits M_l as in equation 2

$$M_r = Sigmoid(M_b) * M_d + (1 - Sigmoid(M_b)) * M_l \tag{2}$$

In this calculation, the predicted boundaries of the object are used as weights in a weighted sum of details predicted by the refiner and initial coarse segmentation. In the areas with high boundary values, fine predicted details are emphasised. Otherwise, initial coarse segmentation is mostly sufficient.

FocalClick concludes with a "Progressive merge" step, ensuring that only intended segmentation modifications are made. Similarly to the Focus Crop stage, a difference map is computed between the previous and new predictions, and the only connected region containing the latest click is incorporated into the previous prediction.

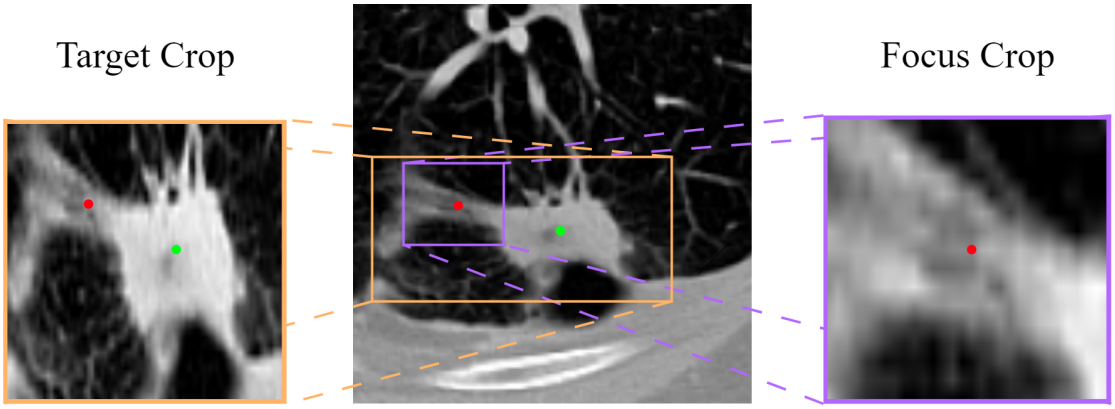


Figure 9. Comparison of Target and Focus crops in FocalClick. User input is denoted in green for positive clicks and red for negative. The Focus crop is confined to the vicinity of the most recent input, whereas the Target crop encompasses all provided inputs. With both cropping strategies, the resulting image is afterwards squeezed into a square $1x1$ aspect ratio.

The training procedure for FocalClick contains minimal changes when compared to RITM - click sampling strategy remains unchanged. However, the refiner block is supervised separately. Input crops to the refiner are sampled by either calculating an expanded bounding box for the ground truth of the object (imitating inference with no prior mask) or generating smaller random crops centred on the boundaries of the ground truth mask. The refiner's output is supervised by corresponding crops of the ground truth mask. Meanwhile, the boundary predicting head is supervised by an object's border map, generated by downsampling and upsampling the ground truth, then identifying the pixels which changed in the process.

3 Methods

Building on top of the ideas discussed in the earlier section, we now shift our focus to the specific methods employed for this thesis. Below, we outline the techniques used to adapt, evaluate and improve upon interactive segmentation methods used for tumour segmentation from CT images. Additionally, we describe our main contributions - dynamic radius disk click encoding, Optimal Crop augmentation strategy and the Iterative IoU Error metric.

3.1 Datasets and Preprocessing

Computer tomography scans are three-dimensional volumes, yet the majority of interactive segmentation models are optimised for two-dimensional inputs. To bridge this gap, we extract 2D slices from CT volumes. These slices, functioning as single-channel images, are better suited for the segmentation tasks at hand. They are acquired from publicly accessible datasets that include a range of tumour types and corresponding annotations. Although some datasets provide granular details like tumour classifications or cyst annotations, our focus is on generating binary segmentation maps that solely delineate tumour regions. To ensure uniformity, all slices are resized to a consistent resolution of 512×512 pixels, using padding and resizing techniques for scans of non-standard dimensions. We employ the following datasets as our 3D volume sources:

- **LIDC-IDRI [29]**: This dataset, obtained from the Cancer Imaging Archive, contains scans of patients who have participated in the lung screening trial. These patients' CT scans contain a number of potentially cancerous lung nodules. Each scan is annotated by at least four different specialists, providing a robust consensus for detected nodules. As our primary motivation is enhancing lung cancer screening initiatives, slices from the LIDC-IDRI dataset are central to our experiments.

- **KiTS [30]**: We used data from the 2023 Kidney and Tumour Segmentation Challenge, containing 489 CT volumes. These scans come with manually annotated kidneys, tumours, and cysts. The scans were collected from M Health Fairview Medical Center and made publicly available through the KiTS competition.

- **LiTS [31]**: This dataset, originally used for the 2017 Liver Tumour Segmentation challenge, includes 130 scans of liver cancer patients from six medical centres.

- **Medical Decathlon [32]**: We incorporate pancreatic cancer patient scans from the Medical Segmentation Decathlon dataset. The dataset encompasses scans from the Memorial Sloan Kettering Cancer Center and corresponding pancreas and tumour annotations.

We employ two sampling strategies to cater to different experimental needs, creating two distinct sets of 2D images from each 3D volume dataset. The first strategy selects slices featuring the largest tumour area within each scan, forming a more condensed dataset ideal for rapid model training and exploratory design decisions. These datasets are marked with an $_S$ suffix. The second strategy includes every lesion-containing slice, regardless of size, resulting in a big and diverse dataset that provides a robust challenge for the models and a thorough evaluation framework. The smaller set is utilised for development and ablation studies, while the larger set is reserved for comprehensive method evaluation.

We distribute our slices across three subsets to enable both training and fair evaluation of our models: 70% for training, 10% for validation, and 20% for testing. Additionally, we compile combined datasets to evaluate the possibility of training models generalized for various types of tissue. To maintain a balanced representation, the sample size from each dataset is adjusted to match the size of the smallest dataset. Training, validation and test splits are preserved in the combined dataset, allowing fair comparison across datasets. datasets.

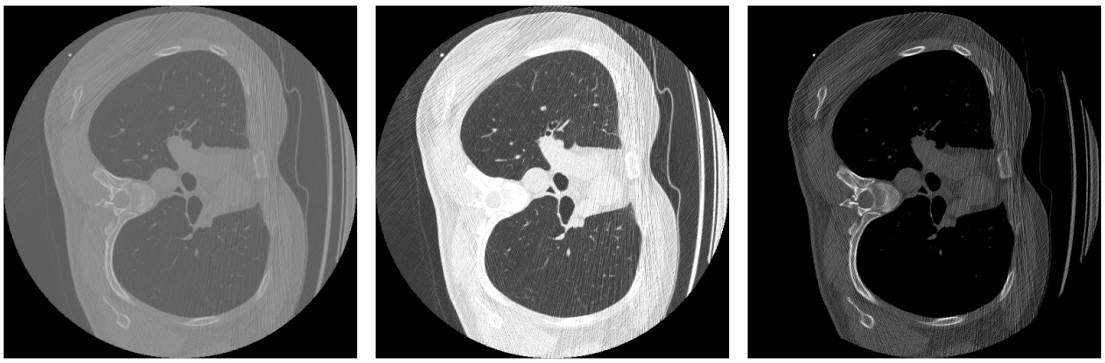
Dataset	Data source	Split size		
		Train	Val	Test
LIDC 2D	LIDC-IDRI	5097	830	1680
LIDC 2D$_S$		416	57	127
KITS 2D	KiTS	8527	1219	2436
KITS 2D$_S$		342	49	98
LITS 2D	LiTS	5033	719	1438
LITS 2D$_S$		82	12	24
MD PANC 2D	Medical Segmentation Decathlon	1,775	254	508
MD PANC 2D$_S$		196	29	56
COMBINED 2D	LIDC 2D, KITS 2D, LITS 2D, MD PANC 2D	7100	1016	2032
COMBINED 2D$_S$	LIDC 2D$_S$, KITS 2D$_S$, LITS 2D$_S$, MD PANC 2D$_S$	328	48	96

Table 1. Computed tomography imaging datasets created for our interactive segmentation experiments.

During training and evaluation, preprocessing of CT slices begins with an important windowing step. It involves clipping the wide range of HU values found in CT scans to a narrower, more precise range. This process is consistent with methods applied in clinical settings - when evaluating a CT scan, radiologists choose a particular HU range which highlights details of the specific region of interest. The parameters defining a window include its width and level, where the width determines the range of HU values displayed,

and the level demarcates the midpoint of this range. We employ an appropriate window for each organ type in all our experiments.

After windowing, the preprocessing steps diverge based on the segmentation approach. In the case of classical algorithms, the slice is normalised to a mean of 0 and a standard deviation of 1. For deep learning methods, the slice tensor is duplicated across 3 channels and normalised to the mean and standard deviation of the ImageNet [33] dataset. This last step aims to best employ deep learning models with backbones pre-trained on colour images.



(a) (b) (c)

Figure 10. Windowing effect on a computed tomography scan slice. Panels display the same slice with different windowing parameters. Panel (a) presents a slice with no window applied, (b) and (c) displays the slice with lung and bone inspection ranges, respectively

We employ three different data augmentation strategies during model training. They are mostly based on the same principles - random crops and flips.

- **RITM-based**: This strategy, adopted from the RITM codebase, involves resizing the image to scales varying between 75% and 140% of its original size. This is followed by horizontal flips, applied 50% of the time, zero-padding and random cropping to dimensions of 480×320.

- **FocalClick-based**: While employing similar principles as RITM, this approach contains some modifications. It incorporates both horizontal and vertical flips, applied 30% of the time, and expands the image crop range to between 50% and 140%. Importantly, the crop intensity for horizontal and vertical axes is picked independently, simulating distortion of the Target crop. Lastly, image dimensions are resized to a square patch of 256×256. This strategy is adopted from the FocalClick codebase.

- **Optimal Crop**: During the evaluation of the FocalClick model, particularly its application to the detection of lung nodules within the LIDC 2D dataset, existing augmentation strategies were identified as a limiting factor. The primary challenge was observed during the Target Crop step of the inference phase, where the highly restrictive cropping around the nodules led to a significant decrease in model performance. This issue was attributed to the disparity between the training phase's crop intensity and the more aggressive cropping required during inference. To bridge this gap, the Optimal Crop augmentation strategy was developed. This strategy is aimed at refining the alignment between these two phases, ensuring that the model is well-adapted to the conditions under which it is tested. Rather than employing an arbitrary cropping range, we opted to determine the crop scale using the validation set of the dataset. We calculated bounding boxes for the largest masked object in each image, deriving average dimensions. Let $\overline{h}$ and $\overline{w}$ represent the average height and width, respectively, with H and W being the image dimensions, the crop scale range can be defined as:

$$b_{low} = \frac{\overline{h}}{H} + \frac{\overline{w}}{W}; b_{high} = b_{low} \times 3 \tag{3}$$

  This cropping strategy is applied in 80% of cases. The parameters for crop probability and crop scale were determined through a grid search on the CT slices from the validation split of LIDC $2D_S$ dataset.

We remove brightness, contrast, and RGB shift augmentations originally used in RITM and FocalClick, as, based on our observations, they did not bring a benefit while training.

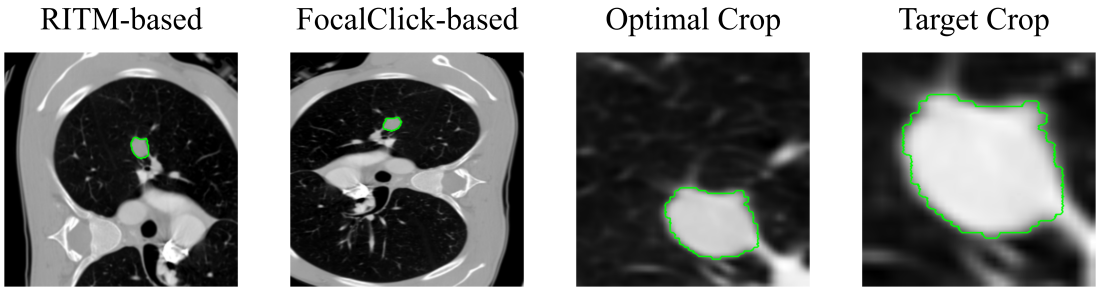


Figure 11. Comparison of crop production using various augmentation strategies against the Target Crop step in the FocalClick inference process. The Optimal Crop augmentation strategy demonstrates the closest alignment with the inference procedure. Ground truth mask is represented by a green contour

We have adapted the data loading procedure from FocalClick for the training of all of the tested deep learning models. FocalClick's original procedure includes additional data

generation steps crucial to train its refinement block. This involves additional crops to simulate the Focus Crop inference step and generating object boundary masks essential for the supervision of refinement block training. This additional generation step can be disabled, allowing the procedure to be universally applied across all trained models.

A significant modification was required in the implementation of this data loading strategy. The original procedure included a preprocessing step that eliminated objects smaller than 900 pixels from the ground truth masks. Considering the characteristics of our datasets, where the median size of the masks in our validation sets is only 364 pixels, maintaining this step would lead to the exclusion of a substantial portion of our data. Therefore, we omit this step to preserve the integrity of our datasets.

3.2 Evaluation Metrics and Strategy

Accurate and meaningful evaluation of predicted masks is crucial for comparing different tumour segmentation approaches. To this end, we employ a widely used semantic segmentation metric - Intersection over Union (IoU). However, IoU alone proves insufficient for evaluating interactive segmentation models, which predict multiple segmentation masks iteratively. Consequently we have defined a novel metric, the Iterative IoU Error. This metric consolidates the IoU values calculated at each iteration into a comprehensive score, facilitating the evaluation of models that incorporate user interactions over time.

Both of the above-mentioned metrics rely on pixel-level classification of segmentation results. Given a tumour is the class we intend to segment, each pixel of a predicted mask can be classified as follows:

- True positive (*TP*) - a pixel correctly predicted to be a part of the tumour.
- True negative (*TN*) - a pixel correctly predicted to be a part of the background.
- False positive (*FP*) - a background pixel misclassified as a part of the tumour.
- False negative (*FN*) - a tumour pixel misclassified as background.

Based on these classes, Intersection over Union can be defined as:

$$\text{IoU} = \frac{TP}{TP + FP + FN} \tag{4}$$

The IoU score increases as the denominator (the union of predicted and actual areas) decreases, and the numerator (accurately identified true positive region) increases, reaching up to 1 for perfect segmentation.

Although a metric like IoU is sufficient for the evaluation of semantic segmentation, where only one prediction is generated per input image, deterministic and fair evaluation of such models presents a challenge. User input has to be simulated consistently, and

representation of iterative mask refinement must be incorporated into the metrics. We base our iterative evaluation strategy on RITM [8]. IoU scores are measured for predictions generated with up to 20 subsequent clicks. The first click is placed in the centre of the largest connected component of the ground truth mask and is always positive. Subsequent clicks are positioned based on an error map, which represents false positive and false negative areas. The biggest connected component is picked from this map, with the new click being placed in the middle of it. The new click is categorised as positive when the component is a false negative and negative for a false positive.

Commonly, metrics such as the Number of Clicks (NoC) and the Number of Failures (NoF) are used to judge iterative segmentation performance. NoC reports an average number of clicks required to reach a predefined IoU threshold, while NoF indicates in how many samples this threshold has not been reached. Although commonly used, they may not accurately reflect all aspects of model performance. For instance, these metrics might show optimal performance when the prediction meets a predefined IoU threshold, even if the following clicks do not improve, or even degrade the IoU

To address this limitation, we propose the Iterative IoU Error. This metric encapsulates the model's performance across all interactions into a single scalar value. Effectively, the weighted area above the average IoU per click curve is measured. The weight of the error increases with every subsequent click, penalizing the model more for low-quality late-stage predictions. The aggregated error is normalised based on the maximum possible error. Given that n is the maximum number of clicks and $\overline{\text{IoU}}_i$ is average IoU after $i+1$ clicks, Iterative IoU Error is defined as:

$$\text{Iterative IoU Error} = \frac{\sum_{i=0}^{n-1}(1-\overline{\text{IoU}}_i)\times(1+0.1i)}{\sum_{i=0}^{n-1}(1+0.1i)} \tag{5}$$

The values of the proposed metric range between $[0, 1]$, with a lower error representing better segmentation performance.

3.3 Interactive Segmentation Methods

Our interactive segmentation methodology can be split into 3 major stages. Initially, we assess classical algorithms for the task of guided segmentation of CT images. Subsequent research focuses on examining interactive deep-learning models with iterative click-based guidance. Finally, an enhancement in the encoding of user inputs is proposed to provide the models with additional information about the dimensions of the segmentation targets.

3.3.1 Classical Algorithms

First, we assess Flooding and Active Contours algorithms in the context of interactive lung lesion segmentation. Performance of both of these algorithms heavily relies on their

initial configuration - threshold parameter for the Flooding algorithm or circle radius in the case of Active Contours. We perform a parameter search to set universal states for these models. The starting coordinates for each sample are picked based on connected components in the segmentation mask - we place one click in the middle of the biggest connected region.

3.3.2 Deep Learning Methods

The tested classical segmentation methods lack an interactive refinement mechanism that would allow for the refinement of existing predictions. To overcome this limitation, we employ deep learning models, which are either designed for continuous user interaction or can be readily adapted for such tasks. Initially, we define a baseline based on a semantic segmentation model proven effective in medical imaging contexts. That is followed by the adaptation, evaluation and modification of interactive segmentation methods.

To set a baseline, we employ the U-Net++ model for the task of interactive segmentation. The objective of this experiment is to assess the extent of iterative mask refinement achievable by modifying a conventional semantic segmentation model without employing architectural ideas from iterative interactive segmentation. Our adaptation uses the first five convolutional blocks from a ResNet50 [4] as the encoder in the U-Net++. Additionally, we add two channels to the input layer, designed to process positive and negative click maps constructed from decaying Gaussian disks. The signal from these disks is strongest at the centre of the click and decays to zero at a radius of 5 pixels. To train these baseline models, we employ an RITM-based training and data augmentation strategy.

In the search for an optimal solution for interactive tumour segmentation, we evaluate RITM and FocalClick as some of the leading approaches on other datasets. HRNetV2+OCR is utilised as the backbone for both methods. We make use of weights pre-trained on the combination of COCO [34] and LVIS [35] as proposed by Sofiiuk *et al.* [9]. Our examination of these approaches mostly focused on optimising data preprocessing, training and inference configuration. Given our focus on lung lesion segmentation, we have employed images from LIDC $2D_S$ to judge most of our adjustments.

The primary objective of our research was to reduce the number of segmentation iterations necessary to achieve optimal quality. Consequently, we propose a novel method designed to maximise the information relayed to the model through a single user interaction. Unlike the fixed-size click encoding used by existing techniques such as RITM and FocalClick, our approach introduces variable-sized disks, which allow for more precise target object delineation by also estimating its area. We refer to models employing this method as dynamic disk models, indicated by a subscript "D" in the results section.

The implementation of dynamic size disks necessitated several modifications to the training, inference, and evaluation protocols. If deployed in practical applications, these

changes would also require a redesign of the user interface to allow users to click and drag, thus specifying the size of the target. While still considered a click, this method significantly alters the nature of the information passed through such interaction.

One of the major adjustments necessary to incorporate dynamic disk model evaluation was to the evaluation procedure. While the location for the next click is chosen as before, based on the centre of the biggest erroneous region, now the size of the disk has to be relayed. We define three different strategies for the disk radius calculation:

- Enclosing radius: This method results in a disk that ensures every pixel of the connected error region is encompassed within the disk's boundaries.
- Tight-fitting radius: This strategy employs the shortest distance between the region's centre and its boundary as the radius.
- Average radius: The mean of tight-fitting and enclosing radii.

These strategies cater to different user preferences and interaction styles. For example, one specialist might choose a tight-fitting click to precisely mark a cancerous region, while another might use an enclosing click while having the same intentions. For deterministic results during model evaluation, we use the average radius. However, we calculate both the enclosing and tight-fitting disks during training. The radii are employed as boundaries of a possible disk radius range, which is expanded by 10% on both ends to account for varying user inputs. We then select the radius for each click from a normal distribution within this range.

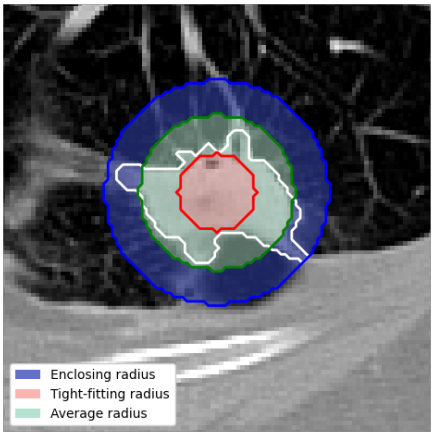


Figure 12. Comparison of dynamic disks produced by different radius calculation strategies. The ground truth mask is contoured in white.

Given the modifications in click dynamics, additional training adjustments were essential. A random sampling of clicks became infeasible as the size of each click is now dependent on the prior segmentation mask. Continuing with the previous sampling strategy would result in negative clicks that excessively expand until reaching the cancerous tissue. This behaviour is unrepresentative of the clicking strategy applied during inference, which typically results in an initial large positive disk followed by progressively smaller disks for further refinement.

Therefore, for training models that incorporate dynamic disks, we only employ an iterative clicking strategy. For each training batch, the total amount of clicks n is drawn from an exponential decay distribution ranging from 0 to 24. $n - 1$ inference iterations are performed, each spawning a new disk based on the existing prediction and the ground

truth. Lastly, the final click is placed, and loss is computed based solely on this last iteration.

3.4 Writing Assistance

It is important to note that machine learning based tools were used to assist with the writing of this thesis. The writing assistant "Grammarly" [36] was used to fix spelling, punctuation and grammar mistakes and to adjust the writing style to fit academic writing guidelines better. Additionally, the chatbot "ChatGPT" [37], based on a generative pre-trained transformer language model (GPT-4), was used for sentence paraphrasing and text restructuring.

4 Results

In the following section, we present and analyse the results of our experiments. We display the possibilities of segmenting lung nodules by employing classical segmentation algorithms. Afterwards, we move to evaluate deep-learning-based interactive segmentation approaches, testing both one-click scenarios and iterative mask refinement. Finally, we assess the model's capabilities at generalizing on unseen CT data and evaluate the effectiveness of the proposed Optimal Crop augmentation strategy.

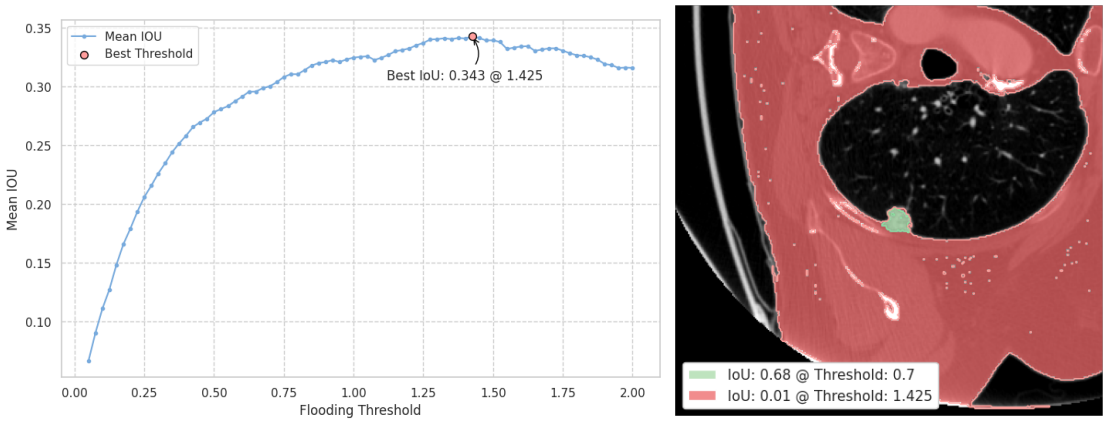
4.1 Experimental Setup

During our experiments, all deep learning models were trained with an adapting learning rate strategy. We begin the training with a learning rate of 0.0005 for all experiments. This rate is decreased by a factor of ten each time the IoU metric plateaued on the validation set. Similarly, we employed an early stopping mechanism, halting training when the validation metrics ceased improving. For our main datasets, the learning rate adjustment patience was set to 10 epochs, and the early stopping patience was set at 30 epochs. For experiments using the smaller counterparts of each dataset, these parameters were adjusted to 25 and 75 epochs, respectively. We train the models using Adam [38] optimiser and a combination of equally-weighted Binary Cross-Entropy and Dice losses. Due to the observed decrease in segmentation performance in our datasets, we disable Progressive Merge for FocalClick in our experiments.

Our experiments were performed on the University of Tartu High Performance Computing Cluster [39]. All experiments were implemented in Python programming language, making use of its various public libraries. Most notably, classical algorithms were implemented and evaluated using scikit-learn [40], and deep learning models were all built in PyTorch [41]. We made use of Hydra [42] for experiment configuration and Weights&Biases [43]. Our code is available at https://github.com/Donce530/Every-Click-Counts-Deep-Learning-Models-for-Interactive-Segmentation-in-Biomedical-Imaging.

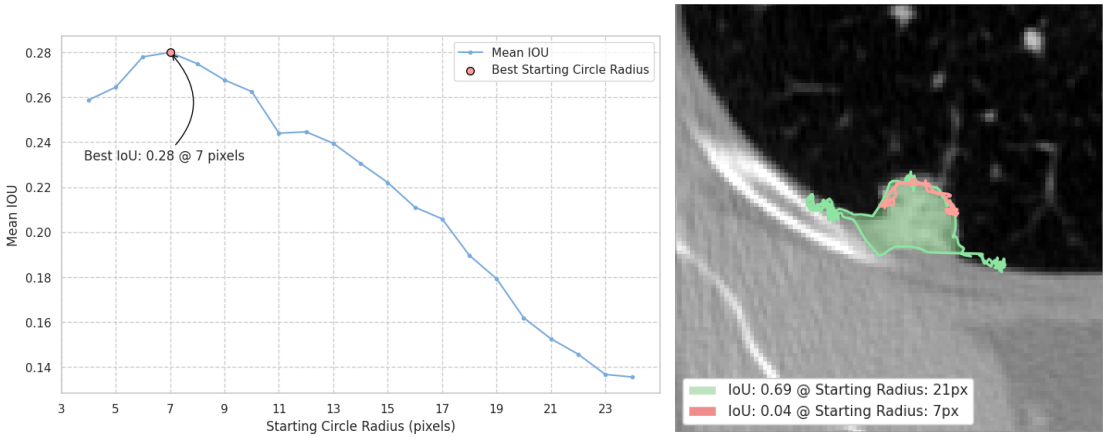
4.2 Evaluating Classical Methods on CT Images

We begin with the evaluation of Flooding and Active Contours algorithms. We test the segmentation performance on the LIDC 2D dataset and optimize the initial states of both algorithms by performing a parameter search on the training split of the dataset. Specifically, the threshold of the Flooding algorithm is picked by testing a range of $[0.05, 2]$ in increments of 0.025 (Figure 13 (a)). In the case of the Active Contours algorithm, we search for optimal initial circle radius in the range of $[4, 24]$ pixels, with a step size of 1 pixel (Figure 14 (a)).



(a) (b)

Figure 13. Flooding algorithm applied on lung nodules in LIDC 2D. Panel (a) shows the results of a parameter search for the optimal flooding threshold performed on the training split of LIDC 2D. Panel (b) provides an example, where the threshold producing the best mean IoU is overshadowed by a hand-picked value.



(a) (b)

Figure 14. Active contours algorithm applied on lung nodules in LIDC 2D. Panel (a) shows the results of a parameter search for the optimal initial "snake" circle radius, performed on the training split of LIDC 2D. Panel (b) shows an example, where the starting circle radius picked by parameter search is considerably worse than a value picked specifically for this nodule.

Table 2 presents the IoU scores for both algorithms across the three dataset splits - train, validation and test. In our observations, the Flooding algorithm consistently outperforms Active Contours. Notably, both algorithms exhibited the weakest performance on the validation split and were the strongest on the test split, suggesting variation of segmentation difficulty across splits.

Algorithm	IoU		
	Train	Validation	Test
Flooding	0.343	0.297	0.386
Active Contours	0.28	0.252	0.312

Table 2. Classical algorithm performance across splits of LIDC 2D. Optimal threshold for Flooding and initial circle radius for Active Contours were picked by a parameter search on the training split.

Despite parameter optimization, the average segmentation performance measured is sub-optimal. Qualitative inspection, however, reveals a more nuanced behaviour. More often than not, the parameter picked by the search is best on average, but it is possible to find a more optimal value on a case-by-case basis (Figures 13 (b) and 14 (b)). The algorithms demonstrate varying responses to suboptimal parameter settings. The Flooding algorithm tends to undersegment when the threshold value is set too low, with the performance improving as the threshold increases. However, when the threshold is excessively high, the algorithm starts significantly over-segmenting, failing to differentiate between the nodule boundary and its surroundings. In contrast, the behaviour of Active Contours is less predictable under varying parameters. The quality of segmentation varies significantly, differing sporadically as the radius of the initial circle increases and behaves differently between one case and another.

In light of these results, we have discontinued our experiments with classical approaches and moved on to more complex segmentation methods.

4.3 One-Click Segmentation with Deep Convolutional Networks

We begin our assessment of deep learning techniques for interactive segmentation with an analysis of one-click IoU performance across our tumour datasets, as summarized in Table 3. The top-performing model varies across different forms of cancerous tissue. RITM excels in segmenting lung nodules and pancreatic tumours, while FocalClick and Unet++ perform best at kidney tumours and liver lesions accordingly. Notably, RITM architecture shows robustness, closely trailing the leading model in datasets where it was not the top performer and producing the best average IoU scores in the combined dataset. Overall, the findings suggest that the choice of the model for one-click segmentation with static binary disk guidance is flexible. While choosing the best-performing architecture per dataset is preferable, employing models other than RITM does not substantially compromise segmentation quality.

Model	LIDC 2D	KITS 2D	LITS 2D	MD PANC 2D	COMBINED 2D
U-Net++	0.638	0.799	**0.734**	0.647	0.688
FocalClick	0.668	**0.827**	0.634	0.680	0.679
RITM	**0.701**	0.816	0.696	**0.686**	**0.707**

Table 3. Deep learning model performance across our datasets. Mean IoU is measured after one guidance click. Click is encoded as a static radius binary disk for RITM and FocalClick, and as a static radius Gaussian disk in the case of U-Net++

4.4 Assessment of Dynamic Click Encoding

Building on top of our findings with models guided by static disks, we continue our experiments with the evaluation of the performance gained by utilizing a dynamic radius disk for guidance. The comparative results are summarized in Table 4. Both interactive segmentation architectures, RITM and FocalClick, effectively utilize the additional information provided by a disk, the size of which correlates with the size of the targeted cancerous object. We observe a significant improvement in performance across all datasets and both models. Specifically, the smallest performance gains are observed with RITM on the KITS 2D dataset, with an IoU increase of 0.031. FocalClick achieves the most substantial gains on the MD PANC 2D dataset, with an improvement of 0.135. While RITM maintains its dominance in lung nodule segmentation, neither architecture is universally superior, with FocalClick marginally outperforming RITM on the combined dataset. Examples of click prompting of static and dynamic encoding models and generated predictions can be observed in Figure 15.

Model	LIDC 2D	KITS 2D	LITS 2D	MD PANC 2D	COMBINED 2D
RITM	0.701 / **0.736**	0.816 / **0.847**	0.696 / **0.751**	0.686 / **0.788**	0.707 / **0.750**
FocalClick	0.668 / **0.708**	0.827 / **0.871**	0.634 / **0.708**	0.680 / **0.815**	0.679 / **0.753**

Table 4. Performance increase from employing dynamic radius disks for the guidance of interactive segmentation models. Models are evaluated based on IoU observed after one click, reported with static and dynamic metrics separated by a slash /.

4.5 Prediction Refinement Capability Evaluation

Although one-click performance is a critical metric, given that the first interaction tends to produce most of the segmentation, evaluation of the models' ability to refine existing prediction is equally crucial. Therefore, we evaluate the refinement performance on the lung nodules in the LIDC 2D dataset. We report the mean IoU values after the first five clicks in Table 5. While models are expected to perform well with minimal user

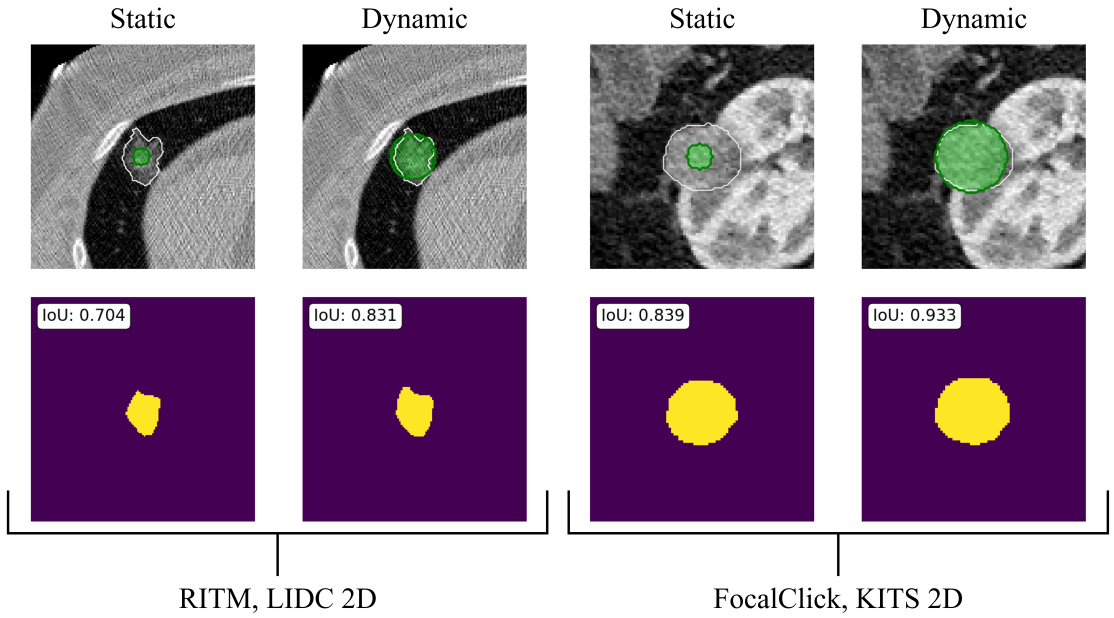


Figure 15. Guidance by static and dynamic radius disks in one-click scenarios. The top row illustrates user inputs with click disks depicted in green and the ground truth delineated in white. The bottom row presents the corresponding predictions.

interaction, understanding their performance with extensive user input is also valuable. Hence, we report Iterative IoU error after 20 clicks as well.

Model	IoU @ Clicks ↑					Iterative IoU Error @ 20 Clicks ↓
	1	2	3	4	5	
U-Net++	0.638	0.653	0.659	0.663	0.668	0.316
RITM	0.701	0.780	0.799	0.814	0.825	0.147
FocalClick	0.668	0.718	0.749	0.777	0.795	**0.129**
RITM_D	**0.736**	**0.801**	**0.816**	**0.828**	**0.834**	0.166
$\text{FocalClick}_\text{D}$	0.708	0.766	0.786	0.802	0.816	0.150

Table 5. Segmentation and iterative prediction refinement performance, LIDC 2D

Analyzing these results reveals several significant trends. First of all, we can see that our naive U-Net++ solution, while shown earlier (Table 3) to perform competitively with the other approaches in one-click scenarios, struggles to refine existing segmentation. Both static and dynamic versions of RITM consistently outperform their FocalClick counterparts in lung nodule segmentation, as also visible in Figure 16.

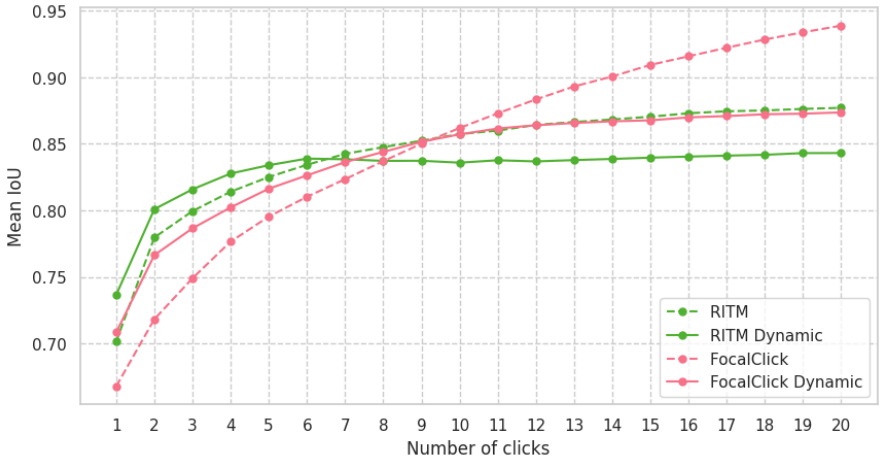


Figure 16. Observed IoU values given 20 subsequent clicks, LIDC 2D. Static disk models are represented by dashed, dynamic - solid curves

The figure highlights two major trends in the models' behaviour, with a shift around the 8th click. Initially, both versions of RITM surpass other methods, with the dynamic approach proving to be the most effective. It is evident that employing a dynamic radius disk to encode user input enhances the early performance of both RITM and FocalClick. As the number of clicks increases, the gains in IoU of dynamic models begin to plateau. Meanwhile, both static models continue to improve the segmentation quality and have a higher precision ceiling. Static RITM reaches its IoU ceiling at approximately 12 clicks, whereas FocalClick shows ongoing improvement in segmentation quality beyond this point. This observation aligns with the numbers presented in Table 5, which indicates the lowest Iterative IoU Error for FocalClick.

A qualitative inspection of simulated user input and the models' predictions after 20 clicks reveals the differences between the models (Figure 17). The static versions of both models tend to over-segment, requiring more simulated negative clicks than their dynamic counterparts. A closer examination of the dynamic models' inputs reveals why they reach their IoU ceiling earlier. Despite 20 clicks being simulated, significantly fewer disks are apparent, implying their overlapping placement. That suggests the models are under-segmenting, requiring multiple positive clicks at the same region which do not provide additional information for the models.

4.6 Examining Generalization Across Datasets

The primary objective of interactive segmentation models differs from semantic segmentation. Semantic segmentation models lack user guidance, making them rely entirely on the content of the image. In contrast, interactive models make use of user input, therefore

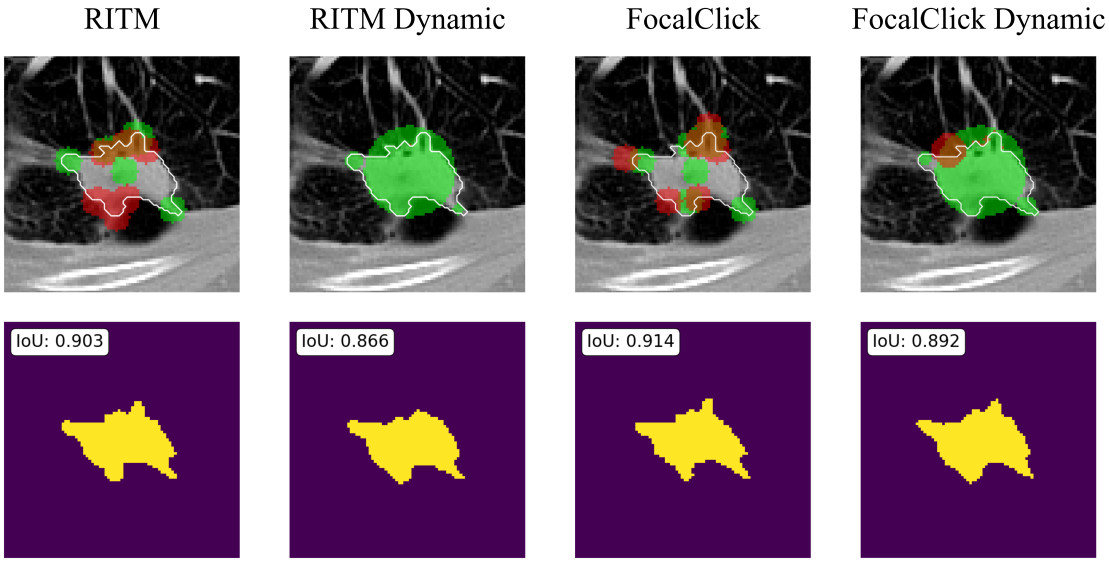


Figure 17. Click arrangement and segmentations predicted after 20 iterations of simulated user input. The top row represents user input, with green and red disks representing positive and negative clicks, accordingly. The bottom row shows produced predictions along with measured IoU values

reducing reliance on imaging. In our case, the model's task is to accurately segment masses of tissue, based on their topology, boundaries and the guidance of user input. Given this objective, it is critical to evaluate the performance of these models on unfamiliar data. Our analysis focuses on assessing how well both static and dynamic models generalize across different CT datasets. Furthermore, we explore the impact of training on a combined dataset comprising various tissue types to determine whether exposure to more diverse structures during training enhances the model's ability to generalize.

Our investigation into model generalization assesses the performance of static and dynamic models on datasets different from their training, in a one-click scenario. The results, detailed in Table 6, indicate that while training on diverse CT datasets enables models to generalize relatively effectively, exposing a model to the specific types of tumours it will encounter during its operational use is often most beneficial.

In comparison to other configurations, the model trained on the combined dataset demonstrated significantly stronger performance than its counterparts. This model showed notably strong performance in segmenting kidney and pancreatic tumours, particularly excelling in the MD PANC 2D dataset, where it outperformed both static and dynamic models originally trained on pancreatic tumours only. For the KITS 2D dataset, the model leveraging dynamic click interactions performed best, surpassing other dynamic approaches, with the static model closely following the performance of a model trained exclusively on kidney tumours.

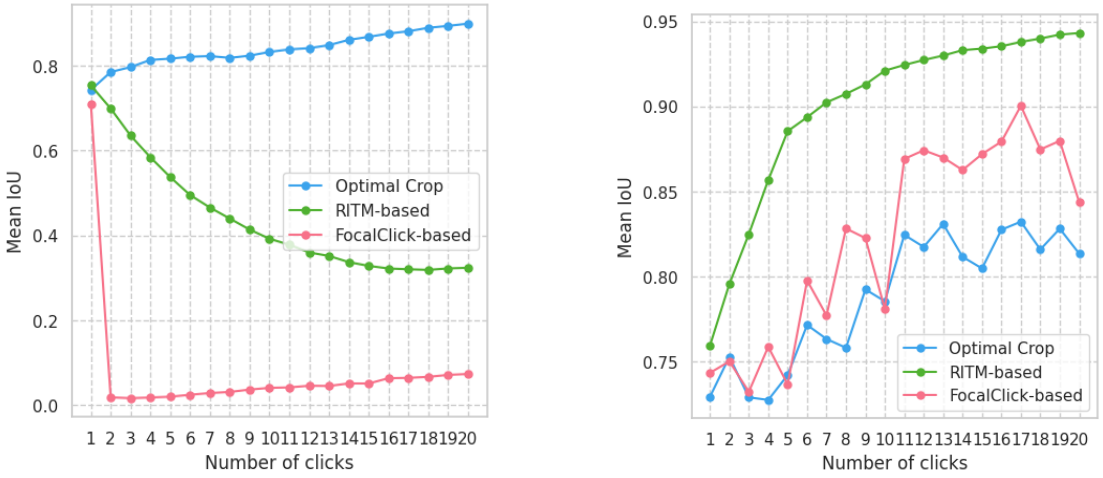
Training Data	Evaluation Data				
	LIDC 2D	KITS 2D	LITS 2D	MD PANC 2D	COMBINED 2D
RITM					
LIDC 2D	**0.701**	0.422	0.187	0.525	0.459
KITS 2D	0.617	**0.816**	0.395	0.511	0.585
LITS 2D	0.385	0.515	**0.696**	0.506	0.527
MD PANC 2D	0.458	0.620	0.396	0.686	0.546
COMBINED 2D	0.675	0.791	0.662	**0.698**	**0.707**
RITM + Dynamic click encoding					
LIDC 2D	**0.736**	0.746	0.430	0.662	0.648
KITS 2D	0.437	0.847	0.485	0.776	0.635
LITS 2D	0.513	0.713	**0.751**	0.756	0.684
MD PANC 2D	0.470	0.773	0.527	0.788	0.639
COMBINED 2D	0.643	**0.851**	0.702	**0.806**	**0.750**

Table 6. Generalization across datasets, mean IoU after one click.

In other scenarios, however, specializing the model on a single dataset proved more advantageous. Overall, models equipped with dynamic click encoding consistently exhibited superior performance in segmenting unfamiliar tissue types compared to their static counterparts. This improvement can likely be attributed to the additional, albeit imprecise, boundary information provided by dynamic clicks, which reduce the need for an in-depth understanding of the object's borders and topology.

4.7 Augmentation Strategy Evaluation

While the behaviour displayed by the trained interactive models aligns well with the tendencies described by their authors [8, 9], achieving these results required modifications from the models' initial configurations, particularly for the adaptation to CT images and segmenting of tiny objects. One of the most notable modifications was the introduction of the Optimal Crop training strategy for the training of FocalClick. Given our research focus on lung nodule segmentation, naturally, most of our experiments were performed on LIDC 2D and LIDC $2D_S$ datasets. FocalClick's original augmentation strategy yielded unacceptable refinement results. While borrowing the RITM-based augmentation strategy showed an improvement, the model still failed to improve on the initially predicted augmentations. Applying the Optimal Crop augmentation strategy addressed this issue, enabling the model to not only produce an initial segmentation but also successfully refine it with subsequent clicks. The performance comparison of the 3 strategies can be seen in Figure 18.



(a) Evaluation on LIDC $2D_S$

(b) Evaluation on KITS $2D_S$

Figure 18. Ablation study on effects of different kinds of augmentation strategies on the performance of FocalClick equipped with static radius disks. Panels (a) and (b) display initial segmentation and mask refinement performance on LIDC $2D_S$ and KITS $2D_S$, respectively.

Despite these improvements on LIDC $2D_S$, when the three augmentation strategies were evaluated on kidney tumours, as shown in the panel (b) of Figure 18, the RITM-based strategy outperformed the others. FocalClick-based and Optimal Crop strategies, while improving on the initial segmentation, demonstrated unsatisfactory, fluctuating behaviour. Similar results were observed with LITS $2D_S$ and MD PANC $2D_S$, with different strategies prevailing on each dataset, highlighting the sensitivity of FocalClick to relatively minor training adjustments.

5 Discussion

This thesis has explored the application of interactive segmentation models, specifically RITM and FocalClick, for identifying various tumours from CT images. The results strongly support the potential application of these models in clinical settings. It is important to acknowledge, however, that substantial modifications were necessary to adapt these models to the CT modality. Typically, interactive (and other) segmentation models are designed with natural images in mind. Such images often focus on the subject, resulting in the segmentation targets occupying a significant portion of the image. In contrast, the segmentation of tumours presents a unique challenge: the scans display cross-sections, where tumours are tiny objects within a larger area.

Initially, applying these models for tumour segmentation exposed various bugs, as well as intentional design decisions, which were irrelevant when segmenting bigger objects. Consequently, adjustments to training and inference procedures, such as the introduction of the Optimal Crop augmentation strategy for FocalClick, were a necessity rather than merely an improvement. It is fair to assume that employing these models for other modalities would similarly require additional changes.

Narrowing our focus to tumour segmentation in tomography scans has also allowed for more specific assumptions about the targets. In natural images, the diversity of potential objects constrains the design of segmentation models. For instance, let's assume we're trying to segment a bicycle from a typical image. The target here is complex, containing various protrusions, hollow wheels, etc. In contrast, tumours generally are solid lumps of tissue. This characteristic has enabled the implementation of strategies such as dynamic radius disk encoding. In this approach, it is assumed that enlarging the disk will likely continue to highlight the tumour without sending mixed signals about which part of the object should or shouldn't be segmented. Our experiments have shown that incorporating this inductive bias into the interactive segmentation models significantly enhanced the segmentation accuracy, demonstrating the value of tailoring tooling strategies to the specific characteristics of the objects of interest.

Overall, the results of our experiments suggest that the capabilities of these models could significantly expedite the examination of CT scans in clinical settings. While the actual increase in efficiency would depend on processes and practices currently in place, we were able to show that interactive segmentation models like RITM and FocalClick are capable of serving as tools for tumour segmentation. Facilitating such deep learning tools would allow for the reallocation of more human resources towards other stages of the diagnostic process. The choice of user input encoding, whether dynamic or static radius disks, should be tailored to specific clinical scenarios. For instance, dynamic disk encoding is preferable in situations where it is critical to minimise user interaction, while static disks are better suited for scenarios where optimal segmentation quality is necessary, regardless of user input required for refinement.

Despite the promising results, it is important to acknowledge several limitations. Our

models displayed varying performance across different types of malignancies and, more importantly, required significant tuning to adapt to specific scenarios. For instance, while the introduction of the Optimal Crop augmentation strategy allowed for the refinement of lung nodules of LIDC 2D with FocalClick, the effect did not translate well to other datasets. Furthermore, the models' performance was highly sensitive to even minor modifications during training and inference. Maintaining a consistent correlation between one-click segmentation and refinement performance remains a significant challenge.

While our research makes a strong case for interactive segmentation models application in diagnostic pipelines, there is vast potential for further development. In future work, we would like to explore the integration of 3D scan volume data to enhance segmentation accuracy. This might involve using adjacent CT slices to possibly improve predictions of 2D segmentation, or developing models that operate on entire volumes and generate predictions in 3D space. Additionally, a combination of both static and dynamic radius disk click encoding should be explored, possibly allowing for optimal initial predictions without the reduced refinement potential.

Classical segmentation methods, while observed to be lackluster as a universal one-click solution, have also shown potential on a case-by-case basis. While they lack segmentation refinement capability, these algorithms could be utilized in one-click scenarios and we would like to explore adapting dynamic radius disk click encoding to provide instance-specific starting conditions for these algorithms.

6 Conclusion

As we progress deeper into the 21st century, healthcare systems and, in particular, radiology services face increasing pressures from an ageing population. These challenges, compounded by an understaffed healthcare workforce, result in a need for technological advancements in diagnostics. Our research aims to address this necessity by exploring the potential of using interactive segmentation deep learning models to streamline the diagnostic process.

Through our experiments, we have explored the potential of employing interactive deep-learning models as tools for lesion segmentation in CT images. Our findings have demonstrated that they could both effectively predict and refine tumour masks. Moreover, we have discovered that interactive segmentation models can utilize additional input data encoded through dynamic radius disks, increasing the segmentation quality across the initial user clicks. These results signify the potential of applying interactive segmentation models in clinical settings and increasing the efficiency of diagnostic pipelines.

Ultimately, by advancing these technologies and assessing their feasibility, we aim to support radiologists and contribute to faster and more accurate diagnostics. That is crucial in improving patient outcomes in the face of ever-growing healthcare demands.

7 Acknowledgments

The author of this thesis is a member of the Biomedical Computer Vision Lab and was supported by the lab throughout this work. Special thanks go to the supervisor, Dmytro Fishman, for his continuous support, insightful feedback, and patience. His belief in the author and the opportunities he provided were greatly appreciated. The author also wishes to thank the other members of the research group for their helpful insights and different perspectives, which were crucial at times.

References

[1] Giles Maskell. Why does demand for medical imaging keep rising?, 2022.

[2] Chih-Hsiang Ko, Li-Nien Chien, Yu-Ting Chiu, Hsian-He Hsu, Ho-Fai Wong, and Wing P Chan. Demands for medical imaging and workforce size: A nationwide population-based study, 2000–2020. *European Journal of Radiology*, 172:111330, 2024.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[4] 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.

[5] Giovanni E Cacciamani, Daniel I Sanford, Timothy N Chu, Masatomo Kaneko, Andre L De Castro Abreu, Vinay Duddalwar, and Inderbir S Gill. Is artificial intelligence replacing our radiology stars? not yet! *European Urology Open Science*, 48:14–16, 2023.

[6] Sharyn LS MacDonald, Ian A Cowan, Richard A Floyd, and Rob Graham. Measuring and managing radiologist workload: A method for quantifying radiologist activities and calculating the full-time equivalents required to operate a service. *Journal of medical imaging and radiation oncology*, 57(5):551–557, 2013.

[7] Youbao Tang, Ke Yan, Jing Xiao, and Ronald M Summers. One click lesion recist measurement and segmentation on ct scans. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part IV 23*, pages 573–583. Springer, 2020.

[8] Konstantin Sofiiuk, Ilya A Petrov, and Anton Konushin. Reviving iterative training with mask guidance for interactive segmentation. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 3141–3145. IEEE, 2022.

[9] Xi Chen, Zhiyan Zhao, Yilei Zhang, Manni Duan, Donglian Qi, and Hengshuang Zhao. Focalclick: Towards practical interactive image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1300–1309, 2022.

[10] Shah Hussain, Iqra Mubeen, Niamat Ullah, Syed Shahab Ud Din Shah, Bakhtawar Abduljalil Khan, Muhammad Zahoor, Riaz Ullah, Farhat Ali Khan,

Mujeeb A Sultan, et al. Modern diagnostic imaging technique applications and risk factors in the medical field: a review. *BioMed research international*, 2022, 2022.

[11] Reem Nooreldeen and Horacio Bach. Current and future development in lung cancer diagnosis. *International journal of molecular sciences*, 22(16):8661, 2021.

[12] Fabian Isensee, Paul F Jaeger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods*, 18(2):203–211, 2021.

[13] Godfrey N Hounsfield. Computed medical imaging. *Science*, 210(4465):22–28, 1980.

[14] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

[15] Laurent D Cohen. On active contour models and balloons. *CVGIP: Image understanding*, 53(2):211–218, 1991.

[16] Chan-Pang Kuok, Tai-Hua Yang, Bo-Siang Tsai, I-Ming Jou, Ming-Huwi Horng, Fong-Chin Su, and Yung-Nien Sun. Segmentation of finger tendon and synovial sheath in ultrasound image using deep convolutional neural network. *Biomedical engineering online*, 19:1–25, 2020.

[17] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014.

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

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[20] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE transactions on medical imaging*, 39(6):1856–1867, 2019.

[21] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Artificial intelligence and statistics*, pages 562–570. Pmlr, 2015.

[22] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020.

[23] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7262–7272, 2021.

[24] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[25] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[26] Qin Liu, Zhenlin Xu, Gedas Bertasius, and Marc Niethammer. Simpleclick: Interactive image segmentation with simple vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22290–22300, 2023.

[27] Shoukun Sun, Min Xian, Fei Xu, Luca Capriotti, and Tiankai Yao. Cfr-icl: Cascade-forward refinement with iterative click loss for interactive image segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 5017–5024, 2024.

[28] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[29] S. G. Armato III, G. McLennan, L. Bidaut, M. F. McNitt-Gray, C. R. Meyer, A. P. Reeves, B. Zhao, D. R. Aberle, C. I. Henschke, E. A. Hoffman, E. A. Kazerooni, H. MacMahon, E. J. R. Van Beek, D. Yankelevitz, A. M. Biancardi, P. H. Bland, M. S. Brown, R. M. Engelmann, G. E. Laderach, D. Max, R. C. Pais, D. P. Y. Qing, R. Y. Roberts, A. R. Smith, A. Starkey, P. Batra, P. Caligiuri, A. Farooqi, G. W. Gladish, C. M. Jude, R. F. Munden, I. Petkovska, L. E. Quint, L. H. Schwartz, B. Sundaram, L. E. Dodd, C. Fenimore, D. Gur, N. Petrick, J. Freymann, J. Kirby,

B. Hughes, A. V. Casteele, S. Gupte, M. Sallam, M. D. Heath, M. H. Kuhn, E. Dharaiya, R. Burns, D. S. Fryd, M. Salganicoff, V. Anand, U. Shreter, S. Vastagh, B. Y. Croft, and L. P. Clarke. Data from lidc-idri. https://doi.org/10.7937/K9/TCIA.2015.LO9QL9SX, 2015.

[30] Nicholas Heller, Fabian Isensee, Dasha Trofimova, Resha Tejpaul, Zhongchen Zhao, Huai Chen, Lisheng Wang, Alex Golts, Daniel Khapun, Daniel Shats, Yoel Shoshan, Flora Gilboa-Solomon, Yasmeen George, Xi Yang, Jianpeng Zhang, Jing Zhang, Yong Xia, Mengran Wu, Zhiyang Liu, Ed Walczak, Sean McSweeney, Ranveer Vasdev, Chris Hornung, Rafat Solaiman, Jamee Schoephoerster, Bailey Abernathy, David Wu, Safa Abdulkadir, Ben Byun, Justice Spriggs, Griffin Struyk, Alexandra Austin, Ben Simpson, Michael Hagstrom, Sierra Virnig, John French, Nitin Venkatesh, Sarah Chan, Keenan Moore, Anna Jacobsen, Susan Austin, Mark Austin, Subodh Regmi, Nikolaos Papanikolopoulos, and Christopher Weight. The kits21 challenge: Automatic segmentation of kidneys, renal tumors, and renal cysts in corticomedullary-phase ct, 2023.

[31] Patrick Christ. Lits – liver tumor segmentation challenge (lits17).

[32] Michela Antonelli, Annika Reinke, Spyridon Bakas, Keyvan Farahani, Annette Kopp-Schneider, Bennett A Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M Summers, et al. The medical segmentation decathlon. *Nature communications*, 13(1):4128, 2022.

[33] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[34] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[35] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019.

[36] Grammarly. https://www.grammarly.com/.

[37] Chatgpt plus. https://openai.com/blog/chatgpt-plus.

[38] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[39] University of Tartu. Ut rocket, 2018.

[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[41] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[42] Omry Yadan. Hydra - a framework for elegantly configuring complex applications. Github, 2019.

[43] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com.

Appendix

I. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Donatas Vaiciukevičius**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Every Click Counts: Deep Learning Models for Interactive Segmentation in Biomedical Imaging,

   supervised by Dmytro Fishman.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Donatas Vaiciukevičius
15/05/2024