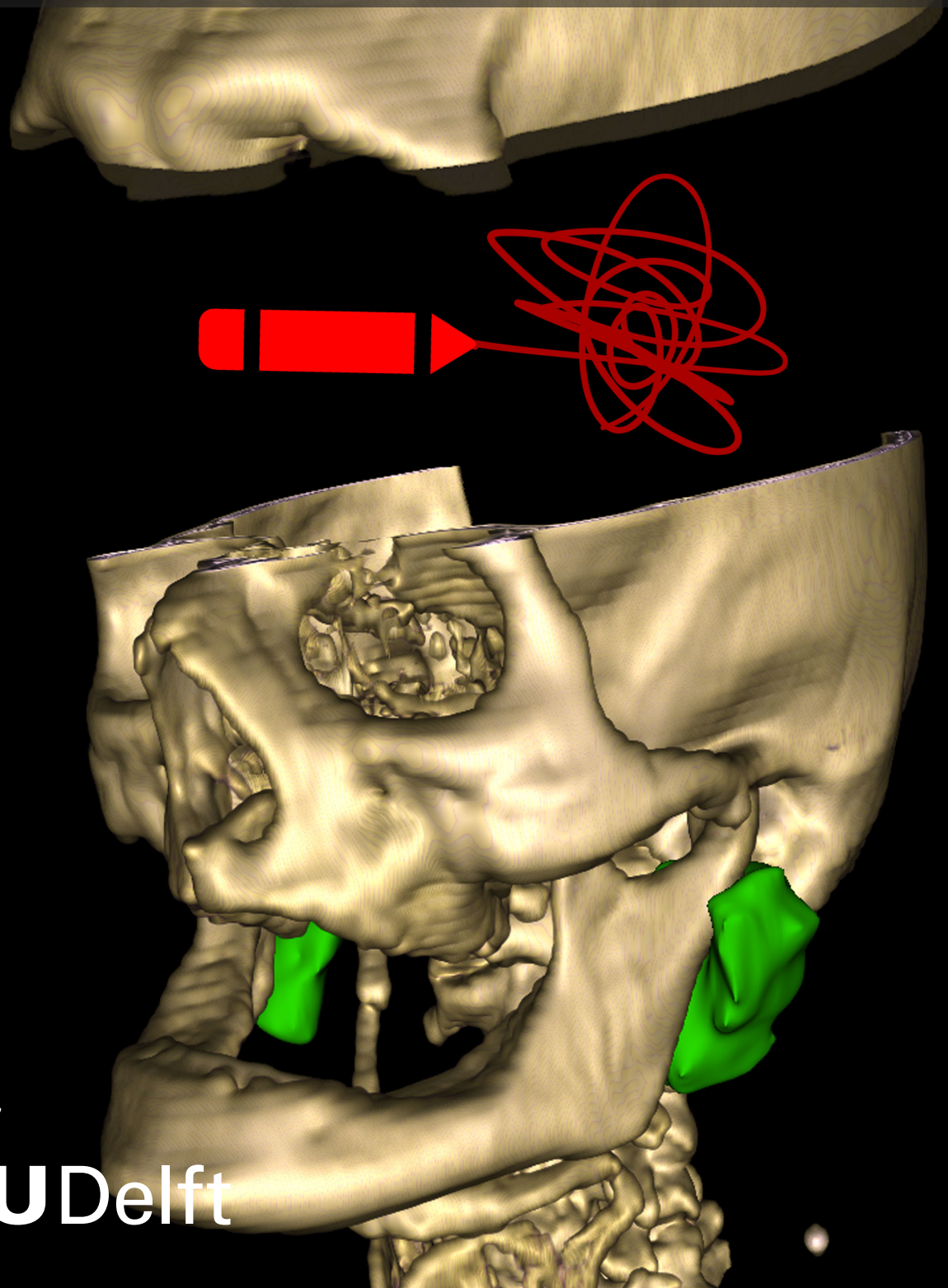
Master Thesis

Bridging Train–Test Domain Shift with Simulated Scribbles for Interactive Segmentation Refinement

Simon Tulling



Delft University of Technology

TU Delft

Master Thesis

Bridging Train–Test Domain Shift with Simulated Scribbles for Interactive Segmentation Refinement

by

Simon Tulling

Student Name	Student Number
Simon Tulling	4694341

Associated People	
Thesis advisor	Boudewijn Lelieveldt
Daily supervisor	Klaus Hildebrandt
External supervisor	Prerak Mody
Committee core member	Pradeep Murukannaiah

Style: TU Delft Report Style, with modifications by Daan Zwaneveld

TU Delft

Preface

This thesis took far longer to complete than it probably should have, and I am very grateful to the people who continued to support me throughout the process.

I would especially like to thank my family, Prerak Mody, Elmar Eisemann, Klaus Hildebrand, and Boudewijn Lelieveldt for their patience, encouragement, and for sticking with me through all of it. I also am grateful for my former manager Clint Kenswiel for giving me the support and time to work on this project.

I look back fondly upon the time I had with my colleagues at LKEB during the start of my thesis. Everyones support made it possible for me to finally bring this work to completion.

Simon Tulling
Delft, September 2026

Abstract

Accurate segmentation of organs at risk in head-and-neck radiotherapy is important for limiting radiation-induced damage to healthy tissue, yet fully automatic segmentation methods still require manual correction in difficult cases. Interactive refinement offers a practical alternative, but many refinement systems are trained with simulated point annotations while being used in practice with human scribbles. This creates a train-test mismatch in the user-guidance channels, which we study in this thesis as annotation-shift. To address this problem, we propose a training pipeline for interactive segmentation refinement based on simulated human-like scribbles, combined with annotation-aware supervision, distance-map encodings, and explicit manipulation of the initial segmentation. The method is evaluated on head-and-neck CT data for parotid gland segmentation and compared with both a standard point-based baseline and a balanced point-based baseline. The results show that outperforms the original point-based refinement strategy, with the clearest gains appearing in the local evaluation near the annotated region. At the same time, the smaller gap between scribbles and balanced points indicates that refinement quality depends not only on annotation geometry, but also on the amount and spatial distribution of corrective information. These findings provide evidence that annotation-shift affects interactive refinement in interactive medical image segmentation and that simulated scribbles provide a useful step toward refinement systems that are better aligned with real clinical annotation practice.

Contents

Master Thesis

Simon Tulling

Friday 4th September, 2026

1 Introduction

1.1. Background

1.1.1. Radiotherapy and the Role of OARs

Radiotherapy remains one of the most effective and widely used modalities for cancer treatment. By directing high-energy ionizing radiation at malignant tissue, clinicians aim to damage the DNA of tumor cells, thereby inhibiting their ability to proliferate and ultimately leading to cell death. The procedure often involves delivering radiation from multiple angles so that the tumor receives a concentrated dose while minimizing exposure to surrounding structures.

Despite its efficacy, radiotherapy carries inherent risks. Radiation cannot be perfectly confined to the tumor; neighboring healthy tissues inevitably receive some exposure. These tissues, commonly referred to as organs at risk (OARs), are of critical importance in treatment planning. OARs are anatomical structures whose impairment would significantly affect the quality of life—examples in the head and neck region include salivary glands, spinal cord, optic nerves, and brain stem [1, 2]. Damage to such structures can result in side effects ranging from temporary discomfort, such as xerostomia (dry mouth), to irreversible organ dysfunction or, in severe cases, life-threatening complications [2].

Consequently, a precise delineation of both the target tumor volume and the nearby OARs is essential. This process, called segmentation, enables clinicians to design radiation beams that deliver a therapeutic dose to the tumor while sparing surrounding tissues as much as possible. Historically, segmentation has been performed manually by radiation oncologists or trained technicians. While manual delineation can be accurate, it is highly time-consuming and subject to both inter- and intra-observer variability [1]. Different clinicians may delineate the same organ in subtly different ways, introducing inconsistencies into treatment plans. The need for efficient, reliable, and reproducible segmentation motivates the exploration of computational methods [3].

1.1.2. Computed Tomography in Radiotherapy Planning

Among imaging modalities used in oncology, computed tomography (CT) plays a central role in radiotherapy planning. CT scanning provides volumetric reconstructions of patient anatomy by acquiring a series of X-ray projections around the body and reconstructing them into cross-sectional slices. Each slice typically spans a few millimeters in thickness, and together these slices form a three-dimensional representation of the patient.

CT imaging is particularly well suited for treatment planning for several reasons. First, it offers excellent contrast for bone structures, which facilitates patient positioning and immobilization during radiation delivery. The ability to visualize skeletal anatomy with high

precision allows clinicians to align the patient consistently across treatment sessions. Second, CT data are expressed in standardized Hounsfield units, which directly relate to tissue density and provide the information required for accurate radiation dose calculations. However, CT scans are less effective at distinguishing between soft tissues compared to modalities such as magnetic resonance imaging (MRI). For head and neck radiotherapy, where many OARs are small and located in close proximity to the tumor, this limitation poses a significant challenge [1, 4, 5]. Structures such as salivary glands and lymph nodes often appear with poor contrast, making precise delineation difficult even for experienced clinicians. These imaging characteristics underscore the importance of advanced segmentation techniques capable of handling ambiguous or noisy boundaries.

1.1.3. Segmentation

Segmentation of organs-at-risk (OARs) in medical imaging may be performed manually, automatically, or through interactive refinement. Manual delineation by experts remains the clinical reference standard, but it is resource-intensive and introduces variability that can propagate to treatment planning and affect reproducibility in radiotherapy [1, 2].

Observer variability. Two main types of variability are commonly recognized. *Inter-observer variability* refers to differences between clinicians; for example, two radiation oncologists may delineate the same OAR boundary slightly differently on a CT scan due to subjective judgment, training background, or ambiguity in the image. *Intra-observer variability* refers to differences in the annotations made by the same clinician at different times, caused by factors such as fatigue, memory, or interpretation inconsistencies.

Computational approaches. Early computational methods aimed to reduce the burden and inconsistency of manual contouring and included thresholding, region growing, graph-based methods such as interactive graph cuts and GrabCut [6, 7], Markov random fields and conditional random fields (CRFs) [8, 9, 10], and atlas-based registration [1]. While innovative, these approaches were often brittle when faced with patient variability or imaging artifacts.

The rise of deep learning, particularly fully convolutional networks (FCNs), transformed the field [11, 12]. Fully convolutional architectures such as FCN, U-Net, 3D U-Net, and V-Net have driven substantial performance gains in biomedical segmentation, including head-and-neck CT [3, 4, 11, 12, 13, 14, 15, 16, 17].

Limitations of fully automatic systems. Despite progress, purely automatic systems may fail on difficult cases or out-of-distribution data. Detecting local segmentation failures is therefore crucial [18]. In radiotherapy, even small OAR delineation errors can be clinically significant [1], and automatic outputs may still require manual refinement [19, 20].

Interactive refinement with user guidance. Consequently, interactive methods incorporate sparse user guidance (e.g., scribbles, clicks) to correct local errors while retaining efficiency. Representative deep interactive frameworks include DeepIGeoS and its variants, which combine geodesic distances from user inputs with CNN-based refinement [19, 21]; minimally interactive segmentation systems such as Slic-Seg and MIDeepSeg [22, 23]; uncertainty-guided refinement to focus interactions where they are most beneficial [20, 24, 25]; and general interactive selection models from computer vision adapted to medical contexts [26, 27]. Efficient computation of geodesic transforms further enables fast propagation of user input in 3D [28, 29, 30]. These approaches reduce annotation burden and improve reliability, but their efficacy depends

critically on how user input is represented and how closely training-time simulations match test-time human interactions.

1.1.4. Supervised Machine Learning

In *supervised* machine learning, a model is trained to predict a target output from an input using a dataset of labeled examples. Formally, given training pairs (x_i, y_i) with inputs x_i and ground-truth labels y_i, learning amounts to fitting a parameterized function f_θ by minimizing a loss function $\mathcal{L}$.

$$\theta^\star = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\big(f_\theta(x_i),\, y_i\big), \tag{1.1}$$

A central goal of supervised learning is *generalization*: achieving low error on unseen examples beyond the training set. In practice, generalization is promoted through appropriate model capacity, regularization, data augmentation, and careful validation protocols. These choices are especially important in medical imaging, where labeled data are expensive and class imbalance is common [3].

Supervision for interactive segmentation. Interactive segmentation can be framed as supervised learning by augmenting the input with user guidance channels and an initial segmentation. Here, each training example consists of the image together with an encoding of the interaction (e.g., clicks, scribbles, distance transforms, probability masks) and a segmentation, and the target is the refined segmentation after applying that guidance. Training requires defining (or simulating) representative interaction signals so that the supervision matches the refinement behavior expected at test time. Common approaches include sampling corrective annotations from current prediction errors and emphasizing hard/uncertain regions to show to the user for annotations [24, 25]. Another important design choice is the *encoding* of user input: robust encodings aim to preserve the semantic intent of the interaction while remaining stable across different geometries (e.g., points versus strokes) [19, 26].

1.1.5. Domain Shift

A recurring assumption in supervised learning is that training data are representative of the data encountered at deployment. When this assumption is violated, performance can degrade due to *domain shift* (also called dataset shift or distribution shift): the input-output relationship learned from the training distribution does not transfer cleanly to the test or deployment distribution [31]. In radiotherapy segmentation, domain shift is not an edge case but a practical reality, as imaging protocols, scanners, reconstruction settings, and patient cohorts often vary across sites and over time. It has been shown that variations in medical scanner manufacturers have significant negative effect on the accuracy of a deep learning model for medical imaging if it has been trained on a different one [32].

Why domain shift matters for interactive refinement. In interactive segmentation, the model consumes both an image and user guidance, otherwise known as annotations. Domain shift therefore affects refinement in two coupled ways:

1. **Image shift:** the refinement network may rely on image cues (edges, intensities, texture) whose statistics change at deployment, reducing its ability to propagate corrections reliably.
2. **Annotation shift:** the distribution of user interactions at deployment can differ substantially from the interactions seen during training, both in *what* is annotated and *how* it is

annotated. This is especially pronounced when training relies on simulated prompts (e.g., synthetic clicks derived from ground truth) that do not match clinician behavior.

As a result, a system can appear robust when evaluated on in-domain test sets yet require substantially more user effort under a new site or protocol.

1.2. Scope of this thesis

Interactive segmentation is an appealing compromise between fully manual contouring and fully automatic pipelines: an algorithm produces an initial segmentation and a clinician provides sparse corrections to fix local failures. In radiotherapy planning, however, even small residual errors in OAR boundaries can be clinically relevant. This makes the reliability of the refinement step crucial.

A common but under-discussed issue is that many interactive refinement networks are trained with *simulated point prompts* but deployed with *human scribbles*. This creates a domain-shift in the user-guidance channels, which we call annotation-shift. We hypothesize that this mismatch contributes to suboptimal refinement and propose training procedures that explicitly simulate *scribbles* to better match deployment conditions. In this thesis, we study this mismatch between training and deployment interaction prompts explicitly and propose training procedures that better align training-time interactions with deployment-time clinician behavior.

1.2.1. Scribbles

The literature often uses the term *scribbles* to denote sparse user guidance; however, the underlying annotation geometry varies across works (freehand lines, dense strokes, scattered points). In order to minimize ambiguity, we adopt precise terminology:

- **Scribbles**: annotations that form freeform, connected line segments drawn by a user or generated by an algorithm intended to mimic user strokes. We distinguish *human-generated scribbles* (drawn by annotators) from *simulated scribbles* (algorithmically generated).
- **Points**: sets of isolated pixels/voxels sampled without enforcing connectivity.

Scribbles naturally encode boundary intent and regional context along a path, whereas isolated points provide sparser, less structured cues. Many interactive pipelines transform user inputs into distance maps or guidance channels; geodesic distances are particularly effective at capturing topology-aware proximity to annotations [19, 28, 29].

1.2.2. Thesis Contributions

This thesis proposes and evaluates a number of solutions for the *annotation-shift problem* in interactive OAR segmentation for radiotherapy planning. We test the central hypothesis that aligning training-time interactions with deployment-time human scribbles reduces annotation-shift and yields more reliable, clinically relevant refinements. Our main contributions are:

1. **Human-like scribble simulation.** We design a training-time scribble generator that produces freeform, connected strokes mimicking typical human corrections near uncertain or mis-segmented regions as they occur in practice. The generator incorporates topology-aware constraints using geodesic guidance, that aim to reduce the difference between training and deployment scribbles.
2. **Constraint-aware supervision.** We introduce loss functions and preprocessing strategies that treat annotated voxels as hard or high-confidence constraints. This has the potential to improves local fidelity at user-marked locations while preserving global consistency via the network and optional regularizers.

3. **Scribble-to-network encodings.** We systematically compare encodings of scribbles (e.g., Euclidean vs. geodesic distance transforms, multi-class guidance channels, uncertainty-targeted masks) and analyze their effect on refinement performance.

Chapter 2 reviews related work with a specific focus on integration of user interaction into deep-learning based segmentation frameworks. Chapter 3 presents the developed methodology for minimizing annotation mismatch. Chapter 4 describes the experimental setup to evaluate the proposed pipeline, with chapter 5 reporting these results.

Together, these contributions provide the first systematic investigation of scribble-based simulation for interactive radiotherapy segmentation, directly addressing the mismatch between training-time point annotations and deployment-time human scribbles.

2

Related Work

2.1. Automatic Segmentation using Deep Learning

From its earliest applications on MNIST [33], convolutional neural networks (CNNs) [34] such as LeNet-5 [34] and AlexNet [35] demonstrated strong performance for image classification. Subsequent advances extended CNNs to object detection [36] and semantic segmentation [12]. In semantic segmentation, a model predicts a label per pixel/voxel, enabling dense delineation of anatomical structures. Among many architectures, U-Net [13] has been particularly influential in biomedical imaging.

2.1.1. U-Net

U-Net [13] introduced a symmetric encoder–decoder with skip connections that fuse context from the contracting path with spatial detail on the expanding path. This design supports precise localization across a range of structure sizes while training effectively with limited annotated data, which is common in medical imaging; see Fig. 2.1 for the canonical diagram.

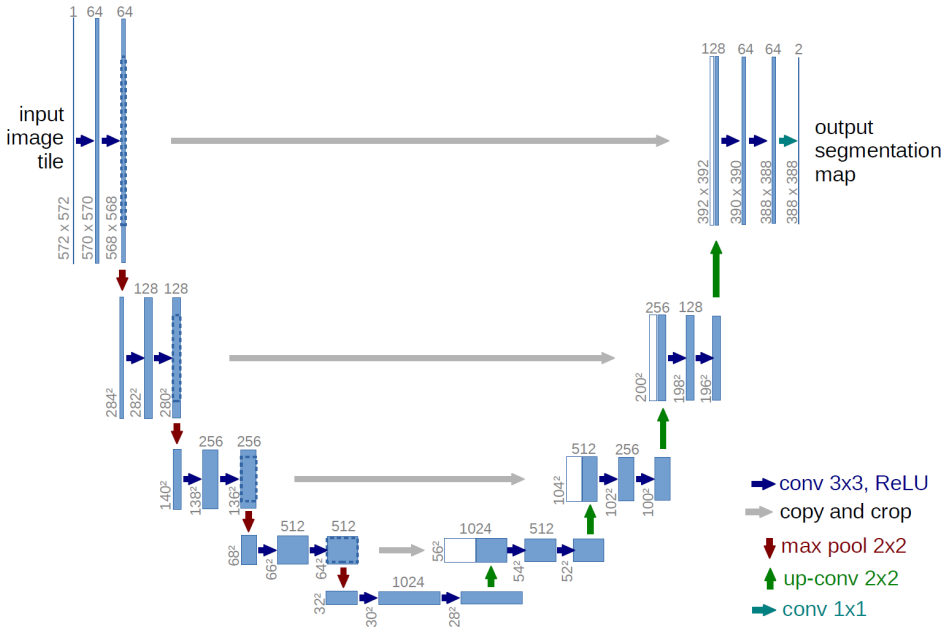


Figure 2.1: The U-Net architecture (adapted from [13]).

U-Net variants

Numerous variants adapt U-Net to specific modalities and constraints. For volumetric CT, 3D U-Net replaces 2D with 3D convolutions to exploit through-slice context [14, 15]. However, repeated down/upsampling can impair the delineation of small, low-contrast soft-tissue organs typical of head-and-neck CT due to class imbalance and loss of fine detail. To address this, FocusNet leverages dilated (atrous) convolutions to enlarge the receptive field without excessive resolution loss, targeting simultaneous large/small organ segmentation [4, 17].

Bayesian approximations have also been explored. For example, FlipOut variational perturbations [37] have been used to approximate Bayesian neural networks (BNNs) for predictive uncertainty estimation in head-and-neck contouring [5]. Such uncertainty estimates can support downstream tasks such as quality control or interaction guidance (Section 2.1.1).

Uncertainty for Segmentation

Uncertainty has been integrated into interactive refinement primarily to *guide* users toward regions likely to benefit from annotation. Wang et al. [38] employ a Bayesian U-Net to highlight uncertain slices in fetal brain MRI, while Zheng et al. [25] formulate interactive segmentation as continual learning with uncertainty-aware interaction policies. These approaches assume a correlation between uncertainty and error; however, uncertainty quantifies model confidence and does not guarantee misclassification localization [39].

In our preliminary experiments, slice selection based on predicted uncertainty did not yield consistent Dice improvements over random selection. We also evaluated feeding uncertainty as an additional input channel to the refinement model and observed no clear gains. Consequently, we focus the remainder of this work on the *annotation modality* (scribbles vs. points) and its encoding, which directly affects the supervision signal presented during interactive refinement.

2.2. Refinement Networks

Deep interactive frameworks refine an automatic base segmentation using sparse user inputs. DeepIGeoS [19] introduced a two-stage pipeline: P-Net produces the base segmentation, and R-Net refines it conditioned on user guidance. The encoding of user input follows the "distance-map" paradigm popularized in DIOS [26, 40], where foreground/background interactions (points or strokes) are transformed into auxiliary channels concatenated with the image and/or base mask (Fig. 2.2).

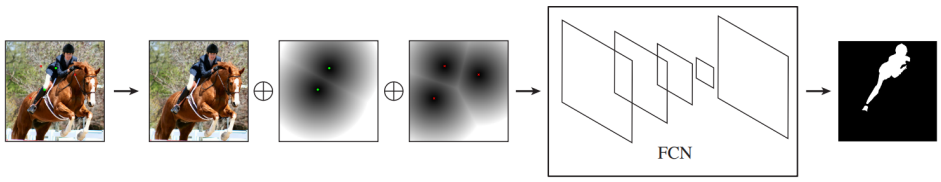


Figure 2.2: Example of distance-map encoding of user interactions (based on DIOS [26]).

DeepIGeoS employs geodesic distance maps to propagate user input along intensity-coherent paths and concatenates these with both the image and base segmentation [19]. Its follow-up, DeepIGeoS-V2, opted for truncated Euclidean distances, reporting sensitivity of geodesic distances to low contrast in their data [21]. Other pipelines combine distance encodings with probabilistic priors such as Markov/Conditional Random Fields (MRF/CRF) [8, 9] or design alternative encodings (e.g., exponentialized geodesic maps in MIDeepSeg [23]).

Selecting an encoding is dataset-dependent: geodesic distances can respect topology and boundaries but may be unstable with weak gradients; Euclidean/truncated distances are robust and cheap but less boundary-aware. Hyperparameters (truncation radius, normalization,

separate FG/BG channels) materially affect behavior and require careful tuning.

2.3. User Interaction Simulation

Interactive refinement requires training triples (image, base segmentation, user input). Since collecting realistic user inputs at scale is prohibitively expensive, most systems rely on simulated annotators. Early frameworks such as DIOS [26] and DeepIGeoS/DeepIGeoS-V2 [19, 21] generated annotations by sampling *points* within misclassified regions (under- or over-segmentation) and converting them into distance maps (Fig. 2.3). This approach made large-scale training feasible but oversimplified the geometry of human interactions.

A common limitation across these studies is a *mismatch* between training and evaluation: models trained on simulated *points* are often tested with human *scribbles* (freehand strokes). This "annotation shift" alters the distribution of auxiliary guidance channels (e.g., distance transforms) presented at deployment, potentially degrading refinement quality. To our knowledge, no prior work has systematically simulated scribbles during training to close this gap. Our work addresses this omission by generating human-like scribbles that better approximate realistic annotations, and by investigating encoding strategies tailored to such inputs.

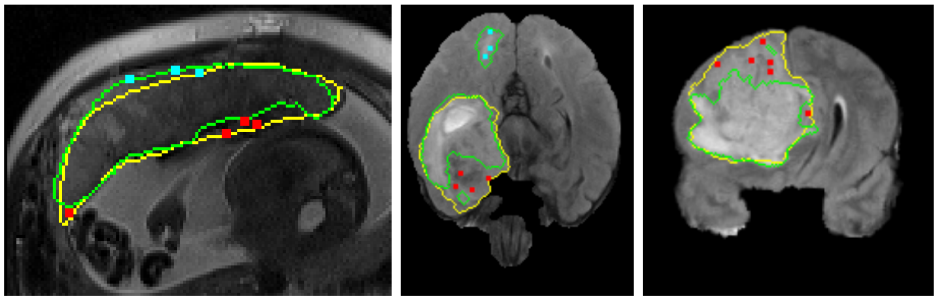


Figure 2.3: Simulated user clicks on under-/over-segmented regions (green: automatic, yellow: ground truth, red/cyan: clicks). Adapted from DeepIGeoS [19].

3

Methodology

Our methodology combines a multi-stage deep learning framework with a scribble simulator designed to mimic human corrections. A base segmentation model (BaseNet) provides coarse masks, while a refinement model (RefNet) updates these masks based on annotations. To close the annotation-shift gap, we simulate scribbles during training. Figure 3.1 shows the overall workflow.

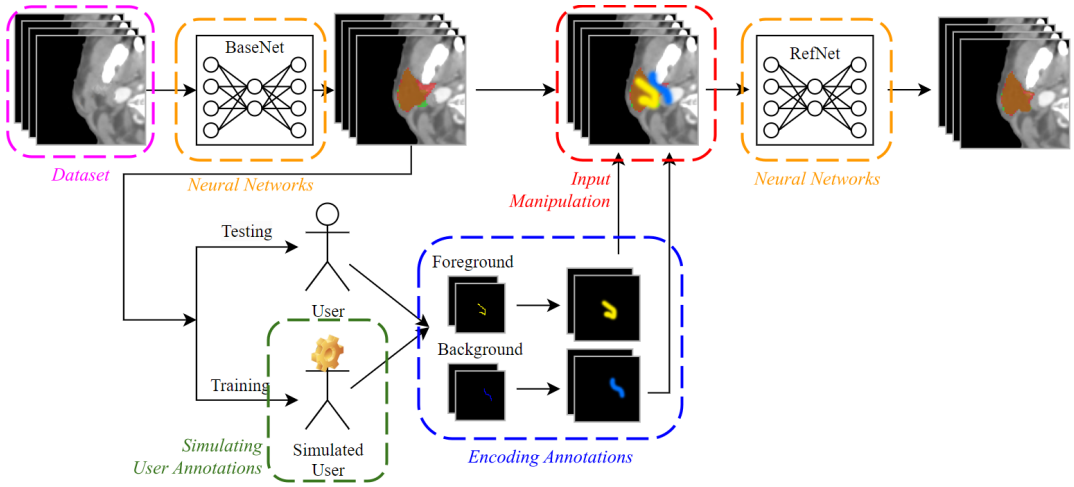


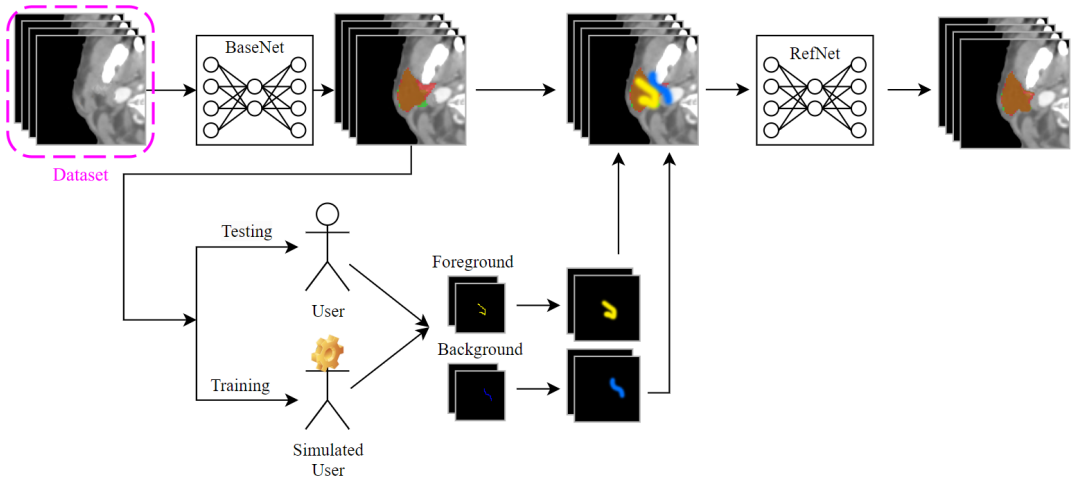
Figure 3.1: Schematic of the training/testing pipeline.

A two-model system was chosen because the generation of an initial segmentation and the refinement of that segmentation based on user input are fundamentally different tasks. BaseNet is designed to produce a coarse but reliable mask directly from the entire image, capturing the overall anatomical structure without requiring any interaction. RefNet, on the other hand, is specifically intended to improve this initial prediction by incorporating sparse user annotations. Separating these tasks allows each network to specialize in a clearly defined objective: BaseNet focuses on global segmentation, while RefNet focuses on localized correction of remaining errors. This design also reflects the intended interactive workflow more naturally, where an automatic prediction is first generated and subsequently refined through user feedback. In addition, the modular setup makes the framework more flexible, since both stages can be trained, evaluated, and improved independently and the BaseNet can be switched out for a state-of-the-art segmentation model.

We break up the methodology into six distinct sections in the order in which data is flowing through the pipeline.

1. ***Dataset:*** describes the dataset used in this work, together with the preprocessing and augmentation steps applied to standardize the scans and improve generalization during training.
2. ***Neural Networks (BaseNet):*** introduces the architecture and training procedure of BaseNet, which generates the initial coarse segmentation that serves as the starting point for refinement.
3. ***Simulating User Annotations:*** presents the proposed scribble-simulation strategy, including the selection of informative error regions and the generation of realistic foreground and background scribbles.
4. ***Encoding Annotations:*** explains how the generated scribbles are transformed into distance-map representations so that they can be incorporated as structured input to the refinement model.
5. ***Input Manipulation:*** describes the additional transformation applied to the base segmentation using the annotation information, ensuring that the correction signal is made explicit before refinement.
6. ***Neural Networks (RefNet):*** details the architecture, loss function, and training pipeline of RefineNet, which combines the image, base prediction, and annotation encoding to produce the final refined segmentation.

3.1. Dataset



This section describes the dataset used in this study and the preprocessing and augmentation steps applied to it. Because segmentation performance depends strongly on the consistency and quality of the input data, these steps are designed both to simplify the incorporation of annotations and to improve model generalization.

At a high level, the preparation pipeline consists of two stages:

- ***Before training (resampling and resizing):*** the scans are standardized by isolating the target organ, centering it within the volume, resampling to isotropic voxel spacing, cropping to a fixed size, and normalizing the intensity values.
- ***During training (data augmentation):*** controlled perturbations are applied to expose the model to realistic anatomical and acquisition-related variation without introducing implausible samples.

Dataset choice

The dataset used for training and evaluation is the MICCAI 2015 Head and Neck Segmentation Challenge dataset, introduced by Raudaschl et al.[1]. It contains 48 CT scans of the head and neck region. Each scan is accompanied by expert delineations of organs-at-risk (OARs), which serve as ground-truth segmentations.

In this work, we restrict the task to binary segmentation of the left and right parotid glands. These glands are known to be challenging to segment, with prior work reporting mean Dice scores of approximately 0.84, although smaller structures such as the optic chiasm and optic nerves are generally even more difficult.[41, 42]

Before training

Several preprocessing operations are applied before training to improve inter-sample consistency while preserving the anatomical characteristics relevant to segmentation.

Resampling The scans appear to originate from different scanners or acquisition protocols, as voxel spacing varies across the x, y, and z axes. This introduces heterogeneity and complicates spatially dependent operations.

First, the left and right parotid glands are isolated and centered within the volume, reducing positional variability across samples. Each volume is then *resampled* to a 1 mm isotropic voxel spacing. This simplifies the computation of distance maps and ensures that spatial relationships are represented consistently in all directions.

Cropping Each sample is then *cropped* to a fixed volume of $64 \times 64 \times 64$ voxels. This allows the full volume to be processed efficiently in GPU memory, avoiding patch-wise segmentation. A side length of 64 was chosen because it is a power of two, which is convenient for the repeated down-sampling and up-sampling operations in a U-Net.

Finally, the CT intensities are normalized to the range $[0, 1]$. Because Hounsfield units have a consistent physical meaning across CT scans, we avoid normalization methods that alter intensities on a per-scan basis. In particular, z-score normalization may reduce comparability between scans and is sensitive to outliers. Instead, we linearly rescale the intensity range $[0, 300]$ to $[0, 1]$. This interval covers the values most relevant to the present task, from water-equivalent tissue to denser soft tissue and lower-density bone.

During training

Data augmentation In addition to preprocessing, we apply online data augmentation during training to improve generalization. This is especially relevant in a relatively small dataset with substantial anatomical variability. The goal is to expose the network to realistic variation in appearance and positioning while preserving anatomical plausibility.

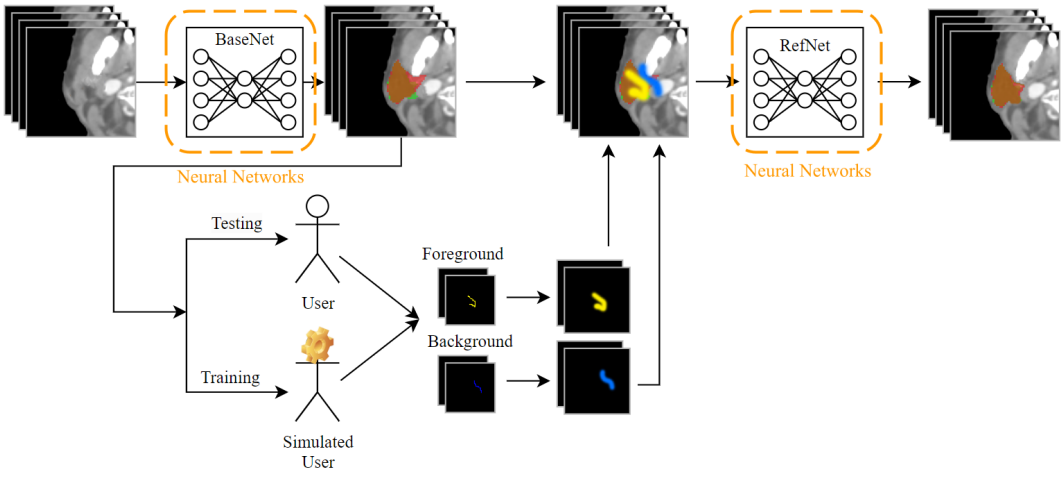
We apply small random spatial transformations to account for minor differences in patient positioning and scan acquisition. In head-and-neck CT, patients are typically positioned carefully, so large translations or rotations are uncommon. For that reason, the augmentation ranges are kept deliberately small.

The transformation ranges are:

- Rotation: $[-5, 5]^\circ$
- Shift: $[-5, 5]$ voxels
- Scale: $[0.90, 1.10]$

We also add Gaussian noise with mean 0 and standard deviation 0.1. This is motivated by the fact that CT images contain scanner-related noise: differences in hardware, acquisition settings, and reconstruction methods can introduce small intensity fluctuations. Adding a limited amount of noise during training improves robustness to such variation while preserving the relevant anatomical information. Although CT noise is not strictly Gaussian from a physical perspective, and is more accurately described by a Poisson–Gaussian model, Gaussian noise remains a common and practical approximation in medical image augmentation.[43, 44]

3.2. Neural Networks (BaseNet)



In contrast to previous approaches that utilized the MICCAI Head and Neck dataset for interactive refinement by categorizing all organs simultaneously, we opted for a binary classification method. This choice is more suitable and practical in clinical settings, where annotators typically annotate one organ at a time before proceeding to the next. [45]

Similar to other interactive refinement approaches, we employ two deep learning models: a base model (BaseNet) and a refinement model (RefNet). The base model generates an initial coarse segmentation of the image, which the refinement model then enhances using user annotations. In this section we will go over BaseNet.

Architecture

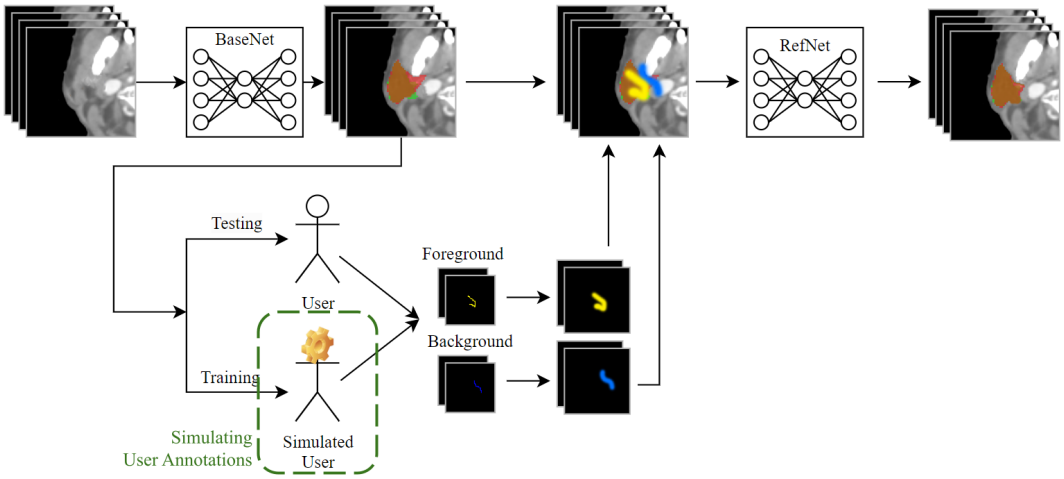
For the BaseNet we aim to look more into accuracy rather than speed, as that model is only used once per CT scan. Most of the previous work used networks similar to or inspired by the U-Net architecture, which is a fully convolutional network that uses skip connections to preserve spatial information. U-Net was originally designed to work on 2D image data, so for CT data, the similarly structured 3D U-Net architecture is more commonly used. We decided to use a model inspired by FocusNet, which is a variation of 3D U-Net, which provides better accuracy on the Head and Neck (HaN) dataset due to solving the problem of imbalanced small and large organs.

Training

Training the base network is a simple binary segmentation process. Since we use a FocusNet inspired architecture, we use a similar training strategy. We use a binary cross-entropy loss function as shown in equation3.1, which is the standard loss function for binary segmentation where y is the ground truth and $\hat{y}$ is the prediction of the BaseNet. We use the Adam optimizer with a learning rate of 0.001 and a batch size of 5. We then train the model for 300 epochs.

$$\mathrm{BCE}(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i) \tag{3.1}$$

3.3. Scribble Simulation



During training, the model requires user-provided annotations as an additional input signal. In prior work, such interactions are often simulated by selecting random points within the current error regions during training, while evaluation is performed using real human scribbles. This introduces a mismatch between training and testing conditions, which we previously called the annotation-shift. The central motivation of this work is therefore to investigate whether training with simulated scribbles that more closely reflect human annotation behavior leads to improved final segmentation accuracy.

An important constraint of the proposed implementation is that annotations must be provided as a pair of complementary scribbles. More specifically, the method requires a *foreground* scribble, indicating voxels that belong to the target organ, and a *background* scribble, indicating voxels that belong to the surrounding background. This requirement is rooted in the way the scribble information is incorporated into the neural network. The model expects a fully defined input representation for each annotation channel. Consequently, it is not sufficient to provide supervision on only one side of the boundary while leaving the complementary side unspecified, since neural network inputs cannot meaningfully encode an 'undefined' state in one of their layers without introducing ambiguity into the representation. This will be explained more thoroughly in the Refinement Network section.

Anatomy of a scribble

To simulate scribbles in a meaningful way, it is first necessary to define which characteristics such annotations should possess. Although human scribbles are inherently subjective and may vary between annotators, we identify several properties that are important for constructing a realistic approximation. These properties are chosen to reflect both the practical way in which users interact with segmentation systems and the type of corrective information that such interactions are intended to provide.

The simulated scribbles should satisfy the following properties:

- **They should focus on large erroneous regions.** In an interactive setting, a user is more likely to correct prominent mistakes first, since these regions contribute most strongly to the perceived segmentation error. [46]
- **They should resemble freeform lines.** Human annotations are typically drawn as continuous hand motions rather than as isolated points or highly regular geometric shapes.

Representing scribbles as freeform lines therefore makes the simulated interactions more consistent with the physical act of manual annotation.

- **Include variation in shape, placement, and extent.** Real human scribbles are not perfectly deterministic: their exact placement, curvature, and extent vary between annotators and even between repeated annotations by the same person. Incorporating randomness prevents the simulation from becoming overly rigid and reduces the risk that the model overfits to an artificial annotation pattern.
- **They should operate primarily on individual slices rather than the full volume.** In clinical practice, annotations are commonly made in a slice-wise manner, as users typically inspect and correct volumetric scans through two-dimensional views. Restricting scribbles to slices therefore better matches the way interactive corrections are performed in real annotation workflows.

Taken together, these design choices aim to produce simulated scribbles that approximate the behavior of a human annotator more closely than simpler point-based interaction models. In the following sections, we propose a scribble generation method that satisfies each of these properties.

Scribble Pipeline

The proposed scribble-generation procedure consists of several stages, each designed to identify informative annotation locations from the disagreement between a base segmentation and the corresponding ground truth. The objective of this pipeline is to place user annotations in the form of scribbles in regions where the current segmentation deviates most strongly from the reference annotation, thereby yielding sparse but meaningful supervision.

Given a base segmentation and its associated ground-truth mask, the pipeline proceeds as follows:

1. selection of a compatible slice;
2. detection and ranking of erroneous regions within that slice;
3. generation of scribbles for the selected erroneous region;
4. rendering of the generated scribbles onto the slice.

3.3.1. Compatible Slice Selection

The first stage consists of selecting a slice on which scribble generation is meaningful. For this purpose, a slice is considered *compatible* if it contains pixels that are in the base segmentation and not in the ground-truth and vice versa. From these slices a candidate slice is randomly selected.

3.3.2. Error-Region Detection

After selecting a slice, we need to find the most erroneous regions between the segmentations in the slice, which involves calculating a score for each erroneous region in the slice and selecting the one with the largest score.

Ranking Erroneous Regions

Erroneous regions are identified by extracting the contours of both the base segmentation and the ground-truth annotation in the selected slice. In general, these contours do not coincide exactly, and the discrepancies between them correspond to oversegmentation and undersegmentation errors.

Pictured in 3.2a is an example of a missegmentation. The ground-truth is represented

by the green contour, and the base segmentation is represented by the red contour. A human can spot the region with the largest error (as defined in the next section) intuitively. To be able to select an appropriate error region we first need to isolate each region. We know that an erroneous region is enclosed by intersections between the contours, as pictured in 3.2b.
By taking the ground-truth and base contours between each crossing we can isolate all error regions as pictured in 3.2c. This contour-based decomposition was chosen because it transforms a potentially complex global mismatch into a set of localized correction candidates, each of which can be evaluated independently.
In the special case where there is no crossing, then we know that the entire segmentation is wrong and the segmentation as a whole will be selected as an erroneous region.

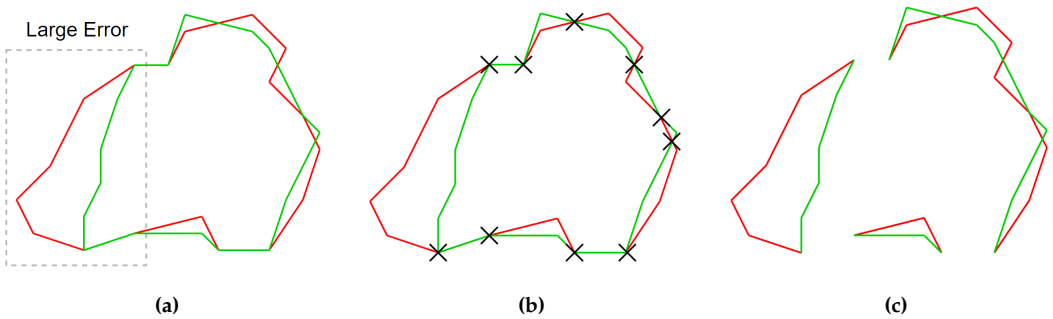


Figure 3.2: Isolating erroneous regions.

Calculating Error

In order to rank erroneous regions we need a way to quantify the error for any region. For this purpose, the Hausdorff distance is used to measure the discrepancy between the contour segment from the base segmentation and the corresponding contour segment from the ground truth. This metric was selected because the aim of the ranking stage is not merely to measure average disagreement, but to identify the region with the most pronounced local error. The Hausdorff distance is well suited to this purpose, since it is sensitive to the largest deviation between two point sets. The Hausdorff distance between two contours is defined as the maximum distance from a point on one contour to its closest point on the other contour. The formula for the Hausdorff distance is shown in 3.2.

$$HD(A,B) = \max\left(\max_{x\in A}\min_{y\in B}\|x-y\|_2,\ \max_{y\in B}\min_{x\in A}\|x-y\|_2\right), \tag{3.2}$$

where $\|x-y\|_2$ denotes the Euclidean distance between contour points x and y.

We calculate the Hausdorff distance for each region and then calculate the one with the largest error. An example of this applied to the slice from the previous section is shown in figure 3.3.

3.3.3. Scribble Generation

The next step in our scribble generation method is to generate scribbles for the selected erroneous region.

Generating a Scribble

Once an erroneous region has been selected, the next step is to generate the corresponding *foreground* and *background* scribbles. Since the selected region is bounded by the ground-truth

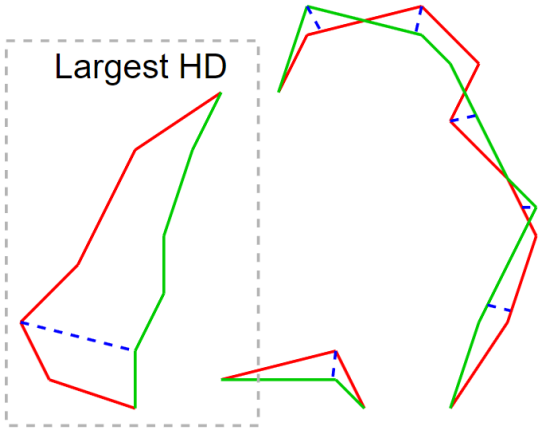


Figure 3.3: Finding the largest error region by Hausdorff distance

and base contours, the ground-truth contour provides a natural geometric reference for this construction. More specifically, because the ground-truth contour represents the boundary of the target organ, two contours drawn approximately parallel to it will, by construction, lie on opposite sides of that boundary: one inside the organ and one in the surrounding background. This makes the ground-truth contour a suitable basis for generating paired corrective annotations. An example is shown in Figure 3.4, where the ground-truth contour is depicted in orange, and the inner and outer annotations are shown in yellow and blue, respectively.

When generating these annotations, the offset from the ground-truth contour is chosen randomly between 1 and 5 units. The inner and outer contours are offset independently, such that the midpoint between them does not trivially coincide with the ground-truth contour. This design introduces asymmetry between the foreground and background scribbles and avoids encoding the target boundary in an overly explicit manner.

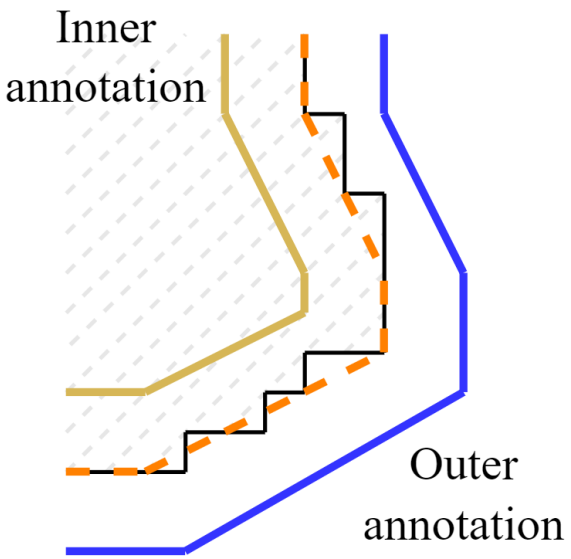


Figure 3.4: Offsetting a scribble from the ground truth

Perturbing the Scribble

However, offsetting and smoothing alone are insufficient to produce realistic scribbles. If the generated contours were obtained only by applying a uniform offset to the ground-truth boundary, the distance from each point on the inner and outer contours to the reference contour would remain approximately constant. As a result, the generated annotations would appear overly regular and would fail to reflect the local shape variations that naturally arise in human freehand input. To better approximate real user interactions, additional variation must therefore be introduced beyond simple geometric offsetting. To illustrate our approach to introduce more realistic perturbations we will apply our technique to the scribble shown in Figure 3.5.

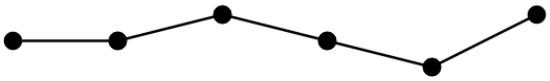


Figure 3.5: Initial scribble

Random Translation Perturbation To make the generated scribbles appear more realistic, we considered perturbing the points that define the curve. A simple approach is to apply an independent random translation to each point, with the displacement in both the x- and y-directions sampled from a uniform distribution, as defined in Equation 3.3, where s represents the maximum translation magnitude.

$$\begin{aligned} x &= x + U(-s, s) \\ y &= y + U(-s, s) \end{aligned} \tag{3.3}$$

This produces a scribble of the form shown in Figure 3.6. However, the method has an important limitation: it ignores the ordering of the points along the curve. Since a scribble is fundamentally an ordered freeform line, independently perturbing its points can disrupt the local continuity of the shape and lead to unrealistic behavior, such as abrupt distortions or self-overlap, as shown in Figure 3.7. In principle, this issue can be mitigated by constraining s such that the displacement remains small relative to the spacing between neighboring points. However, this also limits the amount of variation that can be introduced. For this reason, we did not adopt this strategy and instead sought an alternative method that offers greater control over the perturbation of the generated scribbles.

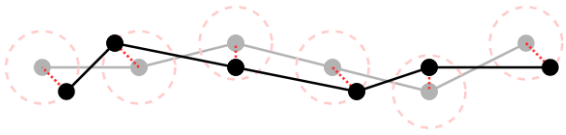


Figure 3.6: Using random offsets with s less than half the distance between points

Translating along the normal In the previous perturbation strategy, each point of the scribble was translated independently in a random direction. As discussed above, this approach treats the points as isolated samples rather than as elements of an ordered curve. Since a scribble is fundamentally a line-like structure, it is more appropriate to base the perturbation on geometric properties of the line itself. A natural choice is the *normal vector*, i.e., a vector perpendicular

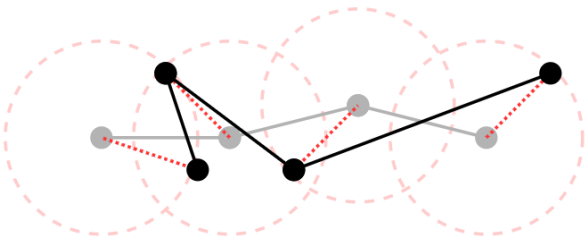


Figure 3.7: Problems when s is greater than half the distance between points

to the local tangent of the curve. For a line segment defined by two points p_1^{xy} and p_2^{xy}, the corresponding unit normal can be computed using Equation 3.4.

$$\begin{aligned} n^x &= p_2^y - p_1^y, \\ n^y &= p_1^x - p_2^x, \\ m &= \sqrt{(n^x)^2 + (n^y)^2}, \\ \hat{n} &= \frac{1}{m} \begin{bmatrix} n^x \\ n^y \end{bmatrix}. \end{aligned} \tag{3.4}$$

Using this definition, a normal can be assigned to every point on the scribble. For the first point, the normal is taken from the segment connecting the first and second points; for the last point, from the segment connecting the final two points. For all intermediate points, the normal is defined as the average of the normals of the two adjacent line segments. This yields a locally consistent perpendicular direction for each point along the curve. An illustration of this construction is shown in Figure 3.8.

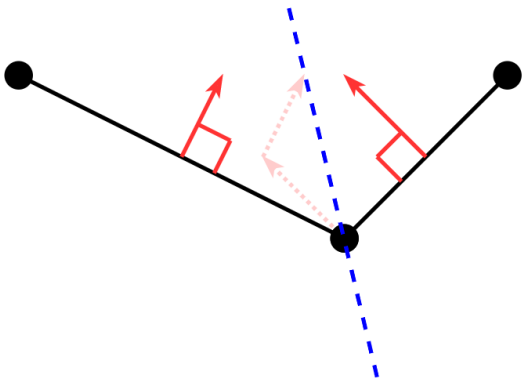


Figure 3.8: Calculating the perpendicular line for a point using the normals

After gathering these perpendicular lines, we can attempt the same perturbation method as before, but now using the normals as a guide instead of a random direction. For the formula we use a slightly adjusted formula to only move along the normals.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + U(-s, s) \begin{bmatrix} n^x \\ n^y \end{bmatrix}. \tag{3.5}$$

An example of the resulting scribble is shown in Figure 3.9. Compared with fully random translation, this approach yields perturbations that better respect the line-like structure of the

scribble, since each point is displaced in a direction that is locally consistent with the curve geometry. This makes the method a better choice than isotropic random offsets when the goal is to preserve the continuity of the annotation.

Nevertheless, this strategy was ultimately not adopted as the final solution. The main limitation is that the perturbation is still applied independently at each point, and therefore remains fundamentally local. Although the direction of motion is now geometrically meaningful, the magnitude is still sampled point-wise, so the method primarily introduces small-scale irregularities rather than coherent large-scale deformations. In practice, this makes it well suited to modeling effects such as local jitter or a shaky hand, but much less effective at representing structured deviations that are common in human annotations, such as a scribble gradually drifting to one side or developing a broader bend over part of its trajectory.

For these reasons, perturbation along the normal was retained as an informative intermediate experiment, but a different method was sought that would provide more control over coherent, larger-scale variation in scribble shape.

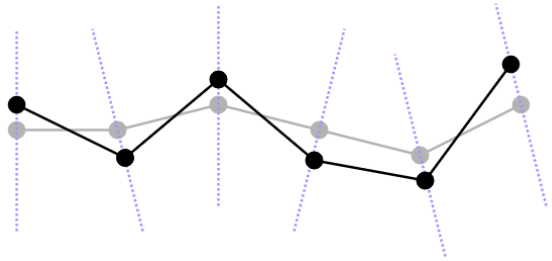


Figure 3.9: Using random offsets along the normal

Scribble Perturbations In the previous methods we looked at each point individually to generate perturbations, and explained why those traits are not necessarily desirable for us. In order to create perturbations that are applied along the entire length of a scribble we decided to use a sine function as a parameter for the perturbation, which allows us to create smooth and continuous perturbations that follow the shape of the scribble.

A further advantage of this formulation is that the frequency, phase, and amplitude can each be controlled independently. The amplitude determines the magnitude of the displacement, the frequency controls how rapidly the perturbation varies along the scribble, and the phase determines where the deformation pattern begins. A schematic illustration of these parameters is shown in Figure 3.10. To generate random perturbations, the sine function is defined as

$$F(i) = a \sin(2\pi f i + p), \tag{3.6}$$

where f denotes the frequency, p the phase, and a the amplitude. The parameter ranges for the corresponding uniform distributions were selected empirically through iterative visual inspection of the resulting scribbles.

After generating the sine function, we can use it to perturb the scribble by using the formula 3.7 where x and y are the coordinates of the point, n^x and n^y are the normal of the point and F is the sine function.

$$\begin{bmatrix} x'_i \\ y'_i \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} + a \sin(2\pi f i + p) \begin{bmatrix} n^x_i \\ n^y_i \end{bmatrix}. \tag{3.7}$$

This will create a perturbation that is applied along the entire length of the scribble, and will be more consistent with the shape of the scribble than the previous methods. An example of this is shown in figure 3.11.

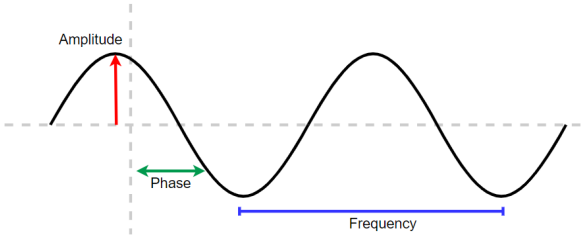


Figure 3.10: Sine function and controllable parameters

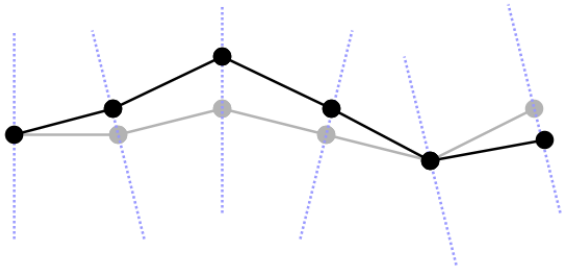


Figure 3.11: Using a sine function to perturb the scribble

3.3.4. Drawing the Scribble

The final step of the proposed procedure is to render the generated scribbles onto the image. At this stage, each scribble is defined by a sequence of points describing its trajectory. In order to obtain a discrete annotation mask, line segments are drawn between consecutive points using the Bresenham algorithm [47]. This choice is motivated by the fact that the algorithm is computationally efficient, operates directly on the discrete image grid, and produces consistent pixel-level line renderings. The resulting rasterized scribbles can then be used in the subsequent computation of the distance maps.

3.3.5. Evaluation of the scribble generation strategy

After evaluating several alternative strategies for synthetic scribble generation, the final method adopted in this work combines three key components: region ranking, contour simplification, and sinusoidal perturbation. This combination was selected because it satisfies the design requirements formulated for the scribbles while remaining transparent, controllable, and straightforward to implement. To recap, the requirements we had for the scribbles were as follows:

- **The scribbles should focus on the largest erroneous regions.**
  This is achieved through the ranking of erroneous regions, which prioritizes locations with the greatest local disagreement between the base segmentation and the ground truth.
- **The scribbles should resemble freeform annotation lines.**
  This is achieved by simplifying and smoothing the contour before scribble construction, and by applying smooth sinusoidal perturbations instead of sharp or discontinuous deformations.
- **The scribbles should variation in shape, placement, and extent.**
  This is achieved through the perturbation stage and the randomized offsets used during scribble construction, which introduce variation while keeping the annotation anatomically meaningful.

– **The scribbles should operate primarily on individual slices rather than the full volume.** This requirement is satisfied by design, since the proposed pipeline generates and applies scribbles on individual two-dimensional slices rather than on the three-dimensional volume as a whole.

For these reasons, the final method represents a practical compromise between realism, controllability, and methodological consistency. Examples of scribbles generated with this approach are shown in Figure 3.12.

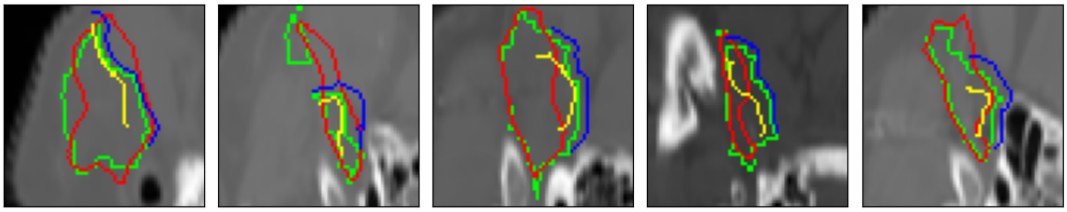
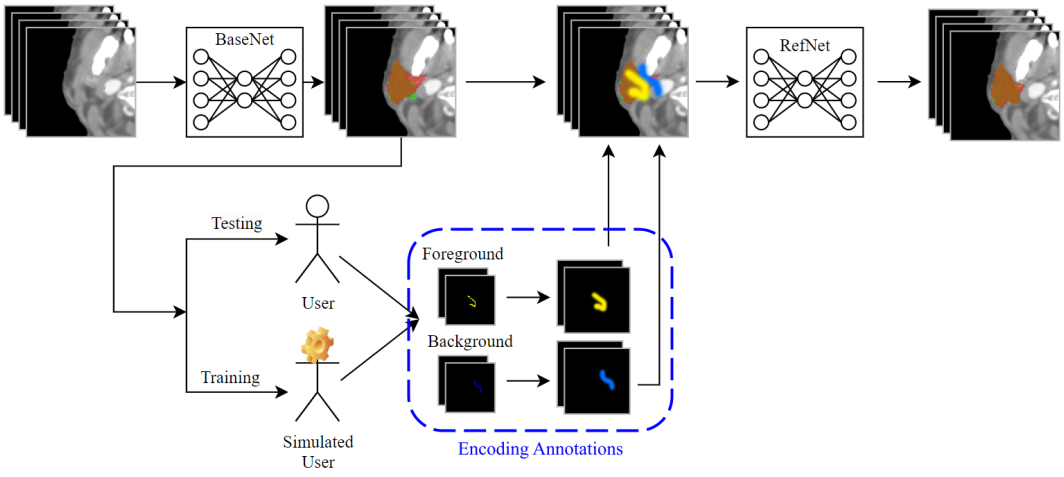


Figure 3.12: Examples of scribbles generated using the final method.

3.4. Encoding User Annotations



In order to incorporate the knowledge of the user into the refinement step we need to convert the user annotations into a format that the refinement model can use. Similar to previous works we will encode the user annotations as an additional channel to be appended to the input to the neural network. In our case this can be encoded in the form of a pair of distance maps, each corresponding to a different class for the binary classification task, i.e. a distance map for the foreground class and a similar map for the background class. This notion of a distance map is inspired by the work of [**DeepIGeos**], which uses a distance map to encode the user annotations. The intuitive idea behind a distance map is that we calculate the distance of each voxel in the image to the closest user annotation. This distance is then encoded as a value in the distance map.

Distance Maps To calculate a distance map M, we can create a simple equation that calculates the distance of each voxel in X to the closest user annotation in A, given a distance function $D(a, b)$, which returns the distance from voxel a to b. In the 3D medical imaging case, this will result in a 3-dimensional matrix of the same size as the input volume. The distance map can be constructed by applying this formula to each voxel in the image:

$$M_i = 1 - (\min_{a \in A} D(X_i, a)) / \max_{a \in A, x \in X} D(x, a) \tag{3.8}$$

This results in an inverse normalized distance value per voxel, which returns a value of 1 for the voxels closest to the user annotation and 0 for the voxels furthest away from the user annotation. The basic idea of the distance map is quite straightforward, but careful consideration should be taken when deciding on which distance function to use.

Context-agnostic Distance Functions All distance functions are quite similar in concept, however the methods they use can be classified in two main categories. Context aware and context agnostic. Context aware methods take into account the content of the image, leveraging important context cues like borders between tissues and voxel values. Context agnostic methods only use the spatial information of the voxels to calculate a distance. An example of this can be seen with the truncated euclidean distance function, which computes the euclidean distance from every voxel to the nearest annotation but truncates the result by some threshold k. This creates a distance that is context agnostic, as it only considers the positions of the pixels, not at the values of the voxels. In this example a and b are the positions of the pixels, while k is the

threshold.

$$D^k_{EUC}(a,b) = \min(\sqrt{(a-b)^2}, k) \tag{3.9}$$

Context-aware Distance Functions In contrast to the context agnostic distance functions, context aware distance functions take into account the context of the image. An example of this is the geodesic distance function shown in the equation below. This function is context aware, as it takes into account the values of the voxels (taken from the CT volume X) and not just their positions. A more in depth explanation of this function is discussed later.

$$\begin{aligned} P_{a,b} &= \text{All possible paths between } a \text{ and } b \\ D_{GEO}(a,b) &= \min_{p \in P_{a,b}} \sum_{a',b' \in p} abs(X_{a'} - X_{b'}) + \epsilon \end{aligned} \tag{3.10}$$

Whether a context-aware or context-agnostic distance function is "better" is a matter of debate. In the case of our dataset we empirically evaluated several distance functions in which context-agnostic functions seemed to suit it better. We hypothesize that this might be related to the characteristics of our dataset work i favor of the context-aware distance maps, which we will expand upon later.

3.5. Choosing a Distance Function

Based on our empirical evaluation, we ultimately chose the **Gaussian distance** as the distance function for the refinement model. We selected this option because it provided the best balance between propagating the user annotation to nearby relevant voxels and limiting undesired information spill into surrounding regions. In particular, compared to the other candidate functions, it offered a smooth and spatially localized encoding without the computational overhead and parameter sensitivity of context-aware alternatives such as the geodesic distance.

In the remainder of this section, we discuss the other distance functions that were considered, together with their advantages and disadvantages, and explain why they were not selected as the final encoding strategy. We investigated the following distance functions:

- Truncated Euclidean Distance
- Geodesic Distance
- Intensity Distance
- Gaussian Distance

Truncated Euclidean Distance

The truncated euclidean distance is a simple distance function that is easy to implement. This is the distance function used by [21] and is the distance map we started out using in our first exploratory experiments. The main advantage of this formulation is its simplicity: it is computationally cheap, easy to implement, and has been shown to work well in prior interactive segmentation methods. For this reason, it served as a strong baseline in our experiments. However, we did not choose it as the final distance function. While it provides a reasonable spatial prior, its linear decay and hard truncation at distance k make the propagated signal relatively abrupt. In practice, we found that this gave a less natural transition of annotation influence than the Gaussian alternative, which provided a smoother decay and better control over local information propagation.

$$D^k_{EUC}(a,b) = \min(\sqrt{(a-b)^2}, k) \tag{3.11}$$

Geodesic Distance

This is a distance function that, as the name suggests, treats the problem geometrically. Instead of computing the straight-line distance between two points, it represents the data as a space or surface and determines the shortest path between the points while remaining on the data manifold. To provide an easier understanding we will provide an explanation of this distance function in 2 dimensions, but it can be expanded to any n-dimensions. We can imagine how this function works by representing a 2D slice of a CT volume as part of a world map, where the intensities of the pixels are the height of the terrain. Then we can calculate the shortest path between two points on this map, where going up and down hills takes more time than finding a straight path. The distance between two points on this map in the x and y directions will be of the same unit, however calculating the height of the points requires careful tuning using a weight factor α. Setting this setting too low will result in slight bumps which will not influence the distance in any meaningful way, but setting it too high would cause a slightly textured surface to not get adequately filled in.

Calculating the geodesic distance can be done by creating a graph that represents the image, where each pixel is a node and the edges between the nodes are weighted by differences between the values of the pixels and some value ϵ. The geodesic distance is then calculated by using the Dijkstra shortest path algorithm on this graph and computing the distance on this path. The epsilon value is used, because moving of two pixels with the same value should still have a cost, as it is still a movement. This does require some fine tuning, as the value of ϵ and α can have a large impact on the distance map. In this example, a and b are the positions of the voxels, while X is set of voxel values.

$$\begin{aligned} P_{a,b} &= \text{All possible paths between } a \text{ and } b \\ D^{\epsilon,\alpha}_{GEO}(a,b) &= \min_{p \in P_{a,b}} \sum_{a',b' \in p} \alpha * abs(X_{a'} - X_{b'}) + \epsilon \end{aligned} \tag{3.12}$$

This is the distance map used by [19], which they chose not to use in the next iteration of their paper [21] in favor of the truncated euclidean distance. This is because they found that the geodesic distance map was too slow and they deemed the results of the truncated euclidean distance map to be better suited for this problem.

Challenges with the Geodesic Distance Map The geodesic distance function aims to address a fundamental issue with using distance maps for propagating information, such as user annotations, to nearby voxels. The primary goal of distance maps is to extend this information effectively. However, without proper context, information indicating that a voxel belongs to the foreground might inadvertently spread to neighboring voxels that shouldn't receive it. An illustrative example is shown in Figure 3.13, where the information spills over to all adjacent voxels, including those that do not benefit or even receive incorrect data. This can lead to erroneous inclusions of voxels across the segmentation boundary during the refinement process. One might attempt to mitigate this issue by reducing the information spread, but this also limits the dissemination of valuable information to potentially beneficial voxels. The geodesic distance function tries to tackle this by penalizing the crossing of intensity gradients, aiming to confine information flow within the segmentation while minimizing annotation spillover.

However, geodesic distance maps have their own drawbacks, particularly when dealing with noisy input data. Analogous to geographical maps where substantial differences in voxel values, akin to mountain ranges, effectively reduce annotation spill, noise can disrupt these barriers. Noise introduces variations that create "tunnels," thereby allowing the distance map to extend further, making more of the image accessible. This effect is demonstrated in the

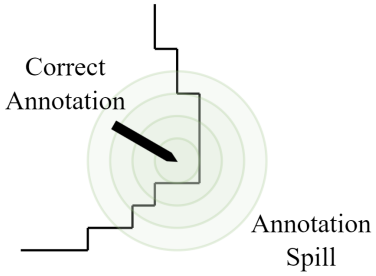


Figure 3.13: Information spilling towards neighbouring pixels

following image, where added noise causes the distance map to spread. Some efforts to mitigate this, such as image blurring, have been explored, but they provided no significant advantage for our dataset. Given the prevalent noise in CT scans, we concluded that the geodesic distance map may not be particularly effective for our data.

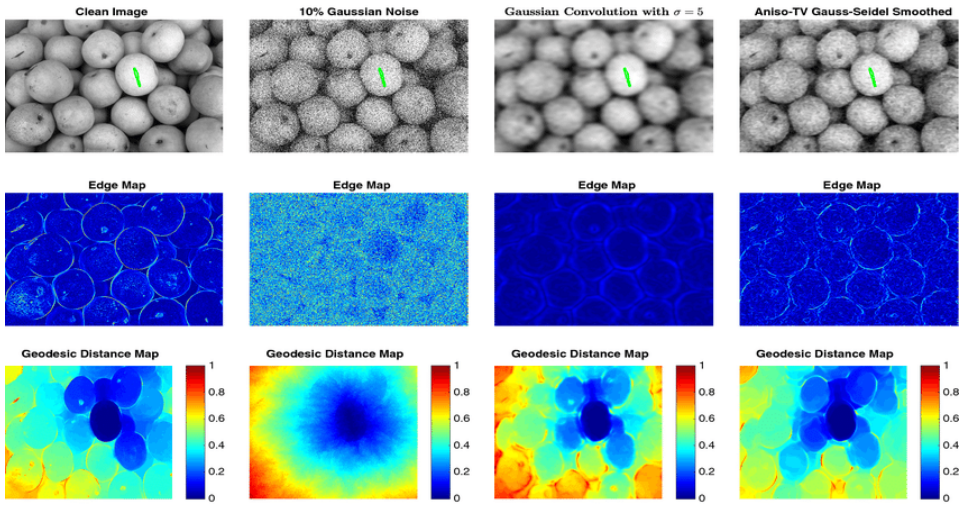


Figure 3.14: Geodesic distance map with noise. From Roberts et al. [30]

Intensity Distance

Another distance function that is context-aware is the intensity distance. This distance function simply looks at the differences between voxel values without taking into account the position of the voxels. This approach is appealing because it is simple, context-aware, and does not require the tuning of an explicit spatial scale parameter. Nevertheless, we did not choose it as the final encoding. Its main limitation is that it ignores spatial proximity, which is crucial in interactive refinement. As a result, voxels that are anatomically far apart but have similar intensities may be treated as equally relevant to the annotation. An example within our domain would be the left and right parotid glands. Using this distance function and a randomly sampled point in the left parotid gland, we see that the distances in the left and right parotid glands to this point are similar, even though they are on opposite sides of the CT scan.

$$D_{INT}(a,b) = abs(X_a - X_b) \tag{3.13}$$

Gaussian Distance

This is the final context-agnostic distance function we explored. This distance function is similar to the truncated Euclidean distance map, but instead of using a linear decay, it uses a

Gaussian decay. This method still requires a parameter σ to be set, which we empirically set to 2 after trial and error. This is because of the information spill principle mentioned previously. A larger σ will cause the distance map to spread out more, which is not desirable. A smaller σ will cause the distance map to be more constrained but may not be able to propagate information to far away voxels. We found that a σ of 2 was a good balance between these two.

$$D^{\sigma}_{GAUSS}(a,b) = exp(-\frac{(a-b)^2}{2\sigma^2}) \tag{3.14}$$

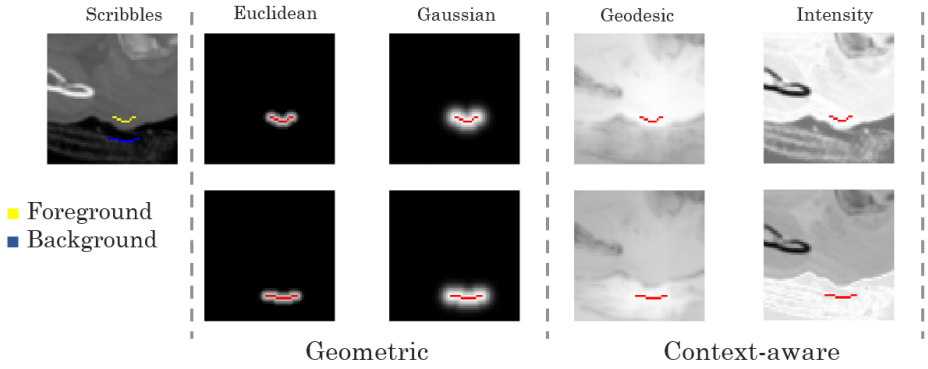
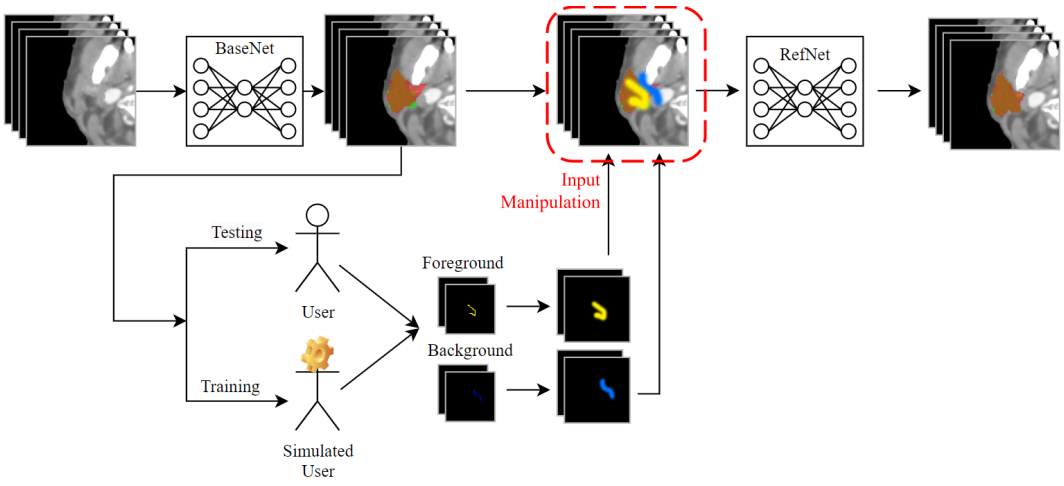


Figure 3.15: Comparison of the different distance functions creating different distance maps.

Overall, our comparison suggests that, for this dataset, a simple context-agnostic distance function is preferable to more elaborate context-aware alternatives. Although context-aware methods are attractive in theory because they can incorporate image structure, in our experiments their benefits were outweighed by their sensitivity to noise and parameter settings. The Gaussian distance offered the best trade-off between simplicity, robustness, and effective propagation of the user annotations, and was therefore used in the final refinement pipeline.

3.6. Input manipulation



In addition to encoding the user annotations as distance maps, we apply a direct manipulation to the base segmentation before passing it to the refinement model. We chose to include this step because the user annotations are assumed to be correct during both training and evaluation: every voxel marked by a foreground scribble is treated as belonging to the foreground class, and every voxel marked by a background scribble is treated as belonging to the background class. Under this assumption, it is undesirable for the refinement network to receive an input segmentation that already contradicts information explicitly provided by the user. Therefore, we modify the base segmentation so that it is consistent with the annotations from the user.

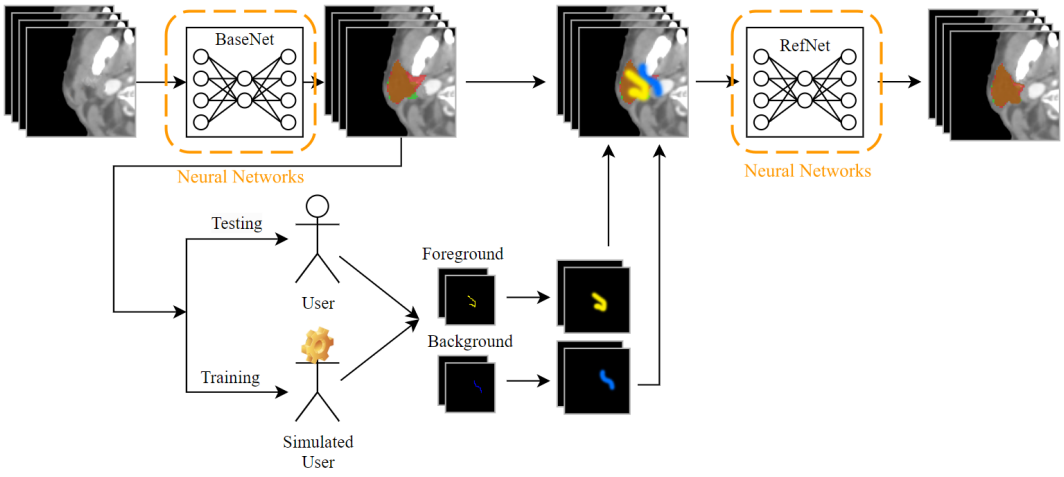
The transformation is intentionally simple and is defined in Equation 3.15. Foreground scribbles are enforced by adding them to the predicted foreground region, while background scribbles are enforced by removing them from it. This produces a manipulated segmentation that satisfies the hard annotation constraints before refinement begins.

$$\text{seg}_{base} = \quad \text{scribble}_{fg} = \quad \text{scribble}_{bg} = \tag{3.15}$$

$$\text{seg}_{manip} = (\text{seg}_{base} \vee \text{scribble}_{fg}) \wedge \neg\text{scribble}_{bg}$$

We chose this formulation for several reasons. First, it directly encodes the strongest supervision signal available in the interactive setting: if the user explicitly marks a voxel as foreground or background, the system should respect that decision. Second, the operation is deterministic and transparent, making its behavior easy to interpret and verify. Third, it introduces no additional trainable parameters and almost no computational overhead, yet it provides a meaningful improvement in refinement performance. In that sense, this manipulation is a pragmatic design choice: rather than asking the network to infer an obvious correction rule indirectly from the data, we explicitly enforce the rule in the input.

3.7. Neural Networks (RefNet)



The refinement network has a fundamentally different role from the base segmentation network. Whereas the BaseNet is responsible for producing a strong initial segmentation from the CT volume alone, the RefNet is intended to make targeted corrections based on user feedback. Because of this difference in purpose, we do not simply reuse the same architecture and training strategy as the BaseNet. Instead, we make several design choices that prioritize fast inference, responsiveness to user annotations, and efficient local correction of segmentation errors.

Architecture

For the refinement model in our proposed method, we chose to use a shallow 3D U-Net rather than the larger and more accurate FocusNet used in the BaseNet. This choice was made primarily to prioritize speed. In an interactive refinement setting, rapid feedback is essential: after placing annotations, the user should be able to see the effect of their interaction quickly. A slower model would negatively affect the usability of the system, even if it were to provide slightly better segmentation quality. Therefore, we deliberately favored a lightweight architecture that can produce fast updates during the refinement process.

A second reason for this choice is that the refinement task is inherently more local than the base segmentation task. The BaseNet must infer the full structure of the target anatomy directly from the image, which requires a large receptive field and a more expressive model. In contrast, the RefNet starts from an existing segmentation and only needs to correct regions indicated by the user annotations. Since the user scribbles already localize the regions of interest, the model does not need to rediscover the full global anatomy from scratch. As a result, a smaller receptive field is acceptable for the refinement stage, and the reduced depth of the 3D U-Net becomes a reasonable trade-off between speed and representational capacity.

We therefore chose a shallow 3D U-Net because it matches the role of the refinement stage well: it is computationally efficient, sufficiently expressive for local corrections, and better suited to interactive use than a heavier architecture such as FocusNet. In other words, this design reflects the principle that the refinement model should complement the BaseNet rather than duplicate it.

Inference Contrary to the base model, the refinement model takes multiple inputs: the 3D CT volume, the 3D segmentation predicted by the BaseNet, and the user annotations encoded as distance maps. The model then outputs a refined segmentation of the image.

We chose these inputs because each provides a different type of information that is necessary for the refinement task. The CT volume provides the underlying anatomical and intensity context, which allows the model to interpret tissue boundaries and local image structure. The base segmentation provides the current estimate of the target structure and serves as the starting point for refinement. Finally, the distance maps communicate the location and intent of the user annotations, indicating where corrections are needed and whether those corrections should favor foreground or background. By combining these inputs, the model can refine the existing segmentation in a targeted manner rather than recomputing the segmentation from scratch.

Loss Function

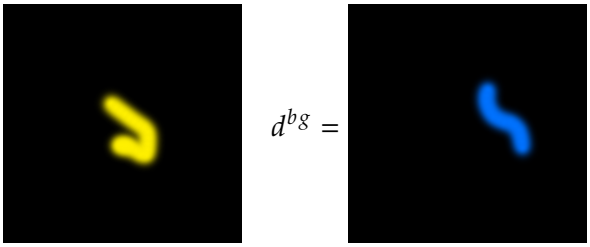
Rather than using the same training strategy as the BaseNet, we use a loss function that focuses more strongly on regions near the user annotations. We made this choice because the goal of the RefNet is not to relearn the entire segmentation problem, but to correct mistakes in response to user feedback. If we were to use the same loss function as the BaseNet with uniform emphasis over the whole volume, the optimization objective would encourage the network to improve overall segmentation quality in a broad sense, instead of specifically learning how to focus mainly on the annotated errors which is the intended purpose of the refinement stage.

To address this, we use a *weighted* binary cross-entropy loss in which voxels close to the user annotations receive a higher weight. This encourages the model to focus its learning on the regions where the user has explicitly indicated that refinement is needed. Since the distance maps already encode how close each voxel is to the foreground and background scribbles, they provide a natural and computationally convenient way to construct such a weight map. We therefore reuse this information directly rather than introducing an additional handcrafted weighting mechanism.

As a regularization term, we assign a small weight of 0.1 to the rest of the image so that the network does not completely ignore non-annotated regions. This is an important choice: while the primary objective is to improve the annotated area, the refinement should still remain globally consistent with the anatomy and should not introduce new errors elsewhere in the volume. The small baseline weight ensures that the model retains some awareness of the full image, while still concentrating most of its learning capacity on the relevant local corrections. The specific value of 0.1 was empirically chosen.

This loss is called the weighted binary cross-entropy loss and is shown in Equation 3.16, where y is the ground truth and $\hat{y}$ is the prediction of the RefNet. We also include the distance maps as d^{fg} for the foreground distance map and d^{bg} for the background distance map.

$$d^{fg} = \quad d^{bg} = \tag{3.16}$$



$$w_i = d_i^{fg} + d_i^{bg} + 0.1$$

$$\text{WBCE}(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} w_i (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i))$$

4

Evaluation

4.1. BaseLine

As a baseline against which to compare our proposed method, we implemented two additional strategies. The first is a *simple simulated annotator* strategy, similar to DeepIGeoS and DeepIGeoS-V2, in which misclassified regions are identified and points are sampled randomly from those regions. We chose this baseline because it reflects a commonly used setup in prior work and therefore provides a meaningful reference point for evaluating our method.

The second is a *balanced simulated annotator* strategy, in which the number of annotated voxels is kept equal for both the point-based and scribble-based methods. We included this baseline to make the comparison between these interaction types fairer, since otherwise scribbles may have an inherent advantage by covering more voxels. This allows us to distinguish the effect of annotation type from the effect of annotation quantity.

Simple Simulated Annotator

Like the annotator in DeepIGeoS and DeepIGeoS-V2, we calculate the misclassified regions and randomly sample points from those regions. Similar to their work we limit the amount of points to sample by looking at the amount of misclassified pixels. For a slice in the dataset, we calculate the amount of misclassified pixels as follows:

$$\begin{aligned} \mathrm{M} &= \sum_{i=1}^{X} \left| y_i^{gt} \oplus y_i^{pred} \right| \\ n_{\text{points}} &= \begin{cases} 0 & \text{if } M < 30 \\ \lceil M/100 \rceil & \text{otherwise} \end{cases} \end{aligned} \tag{4.1}$$

Due to the resampling, cropping and differing sizes of our datasets compared to the ones used in DeepIGeoS and DeepIGeoS-V2, we found that the parameter in the second line of the equation needed to be adjusted such that a visually sufficient amount of points would be placed. By trial and error we ended on the value of 30.

Balanced simulated annotator

During training, the simulated annotators used to provide user interactions are varying in the amount of annotated pixels they provide. This means that usually the scribble based methods will have a higher amount of annotated pixels than the point based methods. To counter any bias that might be introduced to the models by the difference in information provided during

training, we also introduce a *Balanced simulated annotator* that provides an equal amount of annotated voxels as our scribble based annotator.

Unfortunately using the default point generation strategy will often result in there not being enough misclassified pixels in a slice to generate the required amount of points. Therefore we allow the annotator to also generate points inside the correctly classified regions, putting foreground annotations in the foreground region and vice versa. This results in a point based annotator that has the same amount of annotated pixels as the scribble based annotator, thus providing both models with the same amount of information (by the metric of annotated voxels) during training.

An example of the three simulated user interaction strategies is shown in figure 4.1. The first image shows our approach, the second image shows point based annotations with a similar amount of annotated pixels as the scribble based annotations and the third image shows the simple point based annotations as used in previous work.

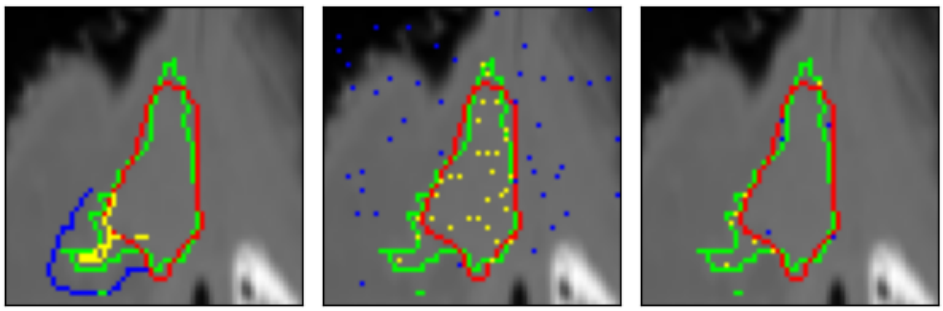


Figure 4.1: Example of the three simulated user interaction strategies. The first image shows our approach, the second image shows point based annotations with a similar amount of annotated pixels as the scribble based annotations and the third image shows the simple point based annotations as used in previous work.

4.2. Test Set

To evaluate and compare the performance of the point-based and scribble-based annotation methods, we require a test set that contains not only the CT images and ground-truth segmentations, but also the corresponding base segmentations and user interactions that are expected to improve the refinement result.

Unfortunately, the MICCAI HaN test set does not include such point or scribble annotations. As a result, it cannot directly be used to evaluate how different forms of user interaction affect refinement performance. We therefore chose to construct our own annotated test set. This decision was necessary to enable a controlled and meaningful comparison between the different refinement methods, since without user annotations it would not be possible to measure how effectively each method incorporates corrective feedback.

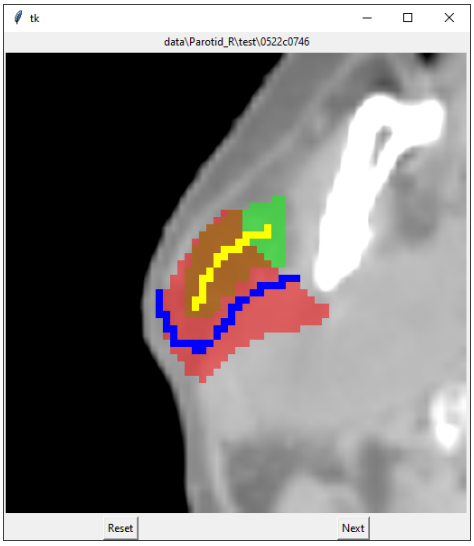


Figure 4.2: Screenshot of the application used to annotate the test set. Green represents an undersegmentation, red an oversegmentation and orange a correct segmentation. The fore and background scribbles are shown in yelow and blue respectively.

Test Set Creation

The MICCAI HaN test dataset, which contains 10 images, offers a limited sample size for robust evaluation. To increase the number of test samples and thereby achieve more accurate and reliable results, we employ a strategy that doubles the dataset size by considering both parotid glands for each patient. This doubles the initial test set to 20 samples with corresponding ground truth and base segmentations.

To supply the annotations to the test set for the purposes of interactive refinement, we developed a custom annotation application. This tool streamlines the annotation process, allowing for quick and efficient annotation of the entire dataset. Figure 4.2 shows a screenshot of the application interface, where users can navigate through selected slices and annotate both foreground and background regions using scribbles.

The selection of slices for annotation was performed systematically. Slices were ranked based on the total number of misclassified voxels in them. The five slices with the highest number of misclassified voxels were selected as candidates for manual annotation by the user. This careful selection process ensures that the slices that are most in need of refinement are prioritized for annotation.

Each annotator contributed annotations for 5 slices per sample, resulting in 100 human-annotated slices in total per annotator. For consistency and reliability, we ultimately used annotations from three annotators, which yielded a final test set comprising 300 user annotations across 20 CT volumes.

Figure 4.3 showcases some annotated samples from the test set, illustrating the variety of the annotations.

4.3. Metrics

In this section, we outline the evaluation metrics employed to assess the performance of our segmentation refinement task. We utilize two widely recognized metrics: the Dice coefficient and the Hausdorff distance. Each metric offers unique insights into different aspects of segmentation quality.

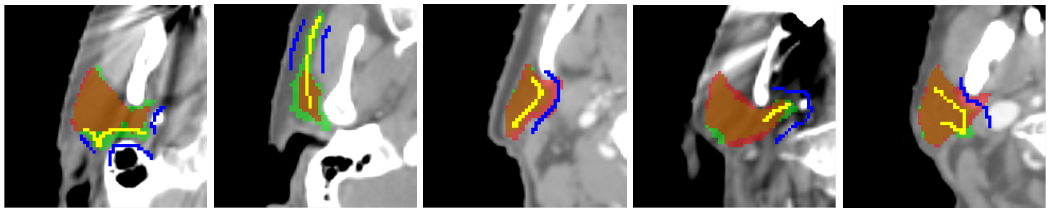


Figure 4.3: Human annotated samples from the test set.

Dice Coefficient

The **Dice coefficient** quantifies the overlap between the predicted segmentation and the ground truth segmentation. Mathematically, it is defined as:

$$\text{Dice} = \frac{2 \times |\text{TP}|}{2 \times |\text{TP}| + |\text{FP}| + |\text{FN}|}$$

where TP represents the number of true positive pixels (correctly identified pixels belonging to the structure of interest), FP denotes the false positive pixels (incorrect pixels identified as belonging to the structure of interest), and FN is the false negative pixels (missed pixels that belong to the structure of interest). The Dice coefficient ranges from 0 (no overlap) to 1 (perfect overlap), with higher values indicating better segmentation performance.

Hausdorff Distance

The **Hausdorff distance** measures the extent of maximum deviation between the predicted segmentation and the ground truth segmentation. We have used this in metric as well in our approach to quantify the missegmentation of regions. It quantifies the largest distance from any point in one segmentation to the nearest point in the other segmentation, and is defined as:

$$\text{Hausdorff}(A, B) = \max\{\sup_{a \in A} \inf_{b \in B} d(a, b), \sup_{b \in B} \inf_{a \in A} d(a, b)\}$$

where $d(a, b)$ is the Euclidean distance between points a and b, with A and B representing the sets of points in the predicted and ground truth segmentations, respectively.

Complementary Insights

Both the Dice coefficient and the Hausdorff distance are crucial metrics in the realm of medical image segmentation tasks, offering complementary insights into the algorithm's performance. The Dice coefficient provides a holistic measure of overlap between the predicted and ground truth segmentations, reflecting the overall accuracy. On the other hand, the Hausdorff distance highlights the maximum deviation, giving a sense of boundary precision and the worst-case errors.

4.4. Evaluation Strategies

In segmentation refinement tasks, evaluating the performance of refinement techniques requires robust and informative metrics. To this end, we use standard segmentation metrics, namely the Dice coefficient and the Hausdorff distance. However, these metrics alone may not fully capture the effectiveness of refinement, especially in localized regions of interest.

The Hausdorff distance measures the largest of the minimal distances between two point sets, but in this context it can be misleading. Refinement may alter regions that do not determine the overall maximum distance, meaning that meaningful local improvements can remain undetected when relying only on global metrics. This limitation motivates the use of additional evaluation strategies that explicitly target the regions affected by refinement.

To address this, we define three evaluation strategies:

- **Global Evaluation**: In this strategy, the Hausdorff distance and Dice coefficient are computed over the entire segmentation volume. Although this provides an overall assessment of segmentation quality, localized improvements introduced by refinement may be obscured.
- **Local Evaluation**: For a more focused analysis, the Hausdorff distance and Dice coefficient are computed on and around the slices where refinement is applied. This localized strategy ensures that the metrics better reflect changes in the targeted regions.
- **Hyperlocal Evaluation**: Going one step further, this strategy evaluates the metrics within a small neighborhood around the refined area. By restricting the analysis to a micro-region, it becomes possible to detect subtle improvements that may be diluted in broader evaluations.

The distinction between local and hyperlocal evaluation was chosen to reflect different perspectives on refinement quality. Local evaluation considers the slice as a whole, which is more consistent with how clinicians visually assess segmentation quality in practice. Hyperlocal evaluation instead focuses on the immediate surroundings of the refined region, allowing for a more precise assessment of the direct effect of the refinement itself.

Figure 4.4 shows a schematic overview of these evaluation strategies and the regions considered in each case.

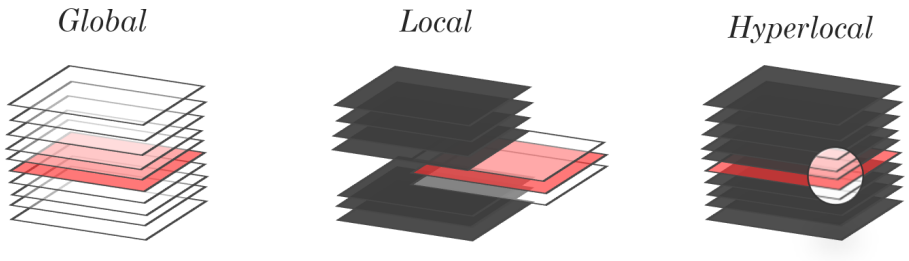


Figure 4.4: Illustration of the different evaluation strategies. Red indicates the annotated slice, while the volume in white represents the voxels included in the metric calculation.

4.5. Experimental Setup

This section describes the experimental setup used to train and evaluate the refinement models. The experiments are designed to test the central hypothesis of this thesis: that reducing the mismatch between training-time and deployment-time annotations leads to more effective interactive refinement. To isolate the effect of annotation type as clearly as possible, all compared models are trained and evaluated under the same protocol, differing only in the way the corrective annotations are generated.

All experiments are conducted on the MICCAI Head and Neck (HaN) dataset introduced in Section 3.1. The provided training set is used for model training, while evaluation is performed on the designated test set. In addition, all models use the same preprocessing and data augmentation pipeline described earlier, so that differences in performance cannot be attributed to differences in input preparation.

Training Configuration

Unless stated otherwise, all refinement models are trained for 300 epochs with a batch size of 5. Optimization is performed with Adam, using a learning rate of 0.001 and a weight decay of

0.001. These settings are kept fixed across all experiments. This choice ensures a controlled comparison between models and avoids introducing unnecessary variation through different optimization schedules or hyperparameter tuning.

The compared models differ only in the form of the simulated interaction used during training. More specifically, we compare a point-based strategy, a balanced point-based strategy, and the proposed scribble-based strategy. By keeping all other components fixed, any differences in performance can be interpreted as the effect of the annotation geometry and its correspondence to realistic deployment-time interactions.

Evaluation Consistency

The same evaluation protocol is applied to all trained models. Performance is assessed using the Dice coefficient and Hausdorff distance under the three evaluation strategies introduced in the previous section: global, local, and hyperlocal evaluation. Using this shared evaluation framework makes it possible to distinguish between overall segmentation quality, slice-level improvement, and improvement in the immediate neighborhood of the refinement itself.

Overall, this setup provides a controlled and fair comparison between annotation strategies. Because the preprocessing, architecture, optimization, loss formulation, and evaluation procedure are all held constant, the experimental results can be attributed as directly as possible to the variable of interest: the form of the user annotation used during training.

5
Results

This chapter presents the results of the experimental comparison between the three refinement strategies introduced in the previous chapters: **Scribbles**, **Points**, and **Balanced Points**. The results are presented in two parts. First, we report the quantitative performance of the models using the Dice score and Hausdorff distance under the global, local, and hyperlocal evaluation strategies defined in Chapter 4. These metric-based results provide an overall view of segmentation quality as well as a more targeted view of refinement quality near the annotated regions. It is important to note that we are not comparing the performance of the models against the BaseNet but are comparing the results amongst eachother.

Second, we present qualitative visual results. These examples are included to illustrate how the different models respond to user corrections in practice, how well the refinements propagate to neighbouring slices, and which types of artefacts or failure modes remain visible after refinement. Together, the quantitative and visual results provide a more complete picture of the behaviour of the proposed method.

5.1. Dice Score

Figure 5.1 shows the Dice scores of the three refinement strategies compared in this thesis: the proposed **Scribbles** method, the **Points** method based on DeepIGeoS(-V2), and the **Balanced Points** method introduced earlier.

The main trend visible in Figure 5.1 is that **Scribbles** achieves the highest Dice score across all three evaluation settings. However, the improvement is small. In other words, the proposed scribble-based training strategy has a consistent positive effect on Dice score, but its advantage over **Balanced Points** is limited. This is confirmed by Table 5.1: **Scribbles** has the highest median Dice score in the global, local, and hyperlocal evaluations, yet it is only significantly better than **Balanced Points** in the local evaluation. Compared to **Points**, **Scribbles** performs significantly better in all three evaluation settings.

This pattern is informative. The strongest advantage of **Scribbles** appears in the local evaluation, which is consistent with the central hypothesis of this thesis: reducing annotation-shift should primarily improve refinement quality in and around the region where the user interacts. At the same time, the relatively small gap between **Scribbles** and **Balanced Points** suggests that, for Dice score, the amount of corrective information presented during training may have the biggest impact on accuracy rather than the annotation geometry.

To assess whether these differences are statistically meaningful, we use the Wilcoxon signed-rank test. This test is appropriate because the comparisons are paired: each method is evaluated on the same test samples, so the question of interest is whether the per-sample Dice

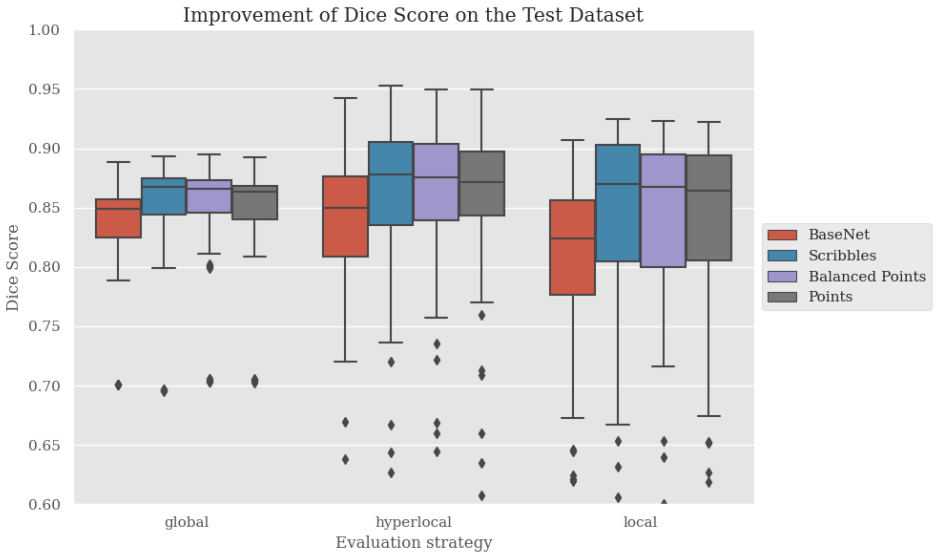


Figure 5.1: Dice score for the three methods that are being compared in this paper.

Table 5.1: Median Dice scores for all methods, together with pairwise Wilcoxon signed-rank test p-values between the refinement methods. Higher Dice is better. Significant p-values ($p < 0.05$) are shown in bold.

	Median Dice				Pairwise p-values		
Evaluation	BaseNet	Points	Balanced Points	Scribbles	S vs BP	S vs P	BP vs P
Global	0.889	0.892	0.887	**0.893**	0.146	**2.5e-6**	**0.019**
Local	0.782	0.806	0.824	**0.827**	**0.012**	**7.2e-4**	0.367
Hyperlocal	0.869	0.882	0.891	**0.897**	0.488	**5.9e-6**	**1.5e-5**

differences between two methods are consistently positive or negative. For the same reason, Table 5.1 reports the median Dice score rather than focusing on the mean.

Looking more closely at the pairwise comparisons, **Scribbles** is not significantly better than **Balanced Points** in the global ($p = 0.146$) or hyperlocal ($p = 0.488$) evaluations, but it is significantly better in the local evaluation ($p = 0.012$). In contrast, **Scribbles** significantly outperforms **Points** in the global, local, and hyperlocal settings. **Balanced Points** also performs significantly better than **Points** in the global and hyperlocal evaluations, but not in the local evaluation.

Overall, the Dice results indicate that scribble-based training provides a small but consistent improvement, with the clearest benefit appearing at the local slice level. This supports the idea that matching training-time interactions more closely to deployment-time scribbles improves refinement quality, although the results also suggest that annotation density and information content remain important factors. There may therefore be a trade-off between annotation geometry and annotation richness, which would be an interesting direction for future work.

5.2. Hausdorff Distance

The second metric used to evaluate the three refinement strategies is the **Hausdorff distance**. In contrast to the Dice score, lower values are better. Figure 5.2 shows the Hausdorff distances for the proposed **Scribbles** method, the **Points** method based on DeepIGeoS(-V2), and the **Balanced Points** method introduced earlier.

The main trend in Figure 5.2 is that both **Scribbles** and **Balanced Points** tend to achieve

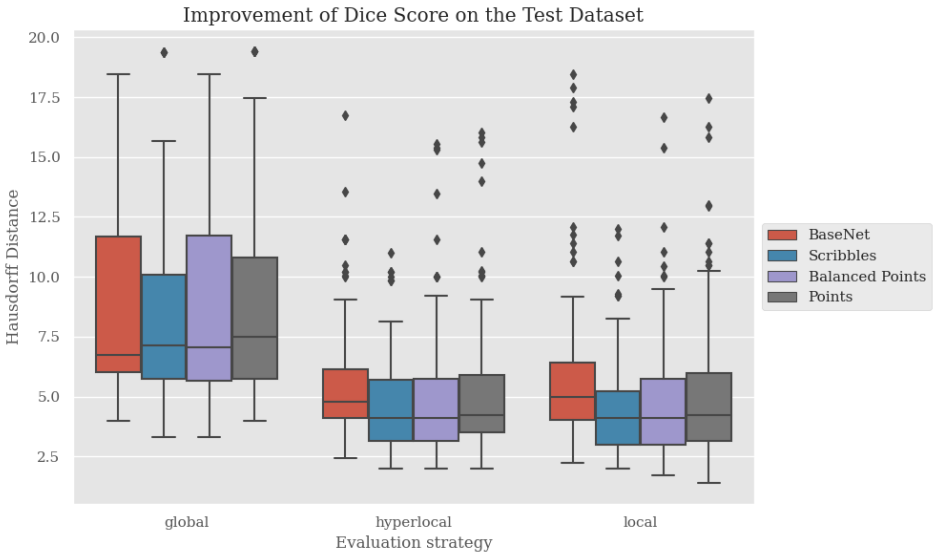


Figure 5.2: Hausdorff distances for the three methods that are being compared in this paper. Lower is better.

lower Hausdorff distances than **Points**, indicating that both methods reduce extreme boundary errors more effectively. However, unlike in the Dice analysis, the difference between **Scribbles** and **Balanced Points** is very small. In fact, Table 5.2 shows that **Scribbles** is not significantly better than **Balanced Points** in any of the three evaluation settings. Compared to **Points**, both **Scribbles** and **Balanced Points** achieve significantly lower Hausdorff distances in the global and hyperlocal evaluations, but not in the local evaluation.

Table 5.2: Median Hausdorff distances for all methods, together with pairwise Wilcoxon signed-rank test p-values between the refinement methods. Lower Hausdorff distance is better. Significant p-values ($p < 0.05$) are shown in bold.

	Median Hausdorff Distance				Pairwise p-values		
Evaluation	BaseNet	Points	Balanced Points	Scribbles	S vs BP	S vs P	BP vs P
Global	**6.164**	6.481	6.403	6.403	0.947	**5.5e-5**	**1.7e-5**
Local	2.828	2.449	2.449	**2.236**	0.959	0.098	0.081
Hyperlocal	3.606	3.606	**3.162**	**3.162**	0.635	**4.8e-5**	**2.5e-5**

Overall, the Hausdorff results support the same general conclusion as the Dice analysis, but in a more conservative way. The proposed scribble-based strategy improves performance relative to the original point-based approach, especially when evaluating extreme errors globally and in the immediate neighbourhood of the refinement. However, it does not show a clear advantage over **Balanced Points**. This suggests that for Hausdorff distance the results again indicate that the amount of corrective information provided during training are more important factors in determining refinement quality.

5.3. Visual Results

Interpretation of visual results

Since the visual results contain a lot of different aspects:

- Foreground scribble
- Background scribble
- Base segmentation
- Refined segmentation
- Ground truth

I have added a guide in figure 5.3 to aid in comparing results of a user interaction along multiple slices with multiple models trained with different annotator strategies. In addition there will be annotations added to highlight important features.

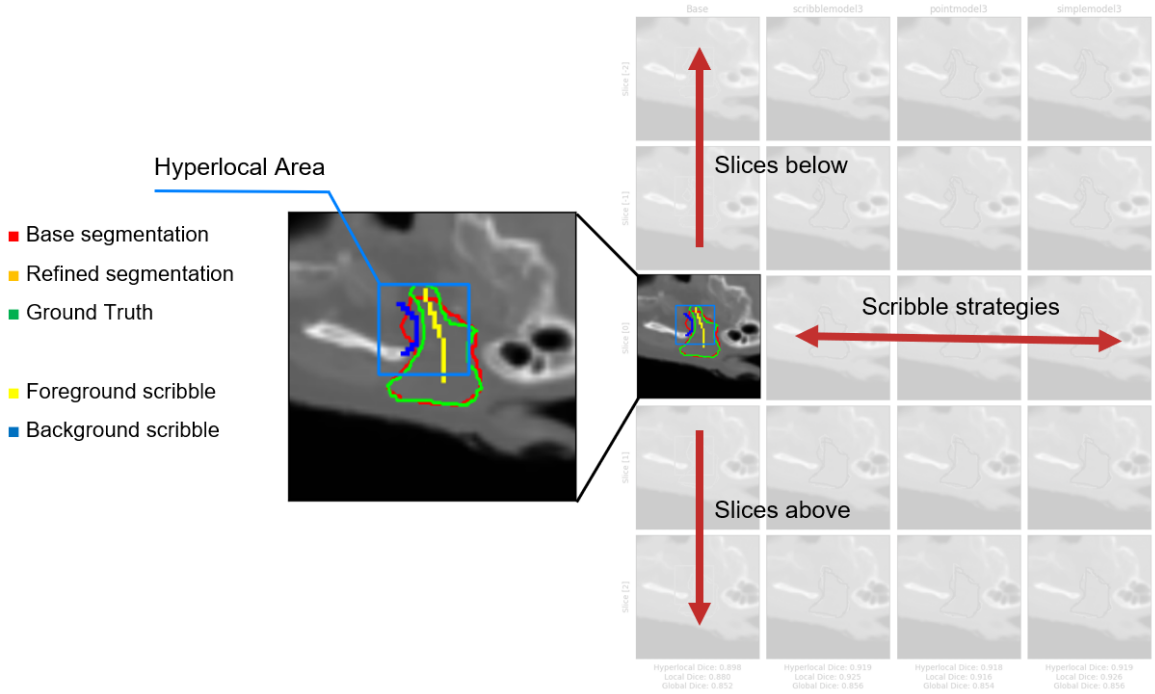


Figure 5.3: How to interpret the visual results

Samples

Addition

Figure 5.4 shows an example in which the user provides a scribble to add a missing protrusion to the segmentation. The most important observation is that the refinement does not only correct the annotated slice itself, but also propagates this correction to neighbouring slices. The example therefore illustrates that the refinement model is capable of incorporating local slice based information and extending it in a spatially meaningful way, rather than only copying the annotation literally into the segmentation. At the same time, the effect remains focused around the annotated region, which is consistent with the intended role of interactive refinement.

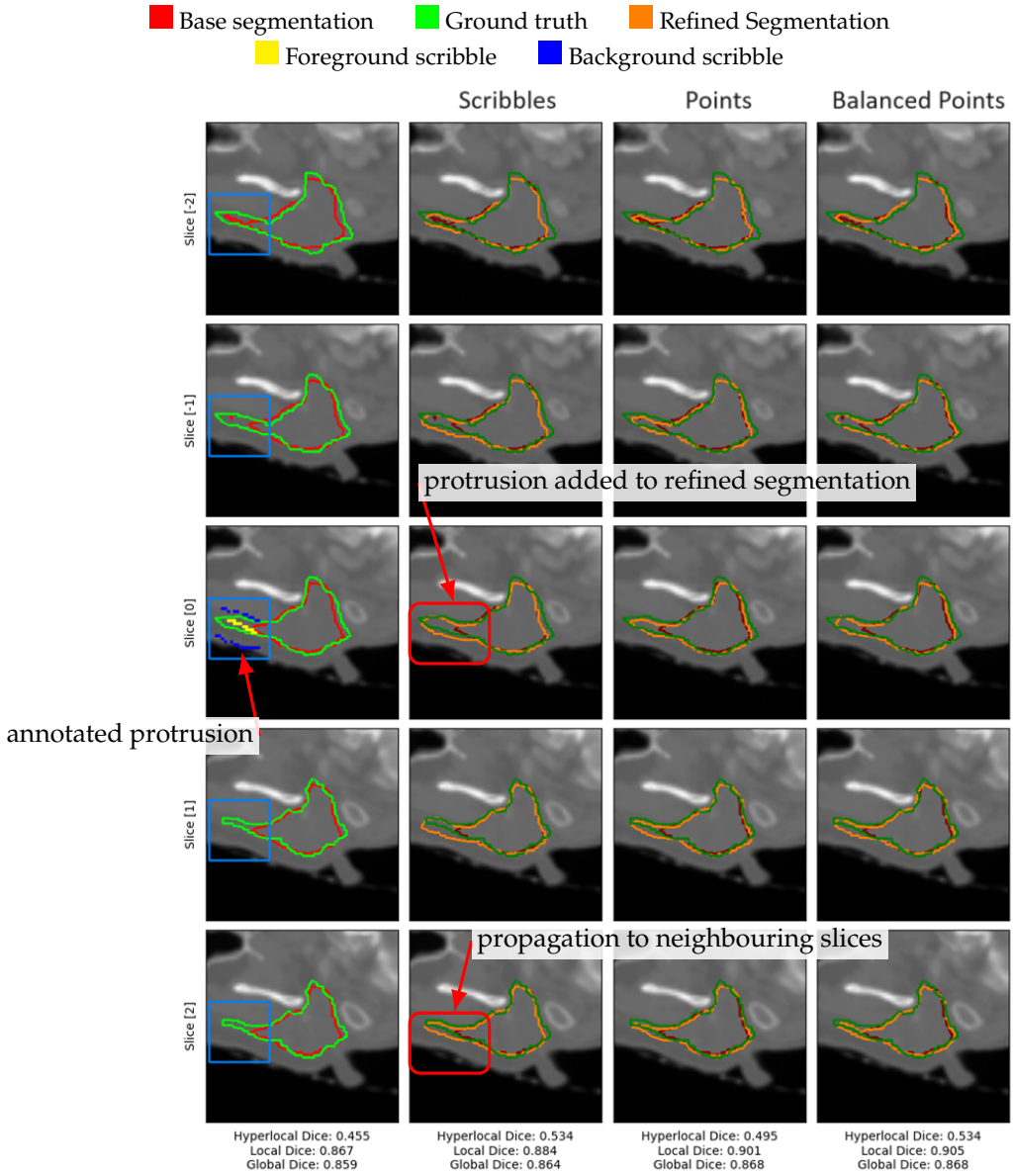


Figure 5.4: A scribble providing information to *add* a region to the segmentation.

Removal

Figure 5.5 shows the opposite situation, where the user interaction is intended to remove an incorrectly segmented region. In this case, the visual result demonstrates that the refinement model is also able to use background information to suppress parts of the segmentation that should not be present. This is an important complement to the addition example, because an interactive refinement method must be able to handle both types of correction: adding missing structure and removing false positives. The figure suggests that the model can perform such deletions in a targeted way, while largely preserving the surrounding structure that was already segmented correctly.

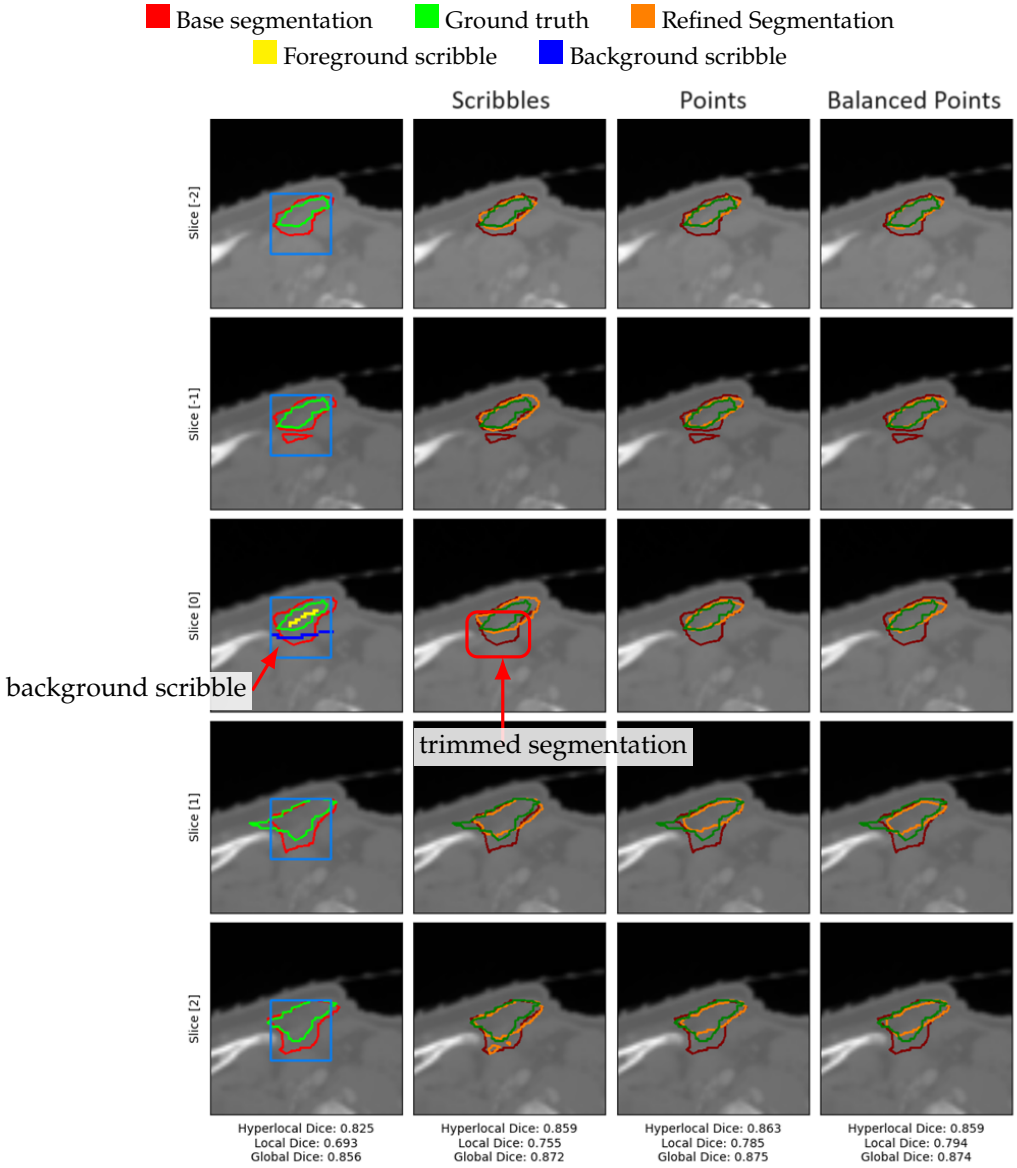


Figure 5.5: A scribble providing information to *remove* a region to the segmentation

Limits to propagation

Figure 5.6 further shows that large deletion-based corrections may leave behind disconnected residual components. This suggests that the refinement model can apply the local correction itself, but is less effective at restoring global shape consistency afterwards. Such artifacts are qualitatively important, since they reveal a limitation that is not always immediately visible in the quantitative metrics alone.

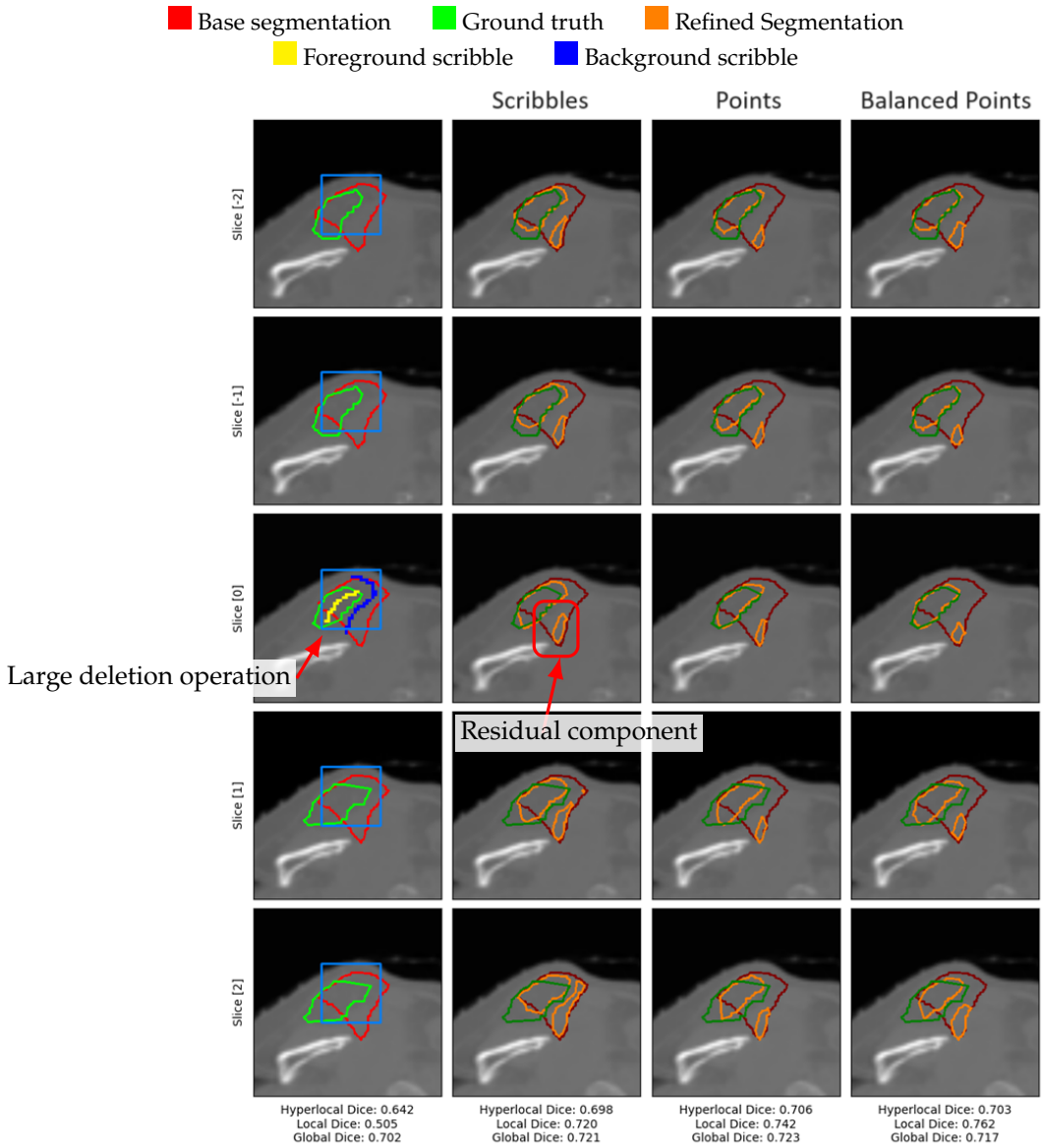


Figure 5.6: A scribble providing information to *remove* a region causing residual components to be left behind

6

Discussion

This thesis set out to study a specific form of train-test mismatch in interactive segmentation refinement, the mismatch between the point-based interactions commonly used during training and the scribble-based interactions that are more likely to be provided by human users at deployment. The results provide partial support for the central hypothesis. Training with simulated scribbles led to the best overall performance, and it consistently outperformed the original point-based baseline. At the same time, the gap between Scribbles and Balanced Points was small, and in several evaluations not statistically significant. The main conclusion is therefore not that scribbles are universally superior, but that aligning training-time interactions with deployment-time behavior is beneficial, while the amount of corrective information in the form of annotated voxels remain at least as important as the interaction geometry.

6.1. Interpretation of the Main Findings

The most important result of this work is that the proposed scribble-based strategy improves refinement performance relative to the original point-based approach, especially in the Dice-based analysis and particularly at the local level. This is meaningful, because the local evaluation is the setting that most directly reflects the intended role of an interactive refinement model, it is improving the segmentation near the region that the user explicitly corrected based on the annotated slice. That suggests that reducing annotation-shift helps the network respond more appropriately to the type of input it will see in practice.

At the same time, the improvement remains small. The goal of this work was not to maximize raw performance by any means, but to isolate the effect of annotation strategy under controlled conditions. Because the pre-processing, architecture, optimization, loss formulation, and evaluation protocol were held constant across experiments, the observed differences can be interpreted primarily as the effect of the annotation modality used during training. The fact that Scribbles clearly improves over Points shows that the mismatch between training and deployment matters. The fact that Scribbles only slightly improves over Balanced Points shows that the interaction geometry is only part of the story.

A plausible interpretation is that two factors contribute simultaneously to refinement quality. The first is *annotation geometry*: scribbles better resemble real human corrections and therefore reduce the train-test mismatch in the guidance channels. The second is *annotation richness*: scribbles usually cover more voxels and provide more spatially structured information than sparse point prompts. The Balanced Points baseline was introduced specifically to separate these effects. Since Balanced Points often performs close to Scribbles, the results suggest that much of the gain may come from exposing the model to a denser and more

informative correction signal during training. However, because equal voxel count does not imply equal information content, the remaining advantage of Scribbles, especially in the local Dice evaluation, still indicates that geometry matters.

6.2. Why the Performance Gap Remained Limited

An important reason why the gains from refinement remain relatively small is that this thesis deliberately starts from a strong initial segmentation. This was a conscious methodological choice. If the base segmentation is weak, almost any refinement strategy can appear highly successful simply because there is a large amount of error left to correct. Such a setup would make it difficult to determine whether the proposed method genuinely improves interactive refinement or merely benefits from an artificially easy starting point. By using a strong BaseNet, the evaluation becomes more stringent and more clinically realistic: the refinement model is tested on the harder problem of making targeted corrections to an already reasonable segmentation.

This choice, however, also reduces the available room for improvement. Once the base segmentation is already close to the ground truth, global metrics such as Dice and Hausdorff naturally become less sensitive to local corrections. This is exactly why the distinction between global, local, and hyperlocal evaluation is important. A method designed for interactive correction should not be judged only by how much it changes the whole volume, but also by how well it improves the specific region that the user marked. The stronger local gains therefore support the relevance of the proposed method even when the global gains are limited.

Because refinement is designed to make a targeted local correction rather than re-segment the entire organ, large global gains should not necessarily be expected from a single user interaction. In our test setting, only one annotation task is provided per case, so the realistic effect of refinement is mainly near the annotated region. For that reason, local and hyperlocal evaluation are particularly important: they measure the model where the user actually intervenes and therefore better reflect the intended purpose of interactive refinement. A very large global improvement after a single local annotation might even be undesirable, since it could indicate that the model is making broader, less controlled changes beyond the user-provided correction.

6.3. Justification of the Main Design Choices

6.3.1. Single-organ focus

The decision to focus on the left and right parotid glands was made to isolate the research question as clearly as possible. The thesis is not primarily about multi-organ segmentation, but about annotation-shift in interactive refinement. Restricting the task to a single organ type reduces the number of confounding factors, such as extreme class imbalance between organs, variation in anatomical scale, and differences in boundary visibility across structures. This makes it easier to attribute differences in performance to the interaction strategy rather than to unrelated sources of variation.

At the same time, this choice limits generalizability. A model trained only on parotid glands may learn a strong organ-specific prior, which could reduce the extent to which it truly relies on the user input. In other words, part of the refinement behavior may still come from knowing what a parotid gland 'should' look like, rather than from correctly interpreting the annotation itself. This does not invalidate the results, but it does mean that future work should test whether the same conclusions hold in a multi-organ setting and on structures with different size, contrast, and shape variability.

6.3.2. Slice-wise scribble simulation

The scribble simulator was designed to operate on individual slices and to target large error regions with freeform, perturbed line annotations. This was justified by the intended deployment setting. In clinical workflows, users commonly inspect and correct volumetric scans through two-dimensional slice views rather than by drawing directly in 3D. Generating slice-wise scribbles therefore makes the training signal more representative of realistic user behavior.

The final scribble-generation strategy was also chosen because it strikes a balance between realism and controllability. Ranking error regions by their severity ensures that annotations are placed where they are most useful. Constructing scribbles relative to the ground-truth contour makes the foreground/background pair anatomically meaningful. The sinusoidal perturbation introduces variation without destroying the continuity of the line. Together, these choices make the generated annotations both interpretable and sufficiently diverse for training.

6.3.3. Gaussian distance encoding

The choice of Gaussian distance maps instead of more elaborate context-aware alternatives such as geodesic distance was also deliberate. In theory, context-aware encodings are attractive because they can prevent annotation information from leaking across boundaries. In practice, however, the CT data used in this work contain noise and weak soft-tissue contrast, which made the geodesic formulation sensitive to parameter settings and less robust than expected. The Gaussian encoding offered a better compromise: it propagated annotation information smoothly and locally, while remaining computationally simple and stable. Since the purpose of the encoding is to provide a reliable refinement signal rather than to solve the segmentation problem on its own, prioritizing robustness over theoretical sophistication was justified.

6.3.4. Shallow RefNet and explicit input manipulation

The refinement model was intentionally made smaller than the base model. This follows from the role of the system. BaseNet must infer the full organ from the CT volume alone, whereas RefNet only needs to make targeted corrections given a current segmentation and explicit user guidance. A shallow 3D U-Net is therefore a reasonable design choice. This design is also consistent with the intended workflow of the method, in which the heavy anatomical reasoning is performed by the BaseNet and the RefNet acts primarily as a correction mechanism rather than a second full segmentation network.

At the same time, the shallowness of RefNet is also a likely limitation of the proposed approach. Because the network has a smaller receptive field and lower representational capacity than a deeper architecture, it may be less able to integrate broader anatomical context across the volume. In practice, this means that the model is well suited to correcting errors near the user annotation, but may be less effective when a correction needs to propagate over a larger spatial extent or when the initial segmentation error reflects a more global structural mistake. This interpretation is consistent with the results of this thesis, where the improvements are clearest in the local setting while the visual results show more limited propagation of corrections. In particular, some deletion-based refinements leave behind disconnected residual components or anatomically unlikely extensions of the organ. This suggests that the shallow refinement network is able to respond to the local annotation itself, but is less capable of enforcing broader shape consistency across the full segmentation. In that sense, the model may not fully capture that the target organ should remain a coherent and relatively homogeneous anatomical structure after refinement. A larger or more context-aware refinement network might be better able to recognize and correct such globally inconsistent shapes, although this remains a question for future work.

Similarly, the direct manipulation of the base segmentation using the user annotations was included because the annotations are assumed to be correct. If a user explicitly marks voxels as foreground or background, it is unnecessary and undesirable to ask the network to 'discover' that rule from data. Enforcing the scribbles in the input makes the system more transparent and ensures that the refinement starts from a segmentation that is already consistent with the user feedback. A possible downside is that hard corrections may contribute to artifacts such as disconnected islands, as seen in some visual examples, but this is a reasonable trade-off in an interactive setting where respecting explicit user input should take priority. More broadly, the combination of a shallow refinement network and hard input manipulation reflects the overall philosophy of the method: prioritize responsiveness, transparency, and local correction, even if this comes at the cost of some global corrective power. Future work could therefore investigate whether slightly deeper or multi-scale refinement architectures can preserve interactive speed while improving long-range propagation of corrections, suppressing residual artifacts, and maintaining global anatomical consistency.

6.4. Limitations and Future Work

The main limitation of this study is that it investigates annotation-shift in a relatively controlled setting: one dataset, one organ class, one refinement architecture, and one main form of human interaction. This narrow scope was useful for isolating the variable of interest, but it also means that the conclusions should not be generalized too broadly. Future work should therefore evaluate whether the same pattern holds for other organs-at-risk, for multi-organ refinement, and for datasets with different imaging characteristics.

A second limitation is that the Balanced Points baseline, while useful, is still only an approximation of equal information content. Matching the number of annotated voxels does not guarantee that the provided information is equally informative. Scribbles encode continuity, local direction, and boundary intent in a way that isolated points do not. This means that the comparison between geometry and annotation density is not yet fully resolved. A natural next step would be to design stronger point-based baselines that better control for distribution and structure, or to study hybrid interaction strategies that combine sparse points with short strokes.

Finally, this work suggests that interactive refinement should not be evaluated using only global metrics. Because refinement is a local correction task, future studies should continue to include local and hyperlocal analyses, and perhaps also measure user effort directly. A method that achieves similar global accuracy with fewer or simpler interactions may ultimately be more valuable in practice than one that performs slightly better numerically but requires more complex input.

6.5. Conclusion of the Discussion

Overall, the results of this thesis support a nuanced conclusion. Reducing annotation-shift by training with simulated scribbles improves interactive refinement, particularly when compared with the original point-based baseline and particularly near the region of interaction. However, the small gap between Scribbles and Balanced Points shows that annotation geometry is not the only important factor; annotation richness and spatial coverage also play a major role. The contribution of this work is therefore twofold: it demonstrates that the mismatch between training-time and deployment-time annotations matters, and it shows that a fair evaluation of interactive refinement must distinguish between the form of the annotation and the amount of corrective information it conveys.

7

Conclusion

7.1. Main Outcome

In this thesis, we investigated annotation-shift in interactive segmentation refinement for head-and-neck radiotherapy planning. More specifically, we studied the mismatch between the point-based interactions commonly used during training and the scribble-based interactions more likely to be provided by human users at test time. To address this mismatch, we proposed a training pipeline based on simulated human-like scribbles, combined with annotation-aware supervision, distance-map encodings, and explicit input manipulation of the initial segmentation.

The results show that reducing this mismatch is beneficial, but in a small way. The proposed Scribbles strategy consistently outperformed the original Points baseline, with the clearest gains appearing in the local evaluation. This supports the central hypothesis of the thesis: making the training interactions more similar to deployment-time scribbles improves the ability of the refinement model to respond to user input where it matters most. At the same time, the difference between Scribbles and Balanced Points was generally small. This indicates that the geometry of the annotation is not the only important factor; the amount, density, and spatial distribution of corrective information also play a major role in refinement quality.

7.2. What This Thesis Contributes

A key contribution of this work is therefore not only the proposed scribble simulator itself, but also the evaluation of segmentation improvement in interactive refinement. The thesis set out to test whether aligning training-time interactions with deployment-time human scribbles would reduce annotation-shift and yield more reliable refinements. The results support that claim, but they also show that future work should take into account the difference between annotation modality and annotation richness. In that sense, this work contributes both a practical refinement strategy and a clearer framework for evaluating interactive segmentation methods.

7.3. Methodological Insights

At the methodological level, several design choices proved justified. The slice-wise scribble generation procedure provided a realistic and controllable approximation of human annotation behavior, while the two-stage BaseNet-RefNet setup matched the intended interactive workflow well: BaseNet produces a strong initial segmentation and RefNet performs targeted correction. The shared evaluation protocol with global, local, and hyperlocal analysis also made it possible to distinguish overall segmentation quality from improvement near the region of interaction.

At the same time, the results show that interactive refinement should be interpreted conservatively. Because refinement is intended to make a local correction rather than re-segment the full organ, large global gains should not necessarily be expected from a single user interaction. This is exactly why local and hyperlocal evaluation are valuable in this setting as they better reflect the intended purpose of the model than global metrics alone.

7.4. Limitations

This study was intentionally narrow in scope. It focused on one dataset, one organ class, one main refinement architecture, and one primary form of human interaction. That controlled setup was useful for isolating the effect of annotation strategy, but it also limits how broadly the conclusions can be generalized. The small difference between Scribbles and Balanced Points also shows that matching only the number of annotated voxels does not create a similar score. Scribbles still contain information that isolated points do not, such as continuity and a clearer indication of the boundary.

7.5. Future Work

Future work should test whether the same conclusions hold in multi-organ segmentation, on other imaging datasets, and with other forms of user interaction. It would also be valuable to explore refinement architectures that preserve interactive speed while improving the propagation of corrections over larger spatial extents. Also, future work in the field of annotation encoding should also evaluate a point-based annotator that provides a similar number of annotated voxels as the scribble-based annotator. Our results suggest that the amount of information in the annotation plays an important role, so comparing scribbles only to sparse points does not fully isolate the effect of annotation type. Such a baseline would make it easier to determine whether improvements come from the scribble shape itself or simply from providing more corrective information. Another interesting topic for future work is whether there is a limit to how much user information should be given to the refinement model. In this thesis, more annotation information generally led to better results, but it is not yet clear whether this will always remain true. At some point, extra annotations may give only limited additional benefit, or could even hurt performance if the model becomes too dependent on dense local corrections. Investigating this would help determine the optimal amount of user input for interactive refinement.

7.6. Final Conclusion

Overall, this thesis shows that annotation-shift is a meaningful problem in interactive medical image segmentation and that simulated scribbles are a useful way to address it. Training with scribble-like interactions improves refinement relative to the original point-based baseline, especially near the region where the user intervenes. At the same time, the limited gap to Balanced Points shows that the success of interactive refinement depends not only on matching annotation style, but also on how much corrective information is provided.

References

[1] Patrik F Raudaschl et al. "Evaluation of segmentation methods on head and neck CT: Auto-segmentation challenge 2015". en. In: *Med. Phys.* 44.5 (May 2017), pp. 2020–2036.

[2] K Kian Ang et al. "Randomized phase III trial of concurrent accelerated radiation plus cisplatin with or without cetuximab for stage III to IV head and neck carcinoma: RTOG 0522". en. In: *J. Clin. Oncol.* 32.27 (Sept. 2014), pp. 2940–2950.

[3] Geert Litjens et al. "A Survey on Deep Learning in Medical Image Analysis". In: *CoRR* abs/1702.05747 (2017). arXiv: 1702.05747. URL: http://arxiv.org/abs/1702.05747.

[4] Yunhe Gao et al. *FocusNet: Imbalanced Large and Small Organ Segmentation with an End-to-End Deep Neural Network for Head and Neck CT Images*. 2019. DOI: 10.48550/ARXIV.1907.12056. URL: https://arxiv.org/abs/1907.12056.

[5] Prerak Mody et al. *Comparing Bayesian Models for Organ Contouring in Head and Neck Radiotherapy*. 2021. DOI: 10.48550/ARXIV.2111.01134. URL: https://arxiv.org/abs/2111.01134.

[6] Y.Y. Boykov and M.-P. Jolly. "Interactive graph cuts for optimal boundary amp; region segmentation of objects in N-D images". In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. Vol. 1. 2001, 105–112 vol.1. DOI: 10.1109/ICCV.2001.937505.

[7] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "GrabCut -Interactive Foreground Extraction using Iterated Graph Cuts". In: *ACM Transactions on Graphics (SIGGRAPH)* (Aug. 2004). URL: https://www.microsoft.com/en-us/research/publication/grabcut-interactive-foreground-extraction-using-iterated-graph-cuts/.

[8] Jean-Luc Starck, Emmanuel J Candes, and David L Donoho. "Markov random fields and their applications". In: *IEEE Signal Processing Magazine* 23.1 (2006), pp. 149–159.

[9] John Lafferty, Andrew Mccallum, and Fernando Pereira. "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". In: Jan. 2001, pp. 282–289.

[10] Shuai Zheng et al. "Conditional Random Fields as Recurrent Neural Networks". In: *CoRR* abs/1502.03240 (2015). arXiv: 1502.03240. URL: http://arxiv.org/abs/1502.03240.

[11] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *CoRR* abs/1411.4038 (2014). arXiv: 1411.4038. URL: http://arxiv.org/abs/1411.4038.

[12] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 3431–3440.

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[14] Özgün Çiçek et al. “3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation”. In: *CoRR* abs/1606.06650 (2016). arXiv: 1606.06650. URL: http://arxiv.org/abs/1606.06650.

[15] Özgün Çiçek et al. “3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2016, pp. 424–432.

[16] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. “V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation”. In: *CoRR* abs/1606.04797 (2016). arXiv: 1606.04797. URL: http://arxiv.org/abs/1606.04797.

[17] Chaitanya Kaul, Suresh Manandhar, and Nick E. Pears. “FocusNet: An attention-based Fully Convolutional Network for Medical Image Segmentation”. In: *CoRR* abs/1902.03091 (2019). arXiv: 1902.03091. URL: http://arxiv.org/abs/1902.03091.

[18] Jörg Sander, Bob De Vos, and Ivana Išgum. “Automatic segmentation with detection of local segmentation failures in cardiac MRI”. In: *Scientific Reports* 10 (Dec. 2020). DOI: 10.1038/s41598-020-77733-4.

[19] Guotai Wang et al. “DeepIGeoS: A Deep Interactive Geodesic Framework for Medical Image Segmentation”. In: *CoRR* abs/1707.00652 (2017). arXiv: 1707.00652. URL: http://arxiv.org/abs/1707.00652.

[20] Guotai Wang et al. “Interactive Medical Image Segmentation Using Deep Learning With Image-Specific Fine Tuning”. In: *IEEE Transactions on Medical Imaging* 37 (Jan. 2018), pp. 1562–1573. DOI: 10.1109/TMI.2018.2791721.

[21] Wenhui Lei et al. “DeepIGeoS-V2: Deep Interactive Segmentation of Multiple Organs from Head and Neck Images with Lightweight CNNs”. In: *Large-Scale Annotation of Biomedical Data and Expert Label Synthesis and Hardware Aware Learning for Medical Imaging and Computer Assisted Intervention*. Ed. by Luping Zhou et al. Cham: Springer International Publishing, 2019, pp. 61–69. ISBN: 978-3-030-33642-4.

[22] Guotai Wang et al. “Slic-Seg: A Minimally Interactive Segmentation of the Placenta from Sparse and Motion-Corrupted Fetal MRI in Multiple Views”. In: *Medical Image Analysis* 34 (May 2016). DOI: 10.1016/j.media.2016.04.009.

[23] Xiangde Luo et al. “MIDeepSeg: Minimally Interactive Segmentation of Unseen Objects from Medical Images Using Deep Learning”. In: *CoRR* abs/2104.12166 (2021). arXiv: 2104.12166. URL: https://arxiv.org/abs/2104.12166.

[24] Guotai Wang et al. “Uncertainty-Guided Efficient Interactive Refinement of Fetal Brain Segmentation from Stacks of MRI Slices”. In: *Lecture Notes in Computer Science* (2020), pp. 279–288. ISSN: 1611-3349. DOI: 10.1007/978-3-030-59719-1_28. URL: http://dx.doi.org/10.1007/978-3-030-59719-1_28.

[25] Ervine Zheng et al. “A Continual Learning Framework for Uncertainty-Aware Interactive Image Segmentation”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.7 (May 2021), pp. 6030–6038. DOI: 10.1609/aaai.v35i7.16752. URL: https://ojs.aaai.org/index.php/AAAI/article/view/16752.

[26] Ning Xu et al. *Deep Interactive Object Selection*. 2016. DOI: 10.48550/ARXIV.1603.04042. URL: https://arxiv.org/abs/1603.04042.

[27] Eirikur Agustsson, Jasper R. R. Uijlings, and Vittorio Ferrari. “Interactive Full Image Segmentation”. In: *CoRR* abs/1812.01888 (2018). arXiv: 1812.01888. URL: http://arxiv.org/abs/1812.01888.

[28] Antonio Criminisi, Toby Sharp, and Andrew Blake. "GeoS: Geodesic Image Segmentation". In: vol. 5302. Oct. 2008, pp. 99–112. ISBN: 978-3-540-88681-5. DOI: 10.1007/978-3-540-88682-2_9.

[29] Muhammad Asad, Reuben Dorent, and Tom Vercauteren. "FastGeodis: Fast Generalised Geodesic Distance Transform". In: *Journal of Open Source Software* 7.79 (2022), p. 4532. DOI: 10.21105/joss.04532. URL: https://doi.org/10.21105/joss.04532.

[30] Michael Roberts, Ke Chen, and Klaus Irion. "A Convex Geodesic Selective Model for Image Segmentation". In: *Journal of Mathematical Imaging and Vision* 61 (May 2019). DOI: 10.1007/s10851-018-0857-2.

[31] Yawei Luo et al. "Taking A Closer Look at Domain Shift: Category-level Adversaries for Semantics Consistent Domain Adaptation". In: *CoRR* abs/1809.09478 (2018). arXiv: 1809.09478. URL: http://arxiv.org/abs/1809.09478.

[32] Brian Guo et al. *The Impact of Scanner Domain Shift on Deep Learning Performance in Medical Imaging: an Experimental Study*. 2024. arXiv: 2409.04368 [eess.IV]. URL: https://arxiv.org/abs/2409.04368.

[33] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. *MNIST handwritten digit database*. http://yann.lecun.com/exdb/mnist/. 2010.

[34] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira et al. Curran Associates, Inc., 2012, pp. 1097–1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[36] Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 779–788.

[37] Aaron Defazio, Francesco Visin, and Yoshua Bengio. "FlipOut: Efficient Pseudo-Independent Weight Perturbations on Mini-Batches". In: *International Conference on Learning Representations* (2018).

[38] Guotai Wang et al. "Uncertainty-Guided Efficient Interactive Refinement of Fetal Brain Segmentation from Stacks of MRI Slices". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*. Springer International Publishing, 2020, pp. 279–288. DOI: 10.1007/978-3-030-59719-1_28. URL: https://doi.org/10.1007%2F978-3-030-59719-1_28.

[39] Andrew Top, Ghassan Hamarneh, and Rafeef Abugharbieh. "Active Learning for Interactive 3D Image Segmentation". In: vol. 14. Sept. 2011, pp. 603–10. ISBN: 978-3-642-23625-9. DOI: 10.1007/978-3-642-23626-6_74.

[40] Ning Xu et al. "Deep Interactive Object Selection". In: *CoRR* abs/1603.04042 (2016). arXiv: 1603.04042. URL: http://arxiv.org/abs/1603.04042.

[41] Annika Hänsch et al. "Evaluation of deep learning methods for parotid gland segmentation from CT images". In: *Journal of Medical Imaging* 6.1 (2019), p. 011005. DOI: 10.1117/1.JMI.6.1.011005.

[42] Wentao Zhu et al. "AnatomyNet: Deep learning for fast and fully automated whole-volume segmentation of head and neck anatomy". In: *Medical Physics* 46.2 (2019), pp. 576–589. DOI: 10.1002/mp.13300.

[43] Murdock G. Grewar, Glenn R. Myers, and Andrew M. Kingston. "Least-squares and maximum-likelihood in computed tomography". In: *Journal of Medical Imaging* 9.3 (May 2022), p. 031508. DOI: 10.1117/1.JMI.9.3.031508.

[44] Evgin Goceri. "Medical image data augmentation: techniques, comparisons and interpretations". In: *Artificial Intelligence Review* (Mar. 2023), pp. 1–45. DOI: 10.1007/s10462-023-10453-z.

[45] Bangyu Luo et al. "Study on Automatic Contour Delineation of Organs at Risk in Intensity-Modulated Radiotherapy for Nasopharyngeal Carcinoma". In: *Digital Medicine* 11.1 (2025), e24–00009. DOI: 10.1097/DM-2024-00009.

[46] Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. "Large-scale interactive object segmentation with human annotators". In: *CoRR* abs/1903.10830 (2019). arXiv: 1903.10830. URL: http://arxiv.org/abs/1903.10830.

[47] J. E. Bresenham. "Algorithm for computer control of a digital plotter". In: *IBM Systems Journal* 4.1 (1965), pp. 25–30. DOI: 10.1147/sj.41.0025.