UNIVERSITY OF COPENHAGEN
DEPARTMENT OF COMPUTER SCIENCE

Deep-Learning Image Segmentation with Corrective Annotation

Abraham George Smith

Submitted at the Faculty of Science on the 2nd of June 2023
University of Copenhagen
Principle Supervisor: Jens Petersen
Assistant Supervisors: Ivan Richter Vogelius and Sune Darkner

Contents

Abstract

Deep learning segmentation has been found to be effective and accurate for a variety of tasks including auto-contouring for radiotherapy and root phenotyping for agricultural and plant physiological research. Prior work has shown that existing models may not generalise well to new datasets, resulting in an unmet need for a method allowing clinical personnel and biological research labs to conveniently train their own models for new segmentation tasks.

We aimed to investigate the effectiveness of Interactive Machine Learning (IML) for the rapid training of models for use in both plant science and radiotherapy. IML puts the human-in-the-loop in the model training process, allowing them to directly observe and influence the characteristics of the model whilst annotating, yet there is little research investigating the potential of IML for deep learning segmentation model training.

We implemented and investigated various corrective-annotation approaches for multiple segmentation tasks and found that IML via corrective-annotation provides a way to rapidly train fully-automatic models for diverse datasets in plant image segmentation in under two hours of interactive training. In the field of radiotherapy, we found the IML process can be directly utilised for task completion, with the model and humans collaborating whilst the model learns from human feedback. The auto-contouring model was found to continuously provide improvements in contouring time for a heart contouring task, eventually out-performing the existing clinical delineation software. We also quantified agreement of the generated heart contours in terms of radiation dose, finding that for the vast majority of automatic contours there is little deviation between the model predicted and human corrected contours, yet in some rare cases there can be large differences with potentially catastrophic consequences, highlighting the need for continuous monitoring and guidance from a human observer, an intrinsic property of the IML Workflow for task completion.

We revealed new dynamics related to IML model training, finding the combination with interactive-segmentation can exacerbate what is known as the cold-start problem, leading to a longer period of interactive training before the approach becomes competitive with manual correction of contours. In spite of this slower convergence the combined method demonstrates a capability that eventually out-performs the more manual alternative.

The IML methods in the thesis are evaluated using real world image segmentation tasks with real human annotators, enabling immediate translation to several subsequent image-analysis based research projects in various domains.

Abstract in Danish

Deep learning segmentering har vist sig at være effektiv og præcis til mange forskellige opgaver, herunder auto-konturering til strålebehandling og rod fænotyping til landbrugs- og plantefysiologisk forskning. Tidligere arbejde har vist, at eksisterende modeller muligvis ikke generaliserer godt til nye datasæt, hvilket resulterer i et udækket behov for en metode der giver klinisk personale og biologiske forskningslaboratorier mulighed for nemt at træne deres egne segmenteringsmodeller.

Vores formål var at undersøge effektiviteten af interaktiv maskinel indlæring (IML) til hurtig træning af modeller til brug i både plantevidenskab og strålebehandling. IML bruger human-in-the-loop i model træningsprocessen, hvilket giver brugeren mulighed for direkte at observere og påvirke modellens egenskaber mens den annotere, men der findes meget lidt forskning der undersøger IMLs potentiale i dyb læring segmenteringsmodel træning.

Vi implementerede og undersøgte forskellige korrigerende annoterings tilgange til en række segmenteringsopgaver og fandt, at IML via korrigerende annotering giver en hurtig måde at træne fuldautomatiske modeller til forskellige datasæt med en segmentering af plante billeder på under to timers interaktiv træning. Inden for stråleterapi, fandt vi at IML-processen kan bruges direkte til segmentering, hvor modellen og mennesker samarbejder, mens modellen lærer af menneskelig feedback. Auto-kontur modellen forbedrede løbende kontureringstiden af hjerter og udkonkurrere til sidst den eksisterende konturerings software i klinikken. Vi kvantificerede også overensstemmelse mellem de genererede hjerte konturer i forhold til strålingsdosis og fandt at der for langt de fleste automatiske konturer er en meget lille afvigelse mellem modellens forudsigelser og de menneskeligt korrigerede konturer. Dog kan der i sjældne tilfælde være store forskelle hvilket potentielt kan have katastrofale konsekvenser for patienten, hvilket fremhæver behovet for løbende overvågning og vejledning fra en menneskelig observatør, en eksisterende egenskab ved IML Workflow. Vi afslørede nye dynamikker relateret til IML-modeltræning og fandt at kombinationen med interaktiv-segmentering kan forværre koldstart-problemet, hvilket medfører en længere interaktiv træningsperiode, før tilgangen bliver konkurrencedygtig i forhold til manuel korrektion af konturer. På trods af den langsommere konvergens, udkonkurrerer den kombinerede metode til sidst det mere manuelle alternativ. IML-metoderne i dette speciale evalueres ved brug af billeder segmenterings opgaver fra den virkelige verden med rigtige menneskelige annotationer, hvilket muliggør øjeblikkelig omsættelse til andre forskningsprojekter inden for forskellige domæner.

Introduction

1.1 Semantic image segmentation

Semantic segmentation involves splitting up an image into one or more regions. For digital imagery, such as 2D photographs consisting of a grid of RGB pixels, that is, red, green and blue values for each discrete point in the image. Segmentation can be considered as a pixel classification task where each pixel is automatically assigned a label corresponding to a semantic category. An application of segmentation is self-driving cars [52], where there is an effort to completely transition the world's fleet of drivers away from having to manually steer their cars, freeing up countless hours of human time. In the agricultural domain, segmentation may be used to identify regions of a field where weeds are most likely to be growing [263, 238], enabling more precise application of herbicide, reducing costs whilst mitigating environmental damage. In medicine, segmentation can detect regions that are more likely to develop into skin cancer [12, 243]. The applications of semantic segmentation to provide benefit to society are practically endless. Any time there is a need to automatically extract information from an image, there's a likelihood that segmentation will be involved, for example as a pre-processing task, for risk scoring of lungs for covid-19 [182].

1.2 Segmentation in radiotherapy

Radiotherapy involves the use of X-rays to kill tumour cells [88], and is effective for many patients in either curative or palliative settings, where it can be used to reduce pain and increase quality of life for patients in their final years. Ultimately, the goal of the radiotherapy treatment planning procedure is to arrive at a patient specific method of radiation delivery that will ensure the prescribed high dose is given to the tumour to eliminate tumour cells, whilst minimising the harmful dose to the surrounding organs [239]. The planning procedure is personalised to each individual patient, involving the creation of a 3D annotated map of a patient's tumour and internal organs [98]. In a radiotherapy context, the process of creating this map is referred to as contouring or delineation but

is functionally equivalent to semantic segmentation. The created contours are used to help guide radiation away from critical organs where there are known associations between radiation exposure and life threatening or debilitating complications [89]. The contouring process is done with the support of one or more imaging modalities, including positron emission tomography (PET), X-ray computed tomography (CT) and magnetic resonance (MR) images in addition to clinical knowledge of both the specific cancer type and patient characteristics.

Delineation is often done manually or semi-automatically, leading to issues typically associated with manual work. There is a degree of intra or inter-annotator variation, in other words, a lack of consistency in the manual contouring process, which, being labour intensive, is also expensive. Delineation is time consuming, requiring extensive training and can lead to bottlenecks that delay treatment, potentially reducing the survival probability of critically ill patients.

Another use case of segmentation in radiotherapy is in research, for example, in the establishment of dose-response relationships, where the response is the likelihood of an outcome for a patient. For example for organs-at-risk, there is a need to understand the association between the amount of radiation given to each structure and the subsequent complications for the irradiated patient.

Dose-response relationships are established using historical or retrospective data and require contours for computing dosimetric parameters which can then be used to establish correlations with different outcomes based on registry records [55] or blood tests [214, 1].

Larger scale dose-response studies are necessary to improve planning procedures, allowing better informed trade-offs to be made [137]. Aznar and Marrazzo [14] also argue there is a need for dose-response studies in lesser studied organs such as the oesophagus and liver.

Creating large datasets of contours has traditionally been a bottleneck, limiting the size and thus statistical power for dose-response studies. There are challenges in getting access to historical patient image data and contouring on images manually is time consuming and expensive, if the organ of interest was not contoured as part of the original planning procedure [232].

Automated methods offer the promise of eliminating the issues with manual contouring, improving consistency, accuracy and enabling the rapid contouring of extremely large datasets of a thousand or more patients with various organs, establishing much-needed dose-response relationships.

The state-of-the-art in auto-contouring today is deep-learning [114], including CNNs or more recently vision-transformer architectures [74, 185]. Deep-learning methods are able to segment 3D scans of many organs in large datasets of patients, labelling each voxel based on its corresponding organ in the visible scan [240].

1.3 Segmentation in root phenotyping

Root phenotyping is the measurement and analysis of plant root characteristics, also known as traits. Examples of root traits include root length and root nodule size or count. Segmentation is required for phenotyping due to the need to automatically quantify traits from images, enabling larger scale experiments [136]. Phenotyping is useful for both breeding and physiological experiments.

Phenotyping with image segmentation can be made easier by using more controlled environments, mitigating the challenges associated with images from more natural and heterogeneous soil. A limitation of phenotyping roots in controlled experiments is that their observed traits show inconsistent correlation with more mature root traits in the field [167], motivating the need for the segmentation of root images grown in field conditions.

Image analysis of roots is difficult due to soil debris, background clutter and scratches on the glass through which the roots are observed. Thus, there was a need for the development of more robust and easy to use root segmentation methods.

Neural networks can segment roots accurately [194] but are difficult to train for non-experts without a computer-science background. Achieving acceptable accuracy on a new plant image dataset often requires training a new segmentation model. Thus, to make accurate root segmentation more accessible and convenient there is a need for a user-friendly way to rapidly train deep learning models for biological image segmentation.

1.4 Interactive Machine Learning

Conventional supervised model training, described by [83] as automatic machine learning (aML) typically involves discrete stages. Firstly, labels are created for the purpose of training a model and then, after the labels have been prepared, they are split up into training, validation and test datasets, along with their corresponding images. The creation of labels for segmentation model training is referred to as annotation and in radiotherapy may involve contouring.

In many cases, when tackling new real world problems using machine learning, a large pre-existing data set with carefully curated annotations is not readily available. It is difficult to know how much data must be annotated to provide a model with an acceptable accuracy. There is a need for methods that provide feedback to the annotator to allow them to make more informed decisions about how much and what specific data must be annotated.

As opposed to aML, interactive machine learning (IML) is a kind of machine learning where the labels are added while the model is being trained. With IML there is a human-in-the-loop that has a real-time influence on the model training procedure. Coupling annotation and training can provide benefits in comparison to aML because the annotator is able to inspect and view the model performance during the annotation procedure, informing their decision on when to stop annotating, what needs to be annotated, and allowing them to better

understand what it is they're actually labelling by having the model inform and guide their predictions. In comparison to active learning, where a model decides which examples should be labelled, IML involves the human user selecting which examples are labelled from a larger pool or stream of potential training data. IML is different to interactive segmentation because in interactive segmentation user input is used during inference to influence the predictions, whereas IML involves using user input to influence the training procedure via annotations added in response to the model's current behaviour. An IML process is a user-driven error minimisation loop arranged around the model being trained [48].

IML systems can be characterised as being either for task completion or model building [160]. Model building is where the focus of the IML process is to train a model that can be subsequently used for fully automated analysis of a typically much larger dataset. In contrast, task completion involves the human and model working together to complete a specific task, for example to contour a heart. In chapter 2 we present a study with an evaluation focused on model building whilst in chapter 3 we focus our evaluation more on task completion, with the time to complete each individual image contour reported, taking into consideration the model prediction and time to correct the model.

1.5 Evaluating IML for Image Segmentation using Real World Tasks

A limitation of many existing IML experiments is that they use simulated interaction to compare approaches. In this thesis (Chapter 2, 3, & 5), real human annotators are used to evaluate the effectiveness of the proposed approaches. Although simulations offer more control, they miss crucial aspects as there is a gap between simulated and real human behaviour. A common measure of the effectiveness of an IML strategy is how much fewer clicks are used, for interfaces where all interaction is expressed in clicks. The methods proposed in chapter 2 include a more free-form painting interface, that allows richer expression when interacting with the model and providing annotations as feedback.

A focus of previous studies is reducing the number of labels that can be assigned without compromising trained model accuracy. Reducing labels is a relevant goal, but it is only a proxy for the true cost of annotation. We argue that an annotator's time to achieve a given task is a more relevant metric, which we directly measure in the studies presented in this thesis.

To evaluate the models trained with IML, we use an approach known as Test-Then-Train [58]. Test-Then-Train allows each annotated image to be used both as an unseen example to evaluate model generalisation performance and to be included in the training data.

Although [58] state a Test-Then-Train strategy removes the need for a held-out set, we still compare accuracy to a separate test set that is kept out of the interactive training process for each study.

In this thesis, we implement and investigate multiple novel methods mani-

festing IML processes utilising deep learning segmentation architectures. Our evaluation uses real world tasks from both plant science and radiotherapy, enabling immediate translation into these disparate biological research fields.

RootPainter: Deep Learning Segmentation of Biological Images with Corrective Annotation

Abraham George Smith[1, 2], Eusun Han[1, 5], Jens Petersen[2], Niels Alvin Faircloth Olsen[1], Christian Giese[3], Miriam Athmann[4], Dorte Bodin Dresbøll[1], Kristian Thorup-Kristensen[1]

1 – Department of Plant and Environmental Science, University of Copenhagen, Højbakkegårds Alle 13, Tåstrup 2630, Denmark
2 – Department of Computer Science, University of Copenhagen, Universitetsparken 1, 2100 Copenhagen, Denmark
3 – Department of Agroecology and Organic Farming, University of Bonn, Regina-Pacis-Weg 3, 53113 Bonn, Germany
4 – Department of Organic Farming and Plant Production, University of Kassel, Nordbahnhofstr. 1a, D-37213 Witzenhausen, Germany
5 – CSIRO Agriculture and Food, PO Box 1700, Canberra, ACT 2601, Australia

Published in the New Phytologist on the 18th of July 2022 [195]

2.1 Summary

Convolutional neural networks (CNNs) are a powerful tool for plant image analysis, but challenges remain in making them more accessible to researchers without a machine learning background. We present RootPainter, an open-source graphical user interface (GUI) based software tool for the rapid training of deep neural networks for use in biological image analysis.

We evaluate RootPainter by training models for root length extraction from

chicory *(Cichorium intybus L.)* roots in soil, biopore counting and root nodule counting. We also compare dense annotations to corrective ones which are added during the training process based on the weaknesses of the current model.

Five out of six times the models trained using RootPainter with corrective annotations created within two hours produced measurements strongly correlating with manual measurements. Model accuracy had a significant correlation with annotation duration, indicating further improvements could be obtained with extended annotation.

Our results show that a deep learning model can be trained to a high accuracy for the three respective datasets of varying target objects, background and image quality with less than two hours of annotation time. They indicate that when using RootPainter, for many datasets it is possible to annotate, train and complete data processing within one day.

2.2 Introduction

Plant research is important because we need to find ways to feed a growing population whilst limiting damage to the environment [126]. Plant studies often involve the measurement of traits from images, which may be used in phenotyping for genome-wide association studies [165], comparing cultivars for traditional breeding [237] or testing a hypothesis related to plant physiology [161]. Plant image analysis has been identified as a bottleneck in plant research [136]. A variety of software exists to quantify plant images [123] but is typically limited to a specific type of data or task such as leaf counting [222], pollen counting [211] or root architecture extraction [252].

Convolutional neural networks (CNNs) are a class of deep learning models that represent the state-of-the-art for image analysis and are currently among the most popular methods in computer vision research. They are a type of multi-layered neural network that uses convolution in at least one layer and are designed to process grid like data such as images [114]. CNNs such as U-Net [170] receive an image as input and then output another image, with each pixel in the output image representing a prediction for each pixel in the input image. CNNs excel at tasks such as segmentation and classification. They have been found to be effective for various tasks in agricultural research [101, 174] and plant image analysis, including plant stress phenotyping [99], wheat spike counting [156], leaf counting [223] and accession classification [207].

For a CNN model to perform a particular task, it must be trained on a suitable dataset of examples. These examples are referred to as training data and are typically a collection of input images paired with the desired output for that image. In the case of segmentation, each input image is paired with a set of labels corresponding to each of the pixels in the input image. The process of creating such labelled training data is referred to as annotation and

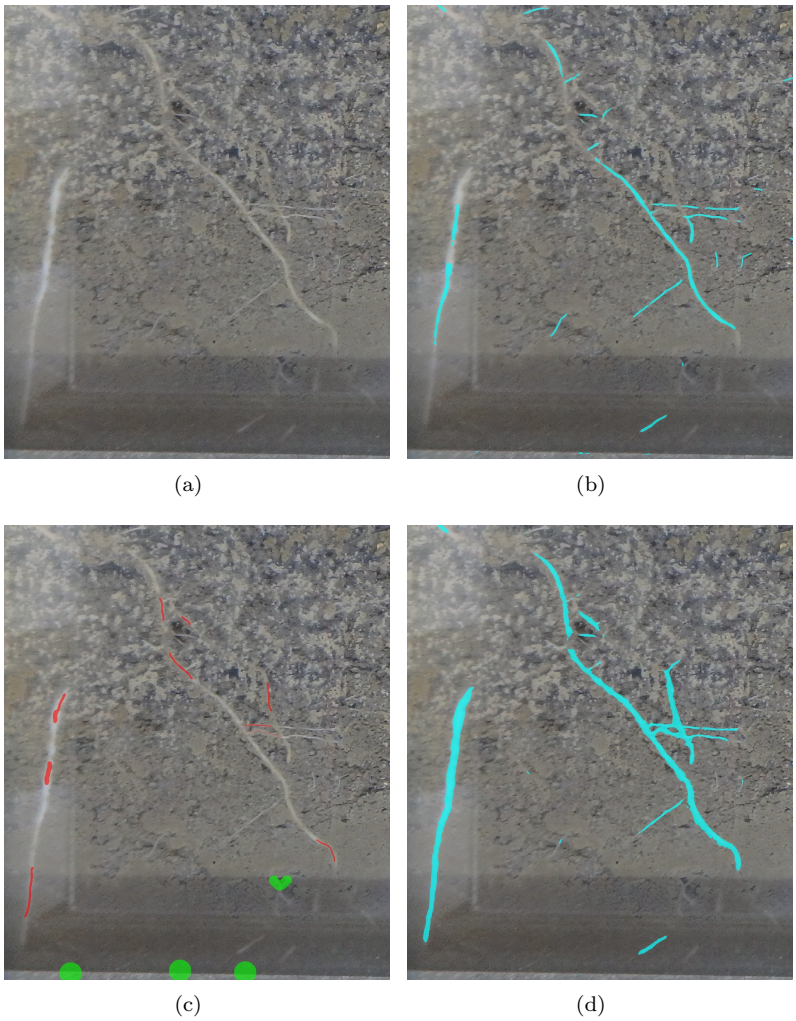


Figure 2.1: RootPainter corrective annotation concept. (a) Roots in soil. (b) AI root predictions (segmentation) shown in bright blue overlaid over photograph. (c) Human corrections of the initial segmentation, with corrections of false negatives shown in red and corrections of false positives shown in green. (d) After a period of training the AI learns from the corrections provided. The updated segmentation is shown in bright blue.

can be time consuming as annotation of complex images can be labour intensive and many images may be desired as larger training datasets typically result in improvements in trained model performance [144].

CNN model training involves a process called stochastic gradient descent (SGD) optimising the parameters of a model such that the error is reduced. The error, commonly referred to as the loss, is a measure of the difference between the model's predictions and the correct labels for the training data examples. A separate validation dataset, consisting of similar examples, is used to provide information on the performance of the model during the training procedure on examples that have not been used as part of SGD optimisation. Validation set performance may be used to decide when to stop training or assist in tuning hyper-parameters, which are variables controlling fundamentals of the model that are not directly optimised by SGD.

Developing a CNN-based system for a new image analysis task or dataset is challenging because dataset design, model training and hyper-parameter tuning are time-consuming tasks requiring competencies in both programming and machine learning.

Three questions that need answering when attempting a supervised learning project such as training a CNN are: how to split the data between training, validation and test datasets; how to manually annotate or label the data; and how to decide how much data needs to be collected, labelled and used for training in order to obtain a model with acceptable performance. The choice of optimal hyper-parameters and network architecture are also considered to be a 'black art' requiring years of experience and a need has been recognised to make the application of deep learning easier in practice [200].

The question of how much data to use in training and validation is explored in theoretical work that gives indications of a model's generalisation performance based on dataset size and number of parameters [229]. These theoretical insights may be useful for simpler models but provide an inadequate account of the behaviour of CNNs in practice [258].

Manual annotation may be challenging as proprietary tools may be used which are not freely available [245] and can increase the skill set required. Creating dense per-pixel annotations for training is often a time consuming process. It has been argued that tens of thousands of images are required, making small scale plant image datasets unsuitable for training deep learning models [222].

The task of collecting datasets for the effective training of models is further confounded by the unique attributes of each dataset. All data are not created equal, with great variability in the utility of each annotated pixel for the model training process [104]. It may be necessary to add harder examples after observing weaknesses in an initial trained model [202], or to correct for a class imbalance in the data where many examples exist of a majority class [27].

Interactive segmentation methods using CNNs such as [86, 173] provide ways to improve the annotation procedure by allowing user input to be used in the inference process and can be an effective way to create large high quality datasets in less time [21].

When used in a semi-automatic setting, such tools will speed up the labelling

process but may still be unsuitable for situations where the speed and consistency of a fully automated solution is required. For example when processing data from large scale root phenotyping facilities such as [205] where in the order of 100,000 images or more need to be analysed.

In this study we present and evaluate our software RootPainter which makes the process of creating a dataset, training a neural network and using it for plant image analysis accessible to ordinary computer users by facilitating all required operations with a cross-platform, open-source and freely available user-interface. The RootPainter software was initially developed for quantification of roots in images from rhizotron based root studies. However, we found its versatility to be much broader, with an ability to be trained to recognise many different types of structures in a set of images.

Although more root specific [194, 57, 146] and more generalist segmentation tools such as Fiji [176] via DeepImageJ [61] make it possible to run trained deep learning models for segmentation, they do not provide easy to use model training functionality, which is the purpose of the presented RootPainter software.

RootPainter allows a user to inspect model performance during the annotation process so they can make a more informed decision about how much and what data is necessary to label in order to train a model to an acceptable accuracy. It allows annotations to be targeted towards areas where the current model shows weakness (Fig. 2.1) in order to streamline the process of creating a dataset necessary to achieve a desired level of performance. RootPainter can operate in a semi-automatic way, with a user assigning corrections to each segmented image, whilst the model learns from the assigned corrections, reducing the time-requirements for each image as the process is continued. It can also operate in a fully-automatic way by either using the model generated from the interactive procedure to process a larger dataset without required interaction, or in a more classical way by using a model trained from dense per-pixel annotations which can also be created via the user interface.

We evaluate the effectiveness of RootPainter by training models for three different types of data and tasks without dataset-specific programming or hyperparameter tuning. We evaluate the effectiveness on a set of rhizotron root images, and in order to evaluate the versatility of the system, also on two other types of data, a biopores dataset, and a legume root nodules dataset, both involving objects in the images quite different from roots.

For each dataset we compare the performance of models trained using the dense and corrective annotation strategies on images not used during the training procedure. If annotation is too time-consuming, then RootPainter will be unfeasible for many projects. To investigate the possibility of rapid and convenient model training we use no prior knowledge and restrict annotation time to a maximum of two hours for each model. We hypothesize that (1) in a limited time period RootPainter will be able to segment the objects of interest to an acceptable accuracy in three datasets including roots, biopores and root nodules, demonstrated by a strong correlation between the measurements obtained from RootPainter and manual methods. And (2) a corrective annotation strategy will result in a more accurate model compared to dense annotations, given the

same time for annotation.

Training with corrective-annotation is a type of interactive-machine-learning, as it uses a human-in-the-loop in the model training procedure. As opposed to active-learning, which involves the learner automatically selecting which examples the user labels [184], interactive-machine-learning involves a human deciding which examples should be added for future iterations of training [8].

Prior work for interactive training for segmentation includes [62] and [110]. [62] evaluated their method using neuronal structures captured using Electron Microscopy, and found the interactively trained model to produce better segmentations than a model trained using exhaustive ground truth labels.

[110] combined interactive segmentation with interactive training by using the user feedback in model updates. Their training approach requires an initial dataset with full ground-truth segmentations, whereas our method requires no prior labelled data, which was a design choice we made to increase the applicability of our method to plant researchers looking to quantify new objects in a captured image dataset.

As opposed to [62] we use a more modern, fully convolutional network model, which we expect to provide substantial efficiency benefits when dealing with larger images. Our work is novel in that we evaluate an interactive corrective annotation procedure in terms of annotation time to reach a certain accuracy on real-world plant image datasets. Synthetic data is often used to evaluate interactive segmentation methods [20, 119, 129]. To provide more realistic measurements of annotation time we use real human annotators for our experiments. As opposed to many competing deep learning methods for segmentation, we provide a graphical user interface (GUI) which allows all operations to be completed using a user-interface, an essential feature for ensuring uptake in the plant image analysis community.

2.2.1 Roots in Soil

Plant roots are responsible for uptake of water and nutrients. This makes understanding root system development critical for the development of resource efficient crop production systems. For this purpose, we need to study roots under real life conditions in the field, studying the effects of crop genotypes and their management [163, 162], cover crops [215], crop rotation [218] and other factors. We need to study deep rooting, as this is critical for the use of agriculturally important resources such as water and nitrogen [217, 219].

Rhizotron based root research is an important example of plant research. Acquisition of root images from rhizotrons is widely adopted [166], as it allows repeated and non-destructive quantification of root growth and often to the full depth of the root systems. Traditionally the method for root quantification in such studies involves a lengthy procedure to determine the root density on acquired images by counting intersections with grid-lines [216].

Manual methods require substantial resources and can introduce undesired inter-annotator variation on root density, therefore a faster and more consistent method is required. More recently, fully automatic approaches using CNNs

have been proposed [194]; although effective, such methods may be challenging to re-purpose to different datasets for root scientists without the required programming expertise. A method which made the re-training process more accessible and convenient would accelerate the adoption of CNNs within the root research community.

2.2.2 Biopores

Biopores are tubular or round-shaped continuous voids formed by root penetration and earthworm movement [103]. They function as preferential pathways for root growth [70] and are therefore important for plant resource acquisition [112, 71]. Investigation of soil biopores is often done by manually drawing on transparent sheets on an excavated soil surface [69]. This manual approach is time consuming and precludes a more in-depth analysis of detailed information, including diameter, surface area, or distribution patterns such as clustering.

2.2.3 Root Nodules

Growing legumes with nitrogen-fixing capacity reduces the use of fertilizer [105], hence there is an increased demand for legume-involved intercropping [75] and precropping for carry over effects. Roots of legumes form associations with rhizobia, forming nodules on the roots, where the nitrogen fixation occur. Understanding the nodulation process is important to understand this symbiosis and the nitrogen fixation. However, counting nodules from the excavated roots is a cumbersome and time consuming procedure, especially for species with many small nodules such as clovers (*Trifolium spp.*).

2.3 Materials and Methods

2.3.1 Software Implementation

RootPainter uses a client-server architecture, allowing users with a typical laptop or desktop computer to utilise a graphics processing unit (GPU) on a more computationally powerful server. The client and server can be used on the same machine if it is equipped with suitable hardware, reducing network IO overhead. Instructions are sent from the client to server using human-readable JSON (JavaScript Object Notation) format. The client-server communication is facilitated entirely with files via a network drive or file synchronisation application. This allows utilisation of existing authentication, authorisation and backup mechanisms whilst removing the need to setup a publicly accessible static IP address. The graphical client is implemented using PyQt5 which binds to the Qt cross-platform widget toolkit. The client installers for Mac, Windows, and Linux are built using the PyInstaller build system which bundles all required dependencies. As opposed to the more generalist annotation software napari

[201] which is also built using Qt, the RootPainter client is designed to specifically facilitate our proposed corrective annotation protocol and to not require python familiarity. Image data can be provided as JPEG, PNG or TIF and in either colour or grayscale. Image annotations and segmentations are stored as PNG files. Models produced during the training process are stored in the python pickle format and extracted measurements in comma-separated value (CSV) text files.

A folder referred to as the *sync directory* is used to store all datasets, projects and instructions which are shared between the server and client. The server setup (Supporting Information, Note S2.9.1) requires familiarity with the Linux command line so should be completed by a system administrator. The server setup involves specification of a sync directory, which must then be shared with users. Users will be prompted to input the sync directory relative to their own file system when they open the client application for the first time and it will be automatically stored in their home folder in a file named *root_painter_settings.json* which the user may delete or modify if required.

Creating a Dataset

The *Create training dataset* functionality is available as an option when opening the RootPainter client application. It is possible to specify a source image directory, which may be anywhere on the local file system and whether all images from the source directory should be used or a random sample of a specified number of images. It is also possible to specify the target width and height of one or more samples to take from each randomly selected image; this can provide two advantages in terms of training performance. Firstly, RootPainter loads images from disk many times during training which can for larger images (more than 2000×2000 pixels) slow down training in proportion to image size and hardware capabilities. Secondly, recent results [121] indicate that capturing pixels from many images is more useful than capturing more pixels from each image when training models for semantic segmentation, thus when working with datasets containing many large images, using only a part of each image will likely improve performance given a restricted time for annotation.

When generating a dataset, each image to be processed is evaluated for whether it should be split into smaller pieces. If an image's dimensions are close to the target width and height then the image will be added to the dataset without it being split. If an image is substantially bigger then all possible ways to split the image into equally sized pieces above the minimum are evaluated. For each of the possible splits, the resultant piece dimensions are evaluated in terms of their ratio distance from a square and distance from the target width and height. The split which results in the smallest sum of these two distances is then applied. From the split image, up to the *maximum tiles per image* are selected at random and saved to the training dataset. The source images do not need to be the same size and the images in the generated dataset will not necessarily be the same size but all provided images must have a width and height of at least 572 pixels, and we recommend at least 600 as this will allow random crop data

augmentation. The dataset is created in the RootPainter sync directory in the datasets folder in a subdirectory which takes the user-specified dataset name. To segment images in the original dimensions, the dataset creation routine can be bypassed by simply copying or moving a directory of images into a subdirectory in the RootPainter datasets directory.

Working with Projects

Projects connect datasets with models, annotations, segmentations and messages returned from the server. They are defined by a project file (.seg_proj) which specifies the details in JSON and a project folder containing relevant data. The options to create a project or open an existing project are presented when opening the RootPainter client application. Creating projects requires specifying a dataset and optionally an initial model file. Alternatively a user may select 'random weights' also known as training from scratch, which will use He initialization [77] to assign a models initial weights. A project can be used for inspecting the performance of a model on a given dataset in the client, or training a model with new annotations which can also be created using drawing tools in the client user interface.

Model architecture

We modified the network architecture from [194] which is a variant of U-Net [170] implemented in PyTorch [152] using Group Normalization [244] layers. U-Net is composed of a series of down-blocks and up-blocks joined by skip connections. The entire network learns a function which converts the input data into a desired output representation, e.g. from an image of soil to a segmentation or binary map indicating which of the pixels in the image are part of a biopore. In the down-blocks we added 1×1 convolution to halve the size of the feature maps. We modified both down-blocks and up-blocks to learn residual mappings, which have been found to ease optimization and improve accuracy in CNNs [76] including U-Net [259]. To speed up inference by increasing the size of the output segmentation, we added one pixel padding to the convolutions in the down-blocks and modified the input dimensions from $512 \times 512 \times 3$ to $572 \times 572 \times 3$, which resulted in a new respective output size of $500 \times 500 \times 2$, containing a channel for the foreground and background predictions. The modified architecture has approximately 1.3 million trainable parameters, whereas the original had 31 million. These alterations reduced the saved model size from 124.2 MB [191] to 5.3 MB, making it small enough to be conveniently shared via email.

Creating Annotations

Annotations can be added by drawing in the user interface with either the foreground or background brush tools. It's also possible to undo or redo brush strokes. Annotation can be removed with the eraser tool. If an image is only

partially annotated then only the regions with annotation assigned will be used in the training. Whilst annotating it's possible to hide and show the annotation, image or segmentation. For convenience during use, a table of keyboard shortcuts is presented in Supporting Information, Note S2.9.2. When the user clicks *Save & next* in the interface, the current annotation will be saved and synced with the server, ready for use in training. The first and second annotations are added to the training and validation sets respectively (see *Training Procedure* below). Afterwards, to maintain a typical ratio between training and validation set, annotations will be added to the validation set when the training set is at least five times the size of the validation set, otherwise they will be added to the training set.

Training Procedure

The training procedure can be started by selecting *Start training* from the network menu which will send a JSON instruction to the server to start training for the current project. The training will only start if the project has at least two saved annotations as at least one is required for each of the training and validation set. Based on [194] we use a learning rate of 0.01 and Nestorov momentum with a value of 0.99. We removed weight decay as results have shown similar performance can be achieved with augmentation alone whilst reducing the coupling between hyperparameters and dataset [81]. The removal of weight decay has also been suggested in practical advice [22] based on earlier results [37] indicating its superfluity when early stopping is used. We do not use a learning rate schedule in order to facilitate an indefinitely expanding dataset.

An *epoch* typically refers to a training iteration over the entire dataset [63]. In this context we initially define an epoch to be a training iteration over 612 image sub-regions corresponding to the network input size, which are sampled randomly from the training set images with replacement. We found an iteration over this initial epoch size to take approximately 30 seconds using two RTX 2080 Ti GPUs with an automatically selected batch size of 6. If the training dataset expands beyond 306 images, then the number of sampled sub-regions per epoch is set to twice the number of training images, to avoid validation overwhelming training time. The batch size is automatically selected based on total GPU memory and all GPUs will be used by default using data parallelism.

RootPainter utilises a supervised learning procedure, which involves training a model to predict the same value (either foreground or background) for a pixel in the input data as the one found in the corresponding location of an annotation. The annotations are created by the user and include pixels annotated as foreground, pixels annotated as background, and pixels without annotation. Only the pixels annotated as foreground or background are used in training. This is achieved by setting the pixels without foreground or background annotation to 0 in both the network prediction and associated annotation. The distance between the annotated pixels and network predictions is computed using a loss function, which is a combination of dice-loss and cross-entropy taken from [194]. This computed loss is then used to update the network weights,

leading to a model with reduced error on subsequent images.

After each epoch, the model predictions are computed on the validation set and F_1 is calculated for the current and previously saved model. If the current model's F_1 is higher than the previously saved model then it is saved with its number and current time in the file name. If for 60 epochs no model improvements are observed and no annotations are saved or updated then training will stop automatically.

We designed the training procedure to have minimal RAM requirements which do not increase with dataset size, in order to facilitate training on larger datasets. We found the server application to use less than 8GB of RAM during training and inference, and would suggest at least 16GB RAM for the machine running the server application. We found the client to use less than 1GB RAM but have not yet tested on devices equipped with less than 8GB of RAM.

Augmentation

We modified the augmentation procedure from [194] in three ways. We changed the order of the transforms from fixed to random in order to increase variation. We reduced the probability that each transform is applied to 80% in order to reduce the gap between clean and augmented data, which recent results indicate can decrease generalization performance [78]. We also modified the elastic grid augmentation as we found the creation of the deformation maps to be a performance bottleneck. To eliminate this bottleneck we created the deformation maps at an eighth of the image size and then interpolated them up to the correct size.

Creating Segmentations

It is possible to view segmentations for each individual image in a dataset by creating an associated project and specifying a suitable model. The segmentations are generated automatically via an instruction sent to the server when viewing each image and saved in the segmentations folder in the corresponding project.

When the server generates a segmentation, it first segments the original image and then a horizontally flipped version. The output segmentation is computed by taking the average of both and then thresholding at 0.5. This technique is a type of test time data augmentation which is known to improve performance [153]. The segmentation procedure involves first splitting the images into tiles with a width and height of 572 pixels, which are each passed through the network and then an output corresponding to the original image is reconstructed.

It's possible to segment a larger folder of images using the *Segment folder* option available in the network menu. To do this, an input directory, output directory and one or more models must be specified. The model with the highest number for any given project will have the highest accuracy in terms of F_1 on the automatically selected validation set. Selecting more than one model will

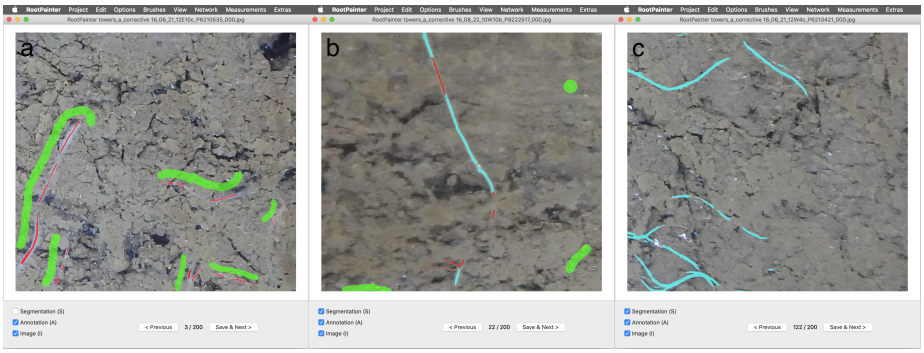


Figure 2.2: Screenshots of the RootPainter software, showing examples of the various stages of the interactive training procedure. (a) Initial annotation created at the start of interactive training, with foreground annotation shown in red and background annotation shown in green. (b) Corrective annotation whilst the model progresses to a suitable solution. The segmentation is shown in light blue. (c) Segmentation shown later in the model training process.

result in model averaging, an ensemble method which improves accuracy as different models don't usually make identical errors [63]. from different projects representing different training runs on the same dataset will likely lead to a more diverse and thus more accurate ensemble, given they are of similar accuracy. It it is also possible to use models saved at various points from a single training run, a method which can provide accuracy improvements without extending training time [183].

The user interface is shown in Fig. 2.2 with various stages of the interactive training procedure shown.

Extracting Measurements

It is possible to extract measurements from the produced segmentations by selecting an option from the measurements menu. The *Extract length* option extracts centerlines using the skeletonize method from scikit-image [227] and then counts the centerline pixels for each image. The *Extract region properties* uses the scikit-image regionprops method to extract the coordinates, diameter, area, perimeter and eccentricity for each detected region and stores this along with the associated filename. The *Extract count* method gives the count of all regions per image. Each of the options require the specification of an input segmentation folder and an output CSV.

2.3.2 Datasets

Biopore Images

Biopore images were collected near Meckenheim (50°37′9″N 6°59′29″E) at a field trial of University of Bonn in 2012 (see [69] for a detailed description). Within each plot an area was excavated to a depth of 0.45 m. The exposed soil surface was carefully flattened to reveal biopores and then photographed in colour.

Bersoft software (Windows, Version 7.25) was used for biopore quantification. Using the eclipse function, the visible biopores were marked, then the count number was generated as a CSV file. Pores smaller than two mm were excluded from biopore counting.

We restricted the analysis to images with a suitable resolution and cropped to omit border areas. For each image, the number of pixels per mm was recorded using Gimp (MacOS, Version 2.10) in order to calculate pore diameter. We split the images into two folders. BP_counted which contained 39 images and was used for model validation after training as these images had been counted by a biopore expert and BP_uncounted which contained 54 images and was used for training.

Nodule Images

Root images of persian clover (*Trifolium resupinatum*) were acquired at 800 DPI in colour using a water-bed scanner after root extraction. We used a total of 113 images which all had a blue background, but were taken with two different scanners. From the 113 images, 65 were captured using an Epson V700 scanner and appeared darker and underexposed whereas 48 were captured using an Epson Expression 12000XL Photo Scanner and appeared well lit, showing the nodules more clearly. The blue background was obtained by blocking the scanner transparency unit with non-transparent blue paper to create a background which provided contrast to the root nodules. Blocking the scanner transparency

Table 2.1: Details for each of the datasets created for training. The number of images and tiles were chosen to enable a consistent dataset size of 200 images. Only 50 images were sampled from for the biopores and nodules, in order to ensure there were enough images left in the test set. The datasets created are available to download from [199].

Object	Name	Source folder	Reference	To sample	Max tiles	Target size
Biopores	BP_750_training	BP_uncounted	[198]	50	4	750
Nodules	nodules_750_training	counted_nodules	[192]	50	4	750
Roots	towers_750_training	grid_counted_roots	[190]	200	1	750

unit meant that the only light source was from the document table (the lower part) of the scanner.

They were counted manually using WinRhizo Pro (Regent Instruments Inc., Canada, Version 2016). Image sections were enlarged and nodules were selected manually by clicking. Then, the total number of marked nodules were counted by the software. We manually cropped to remove the borders of the scanner using Preview (MacOS, Version 10.0) and converted to JPEG to ease storage and sharing. Of these 50 were selected at random to have subregions included in training and the remaining 63 were used for validation.

Roots Dataset

We downloaded the 867 grid counted images and manual root length measurements from [190] which were made available as part of the evaluation of U-Net for segmenting roots in soil [194] and originally captured as part of a study on chicory drought stress [161] using a four m rhizobox laboratory described in [220]. We removed the 10 test images from the grid counted images, leaving 857 images. The manual root length measurements are a root intensity measurement per-image, which was obtained by counting root intersections with a grid as part of [161].

2.3.3 Annotation and Training

For the roots, nodules and biopores we created training datasets using the *Create training dataset* option. We used random sample, with the details specified in Table 2.1. The two users (user a and user b) that we used to test the software were the two first authors. Each user trained two models for each dataset. For each model, the user had two hours (with a 30 minutes break between them) to annotate 200 images. We first trained a model using the corrective annotation strategy whilst recording the finish time and then repeated the process with the dense annotation strategy, using the recorded time from the corrective training as a time limit. This was done to ensure the same annotation time was used for both annotation strategies. With corrective annotations, the annotation and training processes are coupled as there is a feedback loop between the user and model being trained that happens in real time. Whereas with dense the user annotated continuously, without regard to model performance. The protocol followed when using corrective annotations is outlined in Supporting Information, Notes S2.9.3 and annotation advice given in Supporting Information, Notes S2.9.4. For the first six annotations on each dataset, we added clear examples rather than corrections. This was because we observed divergence in the training process when using corrective from the start in preliminary experiments. We suspect the divergence was caused by the user adding too many background classes compared to foreground or difficult examples. When creating dense annotations, we followed the procedure described in Supporting Information, Notes S2.9.5.

When annotating roots, in the interests of efficiency, a small amount of soil covering the root would still be considered as root if it was very clear that root was still beneath. Larger gaps were not labelled as root. Occluded parts of nodules were still labelled as foreground (Fig. 2.3(a)). Only the centre part of a nodule was annotated, leaving the edge as undefined. This was to avoid nodules which were close together being joined into a single nodule. When annotating nodules which were touching, a green line (background labels) was drawn along the boundary to teach the network to separate them so that the segmentation would give the correct counts (Fig. 2.3(b)).

After completing the annotation, we left the models to finish training using the early stopping procedure and then used the final model to segment the respective datasets and produce the appropriate measurements.

We also repeated this procedure for the projects but using a restricted number of annotations by limiting to those that had been created in just 30, 60, 90, 120 and 150 minutes (including the 30 minute break period) to give us an indication of model progression over time with the two different annotation strategies.

2.3.4 Measurement, Correlation and Segmentation Metrics

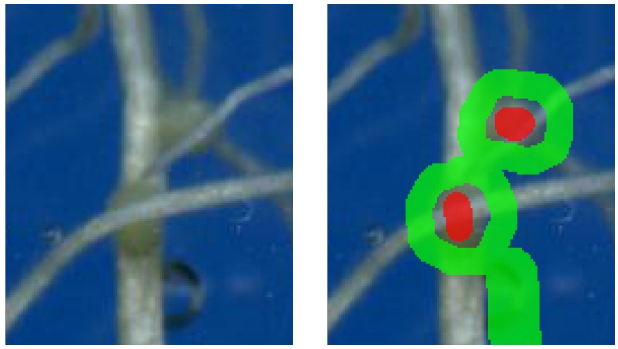
For each project we obtained correlations with manual measurements using the portion of the data not used during training to give a measure of generalization error, which is the expected value of the error on new input [63]. For the roots dataset, the manual measurements were compared to length estimates given by RootPainter, which are obtained from the segmentations using skeletonization and then pixel counting.

For the biopores and nodules datasets we used the extract region properties functionality from RootPainter, which gives information on each connected region in an output segmentation. For the biopores the regions less than 2mm in diameter were excluded. The number of connected regions for each image were then compared to the manual counts.

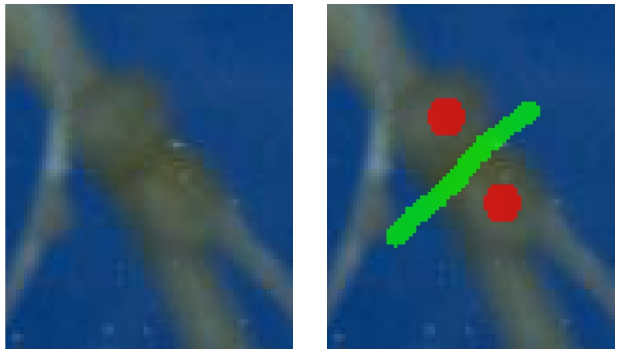
In order to obtain segmentation metrics, we used the extract segmentation metrics function available from the RootPainter extras menu. This function generates a CSV file containing dice score, recall, precision and accuracy for each of the segmented images in a project. The ground truth used for evaluation is the model prediction with the corrections assigned, i.e. the corrected segmentation. This corrected segmentation is then used to evaluate the predicted segmentation which is stored in the segmentations folder.

2.3.5 Filtering nodules by size

We investigated the effect of filtering out nodules less than a certain size by computing the correlation between the automated and manual nodule counts as a function of a size threshold. The size threshold meant that counted nodules would include only those above a specific area in pixels. We computed the



(a)



(b)

Figure 2.3: Nodule annotation. The red brush was used to mark the foreground (nodules) and the green brush to mark the background (not nodules). (a) We annotated nodules occluded by roots as though the roots were not there. (b) Adjacent nodules were separated using the background class.

correlation with each size threshold from 0 to 400 pixels. We did this for the model trained by user b only.

2.4 Results

We report the R^2 for each annotation strategy for each user and dataset (Table 2.2). Training with corrective annotations resulted in strong correlation ($R^2 \geq 0.7$) between the automated measurements and manual measurements five out of six times. The exception was the nodules dataset for user b with an R^2 of 0.69 (Table 2.2). Training with dense annotations resulted in strong correlation three out of six times, with the lowest R^2 being 0.55 also given by the nodules dataset for user b (Table 2.2).

Table 2.2: R^2 for each training run. These are computed by obtaining measurements from the segmentations from the final trained model and then correlating with manual measurements for the associated dataset.

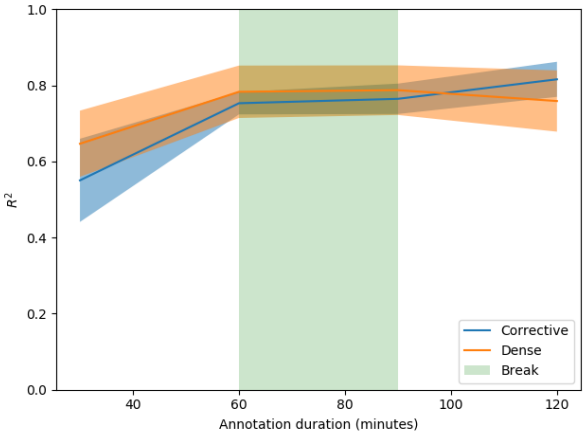
Dataset	User	Corrective R^2	Dense R^2
Biopores	a	0.78	0.58
Biopores	b	0.78	0.67
Nodules	a	0.73	0.89
Nodules	b	0.69	0.55
Roots	a	0.89	0.90
Roots	b	0.92	0.90

Table 2.3: Mean and standard error of the R^2 for each annotation strategy. These are computed by obtaining measurements from the segmentations from the final trained model and then correlating with manual measurements. Using mixed-effects model with annotation strategy as a fixed factor and user and dataset as random factors no significant effects were found ($P \leq 0.05$).

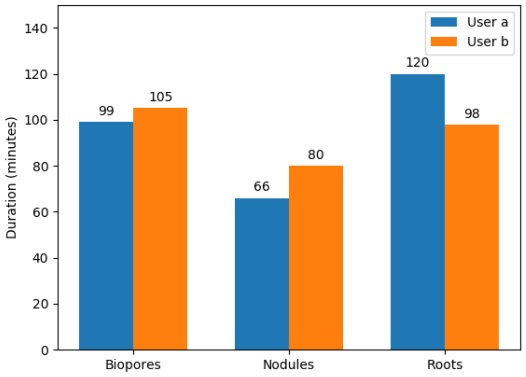
Strategy	Mean	Standard error
Corrective	0.80	0.04
Dense	0.75	0.07

For each annotation strategy, we report both the mean and standard error for the obtained R^2 values from all datasets and both users (Table 2.3). The mean of the R^2 values obtained when using corrective annotation shows they tended to be higher compared with dense, but the differences were not statistically significant (Mixed-effects model; $P \leq 0.05$). We plot the mean and standard error at each time point for which multiple R^2 values were obtained (Fig. 2.4(a)). In

general corrective improved over time, overtaking dense performance just after the break in annotation (Fig. 2.4(a)). The 30 minute break period taken by the annotator after one hour corresponds to a flat line in performance during that period (Fig. 2.4(a)). On average, dense annotations were more effective at the 30 minute time period, whereas corrective were more effective after two hours (including the 30 minute break) and at the end of the training (Table 2.3).



(a)



(b)

Figure 2.4: Annotation duration and accuracy. (a) Mean and standard error for the R^2 values over time. These include the 30 minute break and are restricted to time points where multiple observations are available. The shaded area indicates the standard error. (b) User reported duration in minutes for annotating each dataset, excluding the 30 minutes break taken after one hour of annotation. The annotator would use the same amount of time for both corrective and dense annotation strategies. The time fell below the limit of two hours (excluding break) when they ran out of images to annotate.

We report the duration for each user and dataset (Fig. 2.4(b)). Five out of six times all 200 images were annotated in less than the two hour time limit. The nodules dataset took the least time, with annotation completed in

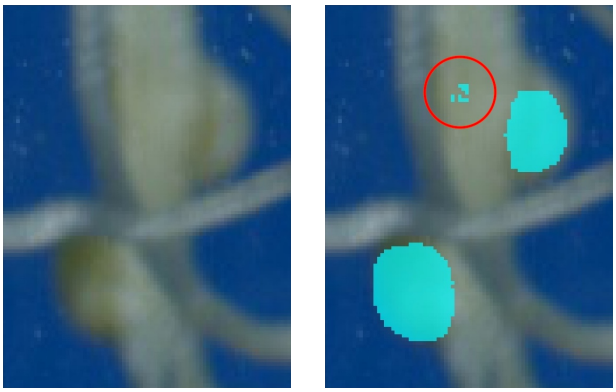
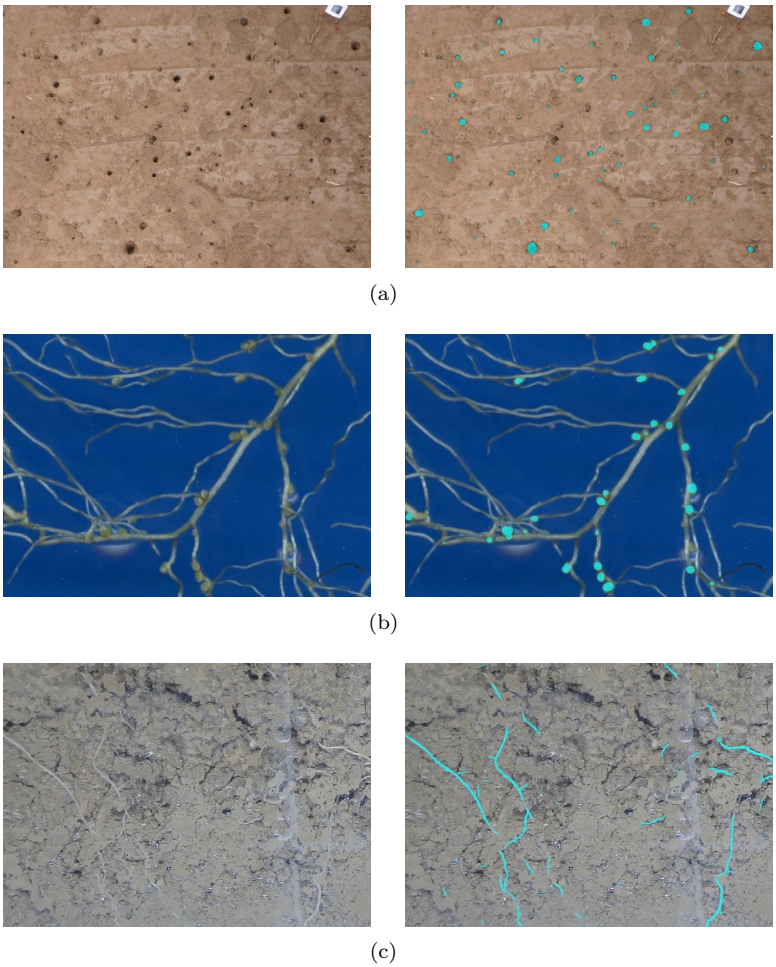


Figure 2.5: Two correctly detected nodules shown with three false positives. Segmentation is shown overlaid in light blue on top of a sub-region of one of the nodule images used for evaluation. The correct nodules are much larger and on the edge of the root. The three false positives are indicated by a red circle. They are much smaller and bunched together.

66 minutes and 80 minutes for users a and b respectively (Fig. 2.4(b)). The roots dataset for user a was the only project where the two hours time limit was reached without running out of images (Fig. 2.4(b)).



(a)

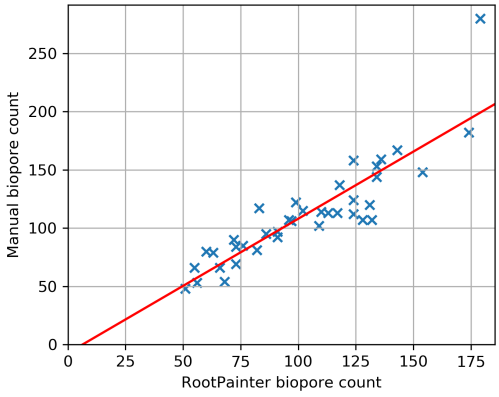
(b)

(c)

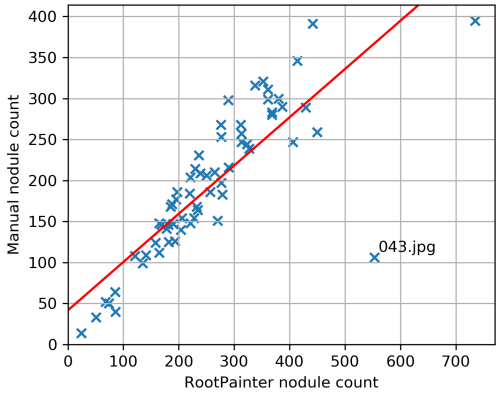
Figure 2.6: Example input and segmentation output from photographs not used in training. The segmentations are shown in light blue and are from models trained from scratch using no prior knowledge with annotations created using the corrective annotation protocol with RootPainter. (a) Biopores. Annotations created by user b in one hour and 45 minutes. (b) Nodules. Annotations created by user a in one hour and six minutes. (c) Roots. Annotations created by user a in two hours.

We show an example of errors found from the only model trained correctively which did not result in a strong correlation (Fig. 2.5). There were cases when the vast majority of pixels were labelled correctly but a few small incorrect pixels could lead to substantial errors in count (Fig. 2.5).

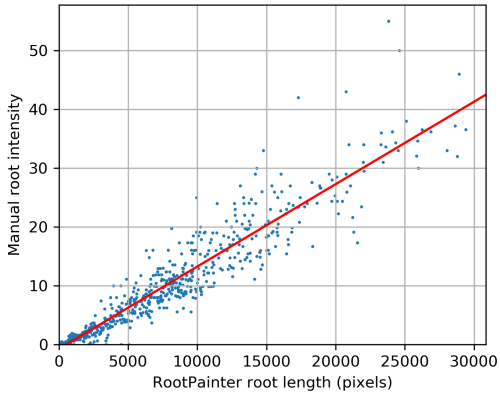
We found filtering nodules less than a certain size to provide substantial reductions in error. There was an improvement in R^2 from 0.69 to 0.75 when



(a)



(b)



(c)

Figure 2.7: Manual measurements plotted against automatic measurements attained using RootPainter. (a) Biopores using user b corrective model. (b) Nodules using user a corrective model. (c) Roots in soil using user a corrective model.

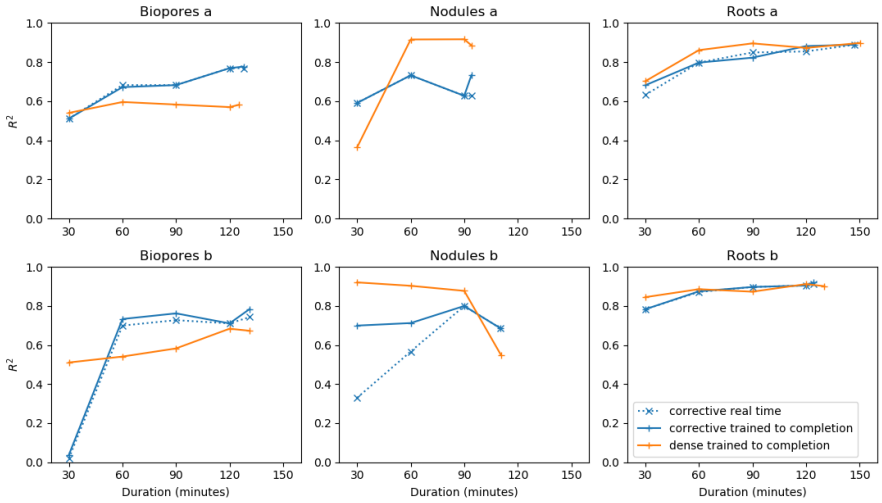
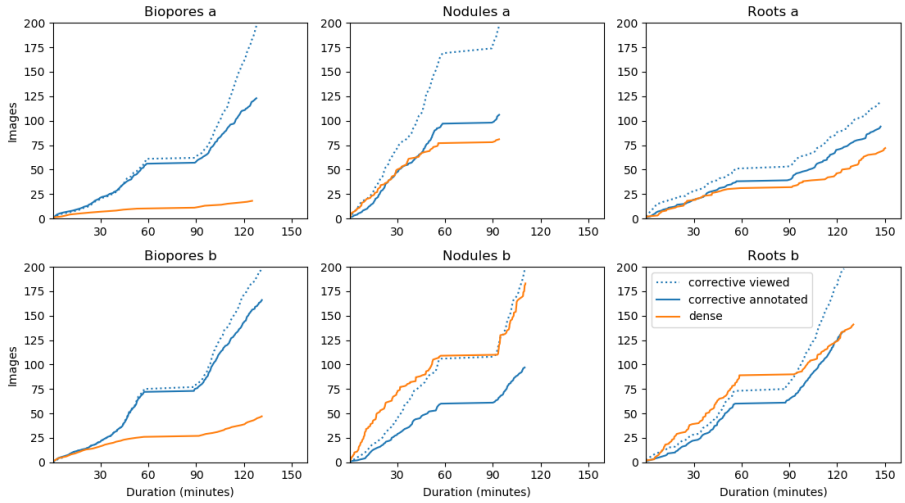
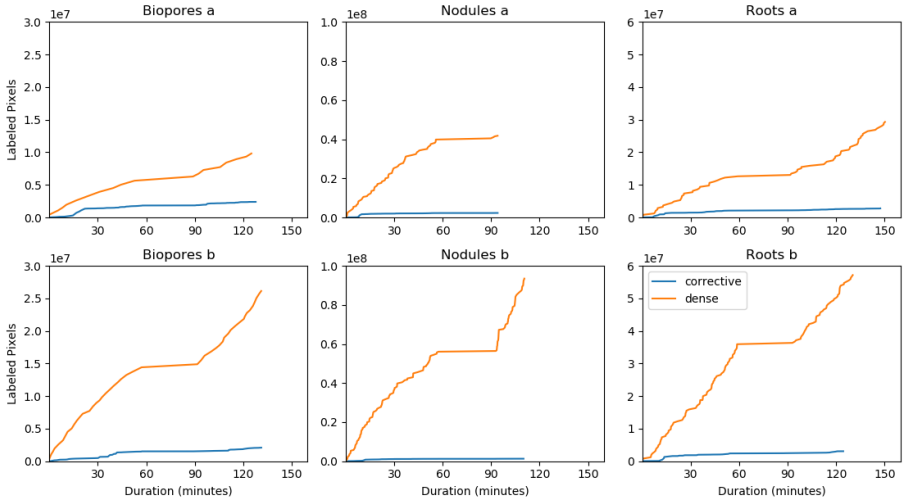


Figure 2.8: R^2 for the annotations attained after 30, 60, 90, 120 minutes and the final time point for users a and b on the three datasets for dense and corrective annotation strategies. *trained to completion* refers to models which were trained until stopping without interaction, using the annotations created within the specified time period, whereas *real time* refers to models saved during the corrective annotation procedure as it happened. For the corrective annotations we plot both the performance of the model saved during the training procedure and the same model if allowed to train to completion with the annotations available at that time.



(a)



(b)

Figure 2.9: Annotation progression. (a) Number of images viewed and annotated for the dense and corrective annotation strategies. For dense all images are both viewed and annotated, whereas corrective annotations are only added for images where the model predictions contain clear errors. (b) Total number of annotated pixels for dense and corrective annotation strategies over time during the annotation procedure. For dense, almost all pixels in each image are annotated. Corrective annotations are only applied to areas of the image where the model being trained exhibits errors.

changing the area threshold from 0 to 5 pixels. The benefits increased up to an area threshold of 284 pixels, giving an R^2 of 0.9 (Supporting Information, Notes S2.9.5).

We show examples of accurate segmentation results obtained with models trained using the corrective annotation strategy (Fig. 2.6(a), 2.6(b) and 2.6(c)) along with the corresponding manual measurements plotted against the automatic measurements obtained using RootPainter (Fig. 2.7).

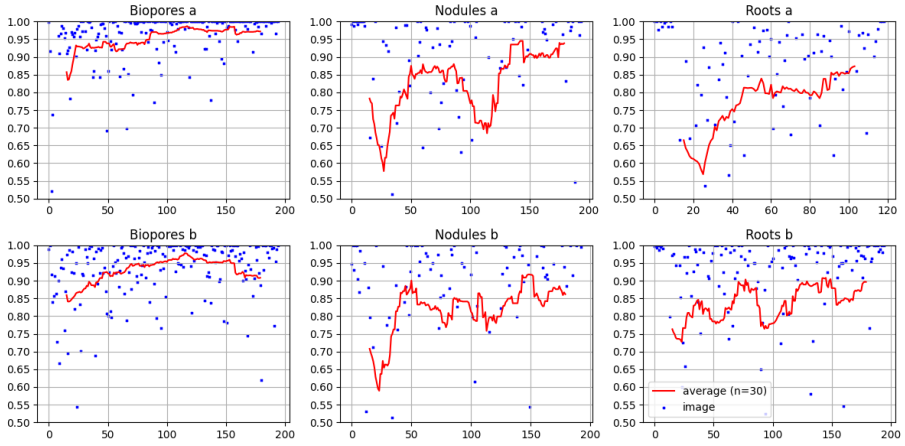
The observed R^2 values for corrective annotation had a significant positive correlation with annotation duration ($P<0.001$). There was no significant correlation between annotation time and R^2 values for models trained using dense annotations.

We plot the R^2 for each project after training was completed along with the R^2 obtained with training done only on annotations at restricted time limits, and refer to these as *trained to completion* along with the models saved at that time point during the corrective annotation procedure as it happened which we refer to as *real time* (Fig. 2.8). After only 60 minutes of annotation, all models trained for roots in soil gave a strong correlation with grid counts (Fig. 2.8, Roots a and b). The performance of dense annotation for user b on the nodules dataset was anomalous with a decrease in R^2 as more annotated data was used in training (Fig. 2.8, Nodules b). The corrective models obtained in real time were similar to those trained to completion, except nodules by user b, indicating that computing power was sufficient for real time corrective training (Fig. 2.8).

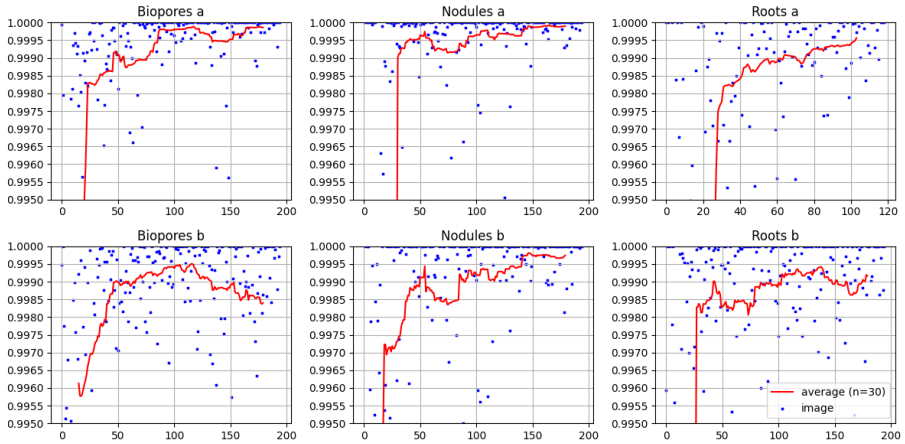
We plot the number of images viewed and annotated for the corrective and dense annotation strategies (Fig. 2.9(a)). For the corrective annotation strategy, only some of the viewed images required annotation. In all cases the annotator was able to progress through more images using corrective annotation (Fig. 2.9(a)).

For the roots and nodules datasets for user b for the first hour of training, progress through the images was faster when performing dense annotation (Fig. 2.9(a), Roots b and Nodules b).

We plot the amount of labelled pixels for each training procedure over time for both corrective and dense annotations (Fig. 2.9(b)). With corrective annotation less pixels were labelled in the same time period and as the annotator progressed through the images the rate of label addition decreased (Fig. 2.9(b)). The dice score is also plotted (Fig. 2.10(a)) and accuracy (Fig. 2.10(b)) along with running averages (n=30) for the models trained using corrective annotation. For both dice and accuracy, the running average shows both large fluctuations and a trend of continuous improvement as more images are annotated. The performance of the biopores model being trained by user b is an outlier, as it appears to decrease in accuracy and to a lesser extent dice score towards the end of training. For the nodules and roots datasets, towards the end of corrective-annotation, the dice score is approximately 0.9 but with larger fluctuations, whereas for the biopores the dice appears to stay consistently above 0.9.



(a)



(b)

Figure 2.10: Segmentation metrics including dice score (a) and accuracy (b) for each of the images segmented as part of the interactive segmentation procedure in order of annotation. Metrics were computed using the segmentations and corrective annotations created as part of the interactive training procedure for user a and user b for each of the three datasets. In this case the ground truth used for evaluation is the model prediction with the corrections assigned, i.e. the corrected segmentation. Dice ranges from 0-1 and is higher when the model prediction agrees with the ground truth. Accuracy ranges from 0-1 and is the ratio of pixels that were predicted correctly to the total pixels in the image. As accuracy is very high, to show the changes in the moving average, accuracy is only shown in the range of 0.99 to 1.0. There is a trend of continuous improvement as more images are annotated and interactive training continues.

2.5 Discussion

In this study we focused on annotation duration, as we consider the time requirements for annotation rather than the number of available images to be more relevant to the concerns of the majority of plant research groups looking to use deep learning for image analysis. Our results, for corrective training in particular, confirm our first hypothesis by showing that a deep learning model can be trained to a high accuracy for the three respective datasets of varying target objects, background and image quality in less than two hours of annotation time.

Our results demonstrate the feasibility of training an accurate model using annotations made in a short time period, which challenges the claims that tens of thousands of images [222] or substantial labelled data [145] are required to use CNNs. In practice, we also expect longer annotation periods to provide further improvement. The R^2 for corrective training had a significant correlation with annotation duration indicating that spending more time annotating would continue to improve performance.

There was a trend for an increasing fraction of viewed images to be accepted without further annotation later in the corrective training (Fig. 2.9(a)), indicating fewer of the images required corrections as the model performance improved. This aligns with the reduction in the rate of growth for the total amount of corrections (Fig. 2.9(b)) indicating continuous improvement in the model accuracy over time during the corrective training.

We suspect the cases where dense annotation had a comparatively faster speed in the beginning (Fig. 2.9(a), Roots b and Nodules b) were due to three factors. Firstly, switching through images has little overhead when using the dense annotation strategy as there is no delay caused by waiting for segmentations to be returned from the server. Secondly, corrective annotation will take a similar amount of time to dense in the beginning as the annotator needs to assign a large amount of corrections for each image. And thirdly, many of the nodule images did not contain nodules meaning dense annotations could be added almost instantly.

The average dice scores of approximately 0.9 for roots (Fig. 2.10(a)) is similar to previous results for root segmentation [194], indicating our trained model was accurate (please see below note on the limitations of such comparisons). For comparison, in previous work, similar root segmentation accuracy was obtained using approximately 25 hours of root annotation time [194]. We found that the dice scores for biopores were higher than those for roots or nodules (Fig. 2.10(a)), which we suspect was due to the biopore dataset being easier to segment, with higher contrast between the biopore and surrounding soil. Our reported dice scores plots (Fig. 2.10(a)) serve as a point of comparison for future work and in combination with RootPainter's functionality to extract segmentation metrics, provide a convenient way for users to confirm and report that their trained models have a suitable accuracy.

Although a trend in improvement over time is shown in the segmentation

metrics (Fig. 2.10(a), 2.10(b)), there were large fluctuations and for user b on the biopores dataset there appears to be a decrease in the model accuracy towards the end of corrective-annotation (Fig. 2.10(b)). We expect that the variation in accuracy was caused by both intra-annotator variation and by the large amount of variation in the quality and difficulty of the images for the network, as can be seen from all plots.

It is important to note that metrics, such as the above, comparing corrected predictions to predictions are not comparable to completely independent annotations done while blinded to the model predictions. Completely independent annotations include to some degree an amount of irreducible error stemming from observer variation and uncertainty. Corrections, on the other hand, are expected to be of clear errors and may therefore in some cases be a more useful measure of the system's performance as assessed by the observer in question.

Although corrective annotation tended to produce models with higher accuracy relative to dense (Table 2.3), the lack of a statistically significant difference prevents us from coming to a more substantive conclusion about the benefits of corrective over dense annotation. Despite being unable to confirm our second hypothesis, that corrective annotation provides improved accuracy over dense in a limited time period, it is still clear that it will provide many real-world advantages. The feedback given to the annotator will allow them to better understand the characteristics of the model trained with the annotations applied. They will be able to make a more informed decision about how many images to annotate to train a model to sufficient accuracy for their use case.

Although strong correlation was attained when using the models trained with corrective annotation, they in some cases overestimated (Fig. 2.7 a) or underestimated (Fig. 2.7 b) the objects of interest compared to the manual counts. For the biopores (Fig. 2.7 a) this may be related to the calibration and threshold procedure which results in biopores below a certain diameter being excluded from the dataset. We inspected the outlier in Fig. 2.7 b where RootPainter had overestimated the number of nodules compared to the manual counts. We found that this image (043.jpg) contained many roots which were bunched together more closely than what was typical in the dataset. We suspect this had confused the trained network and could be mitigated by using a consistent and reduced amount of roots per scan, whilst using more of the images for training and annotating for longer to capture more of the variation in the dataset.

In one case, training with corrective annotation failed to produce a model that gave a strong correlation with the manual measurements. This was for the nodules data for user b, where the R^2 was 0.69. We suspect this was partially due to the limited number of nodules in the training data. Many of the images in the dataset created for training contained no nodules and only included the background. This also meant the annotation was able to finish in less time. We consider this a limitation of the experimental design as we expect that a larger dataset which allowed for annotating nodules for the full two hour time period would have provided better insights into the performance of the corrective training procedure.

Fig. 2.5 shows examples of some of the errors in the nodules dataset. In practice, the annotator would be able to view and correct such errors during training until they had abated. We noticed that many of the nodule errors were smaller false positives, so investigated the effect of filtering out nodules less than a certain size (Supporting Information, Notes 2.9.5). The substantial improvements in nodule count correlation from 0.69 to 0.93 when using a nodule size threshold can be explained by the removal of smaller false positive artefacts. This indicates that the model was producing many small false positive predictions, which could also explain some of the overestimation of nodules (Fig. 2.7 b).

The problem with small false positives may have been mitigated with the dense annotations as a larger amount of background examples are added, suppressing more of the false positive predictions that arise in the limited training time.

The improvement in R^2 when removing small nodules may also be due to differences in subjective interpretation of what is a nodule, between the original counter and annotator training the model.

The reduction in R^2 as dense annotation time increased, shown in nodules b (Fig. 2.8) was highly unexpected. Although in some cases increasing training data can decrease performance when training CNNs [143], it is usually the case that the opposite is observed. We suspect these anomalous results are due to the large amount of variation in the success of the dense training procedure, rather than revealing any general relationship between performance and the amount of data used.

As the nodule images are captured in a controlled environment, further improvements to accuracy could be attained by reducing controllable sources of variation and increasing the technical quality of the images. The lighting was also varying for the nodules with approximately half of the images underexposed. We expect that more consistent lighting conditions would further improve the nodule counting accuracy. Cropping the nodule images manually could also become a time consuming bottleneck, which could be avoided by ensuring all the roots and nodules were positioned inside the border and having the placement of the border be fixed in its position in the scanner such that the cropping could be done by removing a fixed amount from each image, which would be trivial to automate.

Fig. 2.4(a) indicates corrective annotation leads to lower R^2 in the earlier phases of annotation (e.g. within 60 minutes). We suspect this is due to dense annotation having an advantage at the start as the user is able to annotate more pixels in less time using dense annotation with no overhead caused by waiting for segmentations from the server. We suspect in many cases corrective annotation will provide no benefits in terms of efficiency when the model is in the early stages of training as the user will still have to apply large amounts of annotation to each image, whilst slowed down by the delay in waiting for segmentations. Later in training, e.g. after one hour and 40 minutes, corrective overtakes dense in terms of mean R^2 performance (Fig. 2.4(a)). We suspect this is due to the advantages of corrective annotation increasing as the model

converges, when more of the examples are segmented correctly and don't need adding to the training data as they would provide negligible utility beyond what has already been annotated. Our results show corrective annotation achieves competitive performance with a fraction of the labelled pixels compared to dense (Fig. 2.9(b)). These results align with [221] who confirmed that a large portion of the training data could be discarded without hurting generalization performance. This view is further supported by theoretical work [203] showing in certain cases networks will learn a maximum-margin classifier, with some data points being less relevant to the decision boundary.

The corrective training procedure performance had lower standard error after one hour (Fig. 2.4(a)) and particularly at the end (Table 2.3). We conjecture that the corrective annotation strategy stabilized convergence and increased the robustness of the training procedure to the changes in dataset with the fixed hyperparameters by allowing the specific parts of the dataset used in training to be added based on the weaknesses that appear in each specific training run.

In more heterogeneous datasets with many anomalies, we suspect corrective annotation to provide more advantages in comparison to dense, as working through many images to find hard examples will capture more useful training data. A potential limitation of the corrective annotation procedure is the suitability of these annotations when used as a validation set for early stopping, as they are less likely to provide a representative sample, compared to a random selection. Our annotation protocol for corrective annotation involved initially focusing on clear examples (Supporting Information, Notes S2.9.3) as in preliminary experiments we found corrective annotation did not work effectively at the very start of training. Training start-up was also found to be a challenge for other systems utilising interactive training procedures [62], indicating future work in this area would be beneficial.

Another possible limitation of corrective annotations is that they are based on the model's weaknesses at a specific point in time. This annotation will likely become less useful as the model drifts away to have different errors from those that were corrected.

One explanation for the consistently strong correlation on the root data compared to biopores and nodules is that the correlation with counts will be more sensitive to small errors than correlation with length. A small pixel-wise difference can make a large impact on the counts. Whereas a pixel erroneously added to the width of a root may have no impact on the length and even pixels added to the end of the root will cause a small difference.

A limitation of the RootPainter software is the hardware requirements for the server. We ran the experiments using two NVIDIA RTX 2080 Ti GPUs connected with NVLink. Purchasing such GPUs may be prohibitively expensive for smaller projects and hosted services such as Paperspace, Amazon Web Services or Google Cloud may be more affordable. Although model training and data processing can be completed using the client user interface, specialist technical knowledge may still be required to setup the server component of the system. To mitigate the hardware requirements and technical knowledge required for the initial setup, we have prepared an open source Jupyter notebook web ap-

plication (see Data Availability) which is made available via Google Colab and guides a user without specialist technical knowledge through setting up a RootPainter server using a freely available GPU. As part of the online supplementary material we also make a video available showing how to interactively train and use a biopores segmentation model with RootPainter (Video S1).

Although RootPainter automates the annotation process by providing an initial model prediction, the correction process, which involves manually drawing, could still become laborious, especially for larger more complex images. Interactive-segmentation tools such as grabber [24] may be complementary and could be investigated in further work to accelerate RootPainter's corrective-annotation process.

There are a limited number of root traits that can be exported from RootPainter in comparison to other root segmentation software applications such as faRIA [146]. This limitation has been addressed by the addition of a conversion utility, available from the RootPainter extras menu, that enables RootPainter segmentations to be conveniently processed with RhizoVision Explorer [179], facilitating the extraction of many more traits from the RootPainter segmentations.

In addition to the strong correlations with manual measurements when using corrective annotation, we found the accuracy of the segmentations obtained for biopores, nodules and roots to indicate that the software would be useful for the intended counting and length measurement tasks (Fig. 2.6(a), 2.6(b) and 2.6(c)).

The performance of RootPainter on the images not used in the training procedure indicate that it would perform well as a fully automatic system on similar data. Our results are a demonstration that for many datasets using RootPainter will will make it possible to complete the labelling, training and data processing within one working day.

2.6 Acknowledgements

Martin Nenov for proof reading and conceptual development. Camilla Ruø Rasmussen for support with using the rhizotron images. Ivan Richter Vogelius and Sune Darkner for support during the final stage of the project. Prof. Dr. Timo Kautz and Prof. Dr. Ulrich Köpke for provision of the biopore dataset. Guanying Chen, Corentin Bonaventure L R Clement and John Kirkegaard for helping test the software by being early adopters in using RootPainter for their root research. Simon Fiil Svane for providing insights into root phenotyping and image analysis.

We thank Villum Foundation (DeepFrontier project, grant number VKR023338) for financially supporting this study. Eusun Han is a Marie Curie Global Fellow working on a project SenseFuture (No.884364) funded by European Union's Horizon 2020 research and innovation program. The biopore dataset was provided from a study supported by the German

Research Foundation (Deutsche Forschungsgemeinschaft—DFG) within the framework of the research unit DFG FOR 1320.

2.7 Author contribution

AGS implemented RootPainter and wrote the manuscript with assistance from all authors. AGS, EH and JP designed the experiment. AGS and EH annotated and collaborated on the design of the study and the introduction. CG and MA captured and prepared the nodules data. NAFO tested the software and annotation protocol during development. DBD and KTK provided supervision and conceptual input. All authors read and approved the final manuscript.

2.8 Data Availability

The nodules dataset is available from [192]. The biopores dataset is available from [198]. The roots dataset is available from [190]. The client software installers are available from `https://github.com/Abe404/root_painter/releases`. The source code for both client and server is available from `https://github.com/Abe404/root_painter`. The created training datasets and final trained models are available from [199]. The colab notebook is available at `https://colab.research.google.com/drive/104narYAvTBt-X4QEDrBSOZm_DRaAKHtA?usp=sharing`

2.9 Supporting Information

2.9.1 Server Software Setup Instructions

Instructions on how to setup the RootPainter server.

For our tests we use a client-server architecture and run the client and server components of the system on different computers, using Dropbox to facilitate IO between them. We do not use any Dropbox specific functionality so any service which synchronises a folder between two computers should work. It is also possible to run the client and server on the same computer which will reduce lag and eliminate the need to use a third party service or network drive to sync files. We tested the server component using Python 3.7.5.

1. git clone --branch 0.2.4 `https://github.com/Abe404/root_painter.git`
2. cd root_painter/trainer
3. pip install torch==1.3.1 -f `https://download.pytorch.org/whl/torch_stable.html`
4. pip install -r requirements.txt

5. NOTE: pytorch installation may be more involved as it could require and consideration of the current CUDA version. We have tested using pytorch version 1.3.1 but would expect it to work with more recent versions. See `https://pytorch.org/get-started/previous-versions/` for more details on how to install pytorch.

6. python3 main.py

7. You will be prompted to specify a sync location. For our tests we used Dropbox and a folder named paper_rp_sync so we input ~/Dropbox/paper_rp_sync

8. If the folder doesn't exist then it will be created with the necessary sub folders (datasets, projects, models and instructions).

9. The system will start running and watching for instructions from the client. You must share access to the created folder with the users using your file share service (such as Dropbox) or network drive. The users will then need to input this when they first run the client software.

10. See section *Software Implementation* for further instructions for client software setup.

11. For our tests we ran the RootPainter server inside a tmux session but for more long running use cases a systemd service will likely be more robust. See `https://github.com/torfsen/python-systemd-tutorial` for instructions on creating a systemd service with python.

2.9.2 Keyboard Shortcuts

A table showing the keyboard shortcuts that can be used to speed up the annotation process when using the RootPainter client.

Table 2.4: Keyboard shortcuts for corrective annotation with RootPainter

Key	Function
Q	foreground brush
W	background brush
Z	undo
Ctrl+Shift+Z	redo
Alt+scroll	Change brush size
(Windows Key/Command key) + click and drag	Pan view
A	show or hide annotation
I	show or hide image
S	show or hide segmentation
Scroll	zoom

2.9.3 Corrective Training Protocol

Instructions on how to train a model with RootPainter using the corrective annotation protocol, as was done in this study.

Stage 1

- Start a timer immediately before starting to annotate
- Start training after clicking *Save & Next* for the second annotated image.
- Keep track of how many images you have annotated until you have annotated six images.
- Skip images which do not include clear examples of both classes.
- When images contain clear examples of both classes then label the clear and unambiguous parts of the image.
- Aim to label around 5-10 times as much background as foreground.
- Use a thinner brush to avoid boundaries when labelling the foreground class as these can be ambiguous and time consuming to label.
- After clicking *Save & Next* for the 6th image proceed to stage 2.
- Write down the image number for the 6th annotated image.

Stage 2

- For each image press S to view the segmentation. Instead of labelling everything which is clear, focus on labeling the parts of the image which have clearly been segmented incorrectly, whilst following the corrective annotation advice.
- Once you have proceeded through 10 images since the 6th annotated image then set pre-segment (from the options menu) from 0 to 1. Increasing the pre-segment setting causes the server to create segmentations ahead of time for upcoming images. This allows the user to progress through the images faster but presents a trade-off as they could potentially be out of date as they are segmented with the best model available at the time and not updated. Thus we only increase pre-segment once the user has worked through a few images, as their annotation time speeds up and necessitates the adjustment.
- Once you have proceeded through 20 images since the 6th annotated image then set pre-segment from one to two.
- Once one hour has passed on the timer then take a break for 30 minutes.

- After the 30 minute break then click Start Training again and proceed to annotate as before the break for another hour.
- After the second hour has been completed then stop annotating.
- Leave the network to stop training on its own.

2.9.4 Corrective Annotation Advice

Extra tips on how to execute the corrective training protocol effectively.

- Use a large brush for the background (green) as this makes it quicker label all the false positive regions.
- Focus time and attention on the incorrectly predicted parts of the image
- It is not a problem to label some foreground as foreground which has already been predicted correctly.
- It is also not a problem to label some background as background if it has already been predicted correctly.
- Errors to avoid include labelling a background pixel as foreground or labelling some foreground as background. These should be corrected using the eraser tool.
- It is not a problem to leave small areas unlabelled such as boundaries between foreground and background in the interest of avoiding errors whilst annotating quickly.
- Press I (capital i) to hide and show the image in order to better check the networks segmentation prediction for errors before proceeding to the next image.

2.9.5 Dense Annotation Advice

Tips on how to annotate densely, as was done in this study for comparison purposes.

- Set pre-segment (from the options menu) to 10 so that segmentation time does not impact ability to work through the images. Increasing the pre-segment setting causes the server to create segmentations ahead of time for upcoming images. For dense we don't care about the segmentations so by segmenting 10 in advance it means the client software will never stall their progression through the images because the segmentation has not yet loaded.
- Change the background colour from the default transparency level to a transparency level of 8%. This is because altering the brush transparency allows viewing the object of interest through the background annotation.

- Label each image as all background with a single click using the large brush and proceeded to explicitly annotate all objects of interest (using foreground brush) or ambiguous regions (using the eraser brush) before proceeding to the next image.
- Leave ambiguous regions such as boundaries as undefined, rather than labelling them as foreground or background.
- Once the time limit is reached, use the eraser tool to mark areas not yet annotated in the current image as undefined, stop annotating and click *Start Training*.

Nodule Threshold Plot

A plot showing correlation between automatic and manual nodules counts as a function of a nodule size threshold.

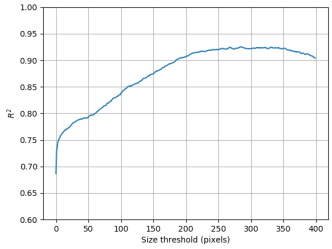


Figure 2.11: Correlation between automated and manual nodule counting as a function of size threshold for the automatically detected nodules. The thresholded nodules include only those above the specified area in pixels.

Video S1: RootPainter Biopore Model Training Video

A 42 minute video showing the training of a biopore segmentation model using RootPainter with corrective annotation. The video is available to download from the online supplementary material available with the published article: `https://nph.onlinelibrary.wiley.com/action/downloadSupplement?doi=10.1111%2Fnph.18387&file=nph18387-sup-0002-VideoS1.mp4`

RootPainter3D: Interactive-machine-learning enables rapid and accurate contouring for radiotherapy

Abraham George Smith[1, 2], Jens Petersen[1, 2], Cynthia Terrones-Campos[2, 3], Anne Kiil Berthelsen[2, 4], Nora Jarrett Forbes[1, 2], Sune Darkner[1], Lena Specht[2], Ivan Richter Vogelius[2, 5]

1 – Department of Computer Science, University of Copenhagen, Universitetsparken 1, 2100 Copenhagen, Denmark
2 – Department of Oncology, Rigshospitalet, University of Copenhagen, Copenhagen, Denmark
3 – Department of Infectious Diseases, Rigshospitalet, University of Copenhagen, Copenhagen, Denmark
4 – Department of Clinical Physiology, Rigshospitalet, University of Copenhagen, Copenhagen, Denmark
5 – Department of Health and Medical Sciences, University of Copenhagen, Copenhagen, Denmark

Published in Medical Physics on the 13th of January 2022 [196].

3.1 Summary

Purpose: Organ-at-risk contouring is still a bottleneck in radiotherapy, with many deep learning methods falling short of promised results when evaluated on clinical data. We investigate the accuracy and time-savings resulting from the use of an interactive-machine-learning method for an organ-at-risk contouring task.

Methods: We implement an open-source interactive-machine-learning software application that facilitates corrective-annotation for deep-learning generated contours on X-ray CT images. A trained-physician contoured 933 hearts using our software by delineating the first image, starting model training, and then correcting the model predictions for all subsequent images. These corrections were added into the training data, which was used for continuously training the assisting model. From the 933 hearts, the same physician also contoured the first 10 and last 10 in Eclipse (Varian) to enable comparison in terms of accuracy and duration.

Results: We find strong agreement with manual delineations, with a dice score of 0.95. The annotations created using corrective-annotation also take less time to create as more images are annotated, resulting in substantial time savings compared to manual methods. After 923 images had been delineated, hearts took 2 minutes and 2 seconds to delineate on average, which includes time to evaluate the initial model prediction and assign the needed corrections, compared to 7 minutes and 1 seconds when delineating manually.

Conclusions: Our experiment demonstrates that interactive-machine-learning with corrective-annotation provides a fast and accessible way for non computer-scientists to train deep-learning models to segment their own structures of interest as part of routine clinical workflows.

3.2 Introduction

Half of all cancer patients receive radiotherapy [41], which is associated with a range of dose dependent side effects [17]. Effective mitigation of these side effects requires accurate delineation of organs-at-risk, such as the heart and oesophagus [50, 127]. Manual delineation is still widely used but time-consuming in comparison to automated methods [208] and subject to large inter-observer variation [100, 250].

A review of auto-segmentation methods for radiotherapy indicated deep-learning methods and convolutional neural networks (CNN) in particular as representing the state-of-the-art [29]. See existing surverys of deep-learning for radiotherapy for explanations of machine learning, deep learning, and CNNs [132].

Although CNNs exhibit impressive performance when the training and testing data are drawn from the same distribution [134, 90, 208, 53, 224], variations specific to on-site clinical data may result in decreased performance [60, 59, 177]. For example it has been found that organ deformations due to an abdominal compression technique impaired the performance of an externally trained CNN model [54].

Training models on-site is a potential solution but can be challenging as training neural networks involves time-consuming trial and error [132] and hiring the appropriate experts is associated with high costs. A lack of large and high-quality publicly available datasets compounds the problem [232] as annotating

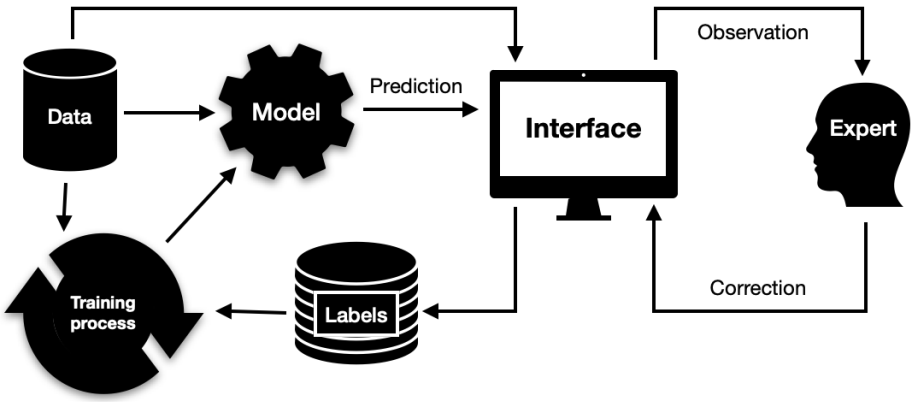


Figure 3.1: Interactive machine learning with corrective annotation puts the human-in-the-loop during the model training process.

large enough datasets for training deep-learning models with purely on-site data may be infeasible.

Corrective-annotation is an annotation sparsification strategy that results in a sub-region of each image being labelled. Annotation-sparsification strategies reduce the amount of user-annotation per-image. For a 3D heart segmentation dataset, annotation-sparsification was found to enable deep-learning models to be trained using just 20% of the labelled data whilst obtaining similar performance [260].

As opposed to approaches that rely on automatic methods to infer which regions of an image are most useful for labelling [260], corrective-annotation utilises human feedback (Figure 3.1) to identify problematic regions of the image for the model [62, 193, 82, 111]. Methods such as corrective-annotation which utilise a human-in-the-loop to improve machine learning implementations are known as interactive machine learning (IML) [83].

In prior work, corrective-annotation IML methods have been used to train deep-learning segmentation models for a diverse array of modalities, including 2D plant photographs [193, 72, 43], laser ablation tomography [49], histopathology [82] and 3D X-ray CT images of methane in sand [7].

To the best of our knowledge, no prior work has evaluated an IML approach applied to medical X-ray CT data. Therefore we implement an IML system for X-ray CT data and investigate its effectiveness for organ-at-risk segmentation.

We hypothesise that (1) semi-automatic contouring via an IML corrective-annotation method will provide similar accuracy to existing manual methods; And (2) that it will offer continuous improvements in contouring time as more images are annotated, eventually leading to substantial time savings.

3.3 Materials and Methods

We use a corrective-annotation approach to contour a large dataset of hearts by having a physician correct all generated contours during training (Figure 3.1). The process is semi-automatic, with the assisting model continuously learning as more images are annotated.

3.3.1 Dataset

We used X-ray CT scans from a cohort of patients that had been collated for a study on the association between mean heart dose and radiation-induced lymphopenia [213]. The CT scans were from patients who had started to receive radiotherapy to the chest region between 2009 and 2016 at Rigshospitalet in Denmark. The cohort was restricted to patients who were at least 18 years old, had dosimetric data available, a solid malignant tumour (excluding lymphoma) and a blood count both before and after receiving radiotherapy. This resulted in 933 X-ray CT scans from 923 patients, of which 308 had breast cancer, 291 had esophageal cancer, 56 had small-cell-lung-carcinoma and 268 had non-small-cell lung carcinoma. The 933 images had varying slice thicknesses. 209 images had a slice thickness of 3mm, 721 had a slice thickess of 2mm , and 3 had a slice thickness of 1mm. The number of voxels in each dimension varied. The total number of slices (depth) varied from 96 to 489 with a mean of 189. The width and height of the axial plane ranged from 512 to 658, with a mean of 537 voxels. No pre-processing was performed to normalize the images in any way. All files were saved in the gzip compressed NIfTI file format with extentsion .nii.gz. to facilitate convenient loading with nibabel library [25].

3.3.2 Software Implementation

We implemented RootPainter3D, which is based on RootPainter, an open-source corrective-annotation software application that utilises a client-server architecture and makes the necessary operations to train and use a deep-learning model for image segmentation available via a cross-platform graphical user interface [193].

The original RootPainter uses a variant of U-Net [169] modified to utilise group norm [244] and residual connections [76]. Compared to the 2D version we modify the software in several ways. The 2D convolutional layers were converted to 3D, resulting in a variant of the 3D U-Net architecture which is known to perform well for organ-at-risk segmentation [53] and sparse data [35]. The interface was modified to allow contrast settings to be adjusted, navigation and annotation of 3D images, with viewing enabled in both sagittal and axial views simultaneously. The data augmentation was removed. Although it is claimed data augmentation is critical to achieving favourable generalisation performance [178], results indicate the advantages of data-augmentation can be inconsistent and dataset specific [92].

RootPainter3D allows a user to inspect model predictions on a dataset that they work through sequentially - one image at a time. For each image, they initially define a bounding box in order to obtain predictions for a region containing the organ of interest. The user is able to assign corrections to the model predictions given for the defined region. When the user clicks 'save and next' in the interface the current annotation is saved to disk in a folder of annotations which is shared with a remote server with a more powerful graphics processing unit (GPU). The server component of the software continuously trains a 3D U-Net on the available annotations using stochastic gradient descent. For RootPainter3D, batches of 4 annotations, with their associated image regions are sampled from the annotation folder without replacement. The regions of the images used for training are only those where user-annotations are assigned. Supervised training is used but the annotations are sparse. Only the defined regions (what the user has specified as either foreground or background) are used for computing the loss which is used for updating the model weights.

The contrast settings (see Contouring procedure for more details). can be changed to a preset option in the view menu, such as mediastinal, which was used for our experiments.

3.3.3 Contouring procedure

To enable comparison to a manual method widely used in the clinic, 20 hearts were also delineated using the Eclipse treatment planning software from Varian Medical Systems, Inc.

From the 933 heart CT scans, the annotator delineated the first 10 in Eclipse and then all 933 in RootPainter3D and then the last 10 in Eclipse. This was done to allow the dice score (see section 3.3.6) to be computed between the contours done in RootPainter3D and Eclipse, and also to compare delineation time between RootPainter3D and Eclipse at the start and end of the RootPainter3D model training process.

Contouring in both Eclipse and RootPainter3D was done using the mediastinal hounsfield unit (HU) window which ranges from -125 HU to 250 HU. For both software applications, annotation can be performed in both the sagittal and axial views but for our experiment all annotation was assigned in the axial view. We used a single physician and contours were collected for research purposes, rather than as part of a routine procedure.

In RootPainter3D the user is able to view the model's initial segmentation in blue (Figure 3.2). They can annotate foreground (heart) regions in red and background (not heart) regions in green (Figure 3.2). A 3D segmentation consists of a model prediction for each voxel in an image. If annotating correctively, the red foreground annotation should correspond to model predictions that were false negatives, and the green background regions should be targeted towards false positive model predictions. Taking corrections into account, the corrected contour can be viewed using the outline view (Figure 3.3).

When contouring in Eclipse, every third slice was contoured and then interpolation was used to join the slices, as this is a standard clinical procedure

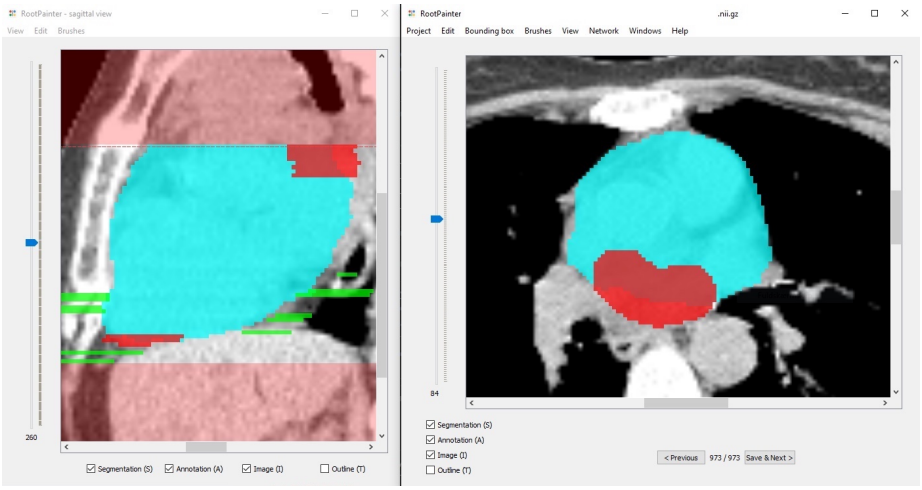


Figure 3.2: Screenshot of RootPainter3D software showing segmentation in blue and annotation in red for corrections to false negative regions and green for corrections to false positive regions. The area in light red is outside of the bounding box and is not part of the predicted or corrected region. The dashed red line shows the position of the axial slice in the sagittal view. This dashed line was added after the experiment in this study was completed based on user-feedback.

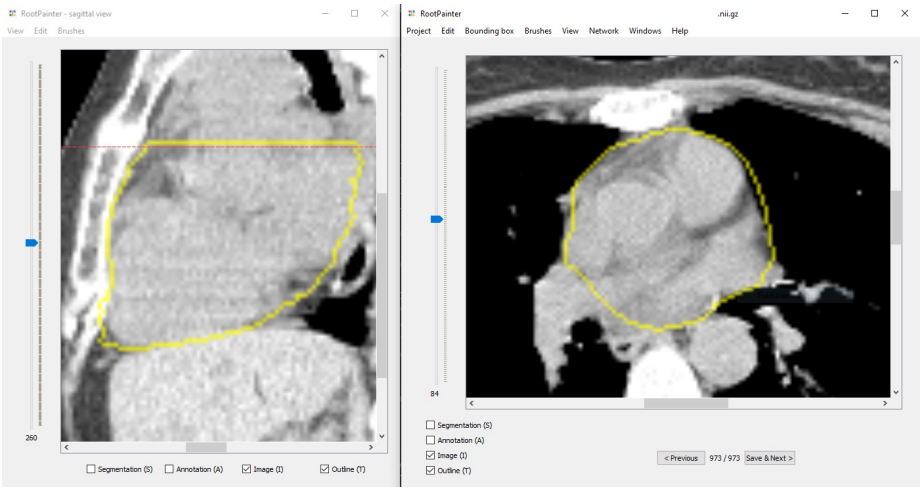


Figure 3.3: Screenshot of RootPainter3D software showing the outline view with segmentation and annotation hidden. The axial view is shown on the right and sagittal view on the left. The user is able to change what data is shown via keyboard shortcuts or with the checkboxes shown in each viewer.

used to save time compared to contouring all slices. The contours were not further modified after the interpolation had been applied. The same trained physician delineated all hearts in both RootPainter3D and Eclipse using the Danish guidelines for whole heart contouring [135].

In order to ensure the physician's familiarity with both software applications prior to the experiment, 20 hearts that were not used as part of the experiment were contoured in both applications.

In RootPainter3D, the initial image was densely delineated without considering the model's segmentation. After delineating the first image, the training procedure started. The subsequent 932 images were delineated by having the physician evaluate the model prediction and then correct all errors using the user interface.

3.3.4 Training

The training process for neural networks is a procedure where the network weights and biases are iteratively updated by a gradient descent algorithm, which compares the network outputs with the desired output [114].

When a user completes a delineation in the RootPainter3D client software, the annotation is saved to disk and training is started automatically by sending an instruction to the server.

The training routine, once started, first creates a model using Kaiming initialisation to assign random parameters [77], and then trains that model by continuously iterating over all saved annotations.

The training procedure treats newly saved images similarly to previously saved images as all images are equally likely to be used in the next batch of the training procedure. It loads the annotations from disk and uses these, combined with the image data, to train a model to predict the annotated regions correctly. For the RootPainter3D trainer, regions that are not annotated are set to 0 in both the network predictions and annotation. This means only the regions of the image explicitly annotated as either foreground or background by the annotator are used to compute the loss. The loss is then used to update the network weights. The loss function used in RootPainter3D is a combination of cross-entropy and dice loss [194].

A GPU is required for achieving optimal performance when training large parameter models such as deep neural networks. As the entire dataset cannot fit in the memory of the GPU, stochastic gradient descent is used to make network updates based on a subset (known as a batch) of the full available dataset. In our case the network is trained with a batch size of 4. We used two NVIDIA Titan RTX GPUs, with 24GB of memory each, using a data-parallel [39] approach. Due to GPU memory constraints, instead of using the full images, a sub-region (patch) is sampled from a random location that contains annotation, from each of the randomly sampled images in the training batch. The patch dimensions are 228, 228 and 52 voxels, corresponding to width, height and depth respectively.

For the RootPainter3D trainer application, once training has started, the process is continuous, even when the client application is closed. The trainer ap-

plication keeps track of a counter named *epochs_without_progress*. This counter is used to measure the length of time that the network has been trained without obtaining a new high score on the validation data.

3.3.5 Validation

In a machine learning context, the purpose of validation is to estimate model performance on unseen data. Networks with large capacity can overfit arbitrary datasets [258]. Fortunately, they have a tendency to learn smooth functions and overfitting can be conveniently mitigated with early stopping [31]. Early stopping involves checking model performance on a portion of the data not used for training. This allows generalisation performance to be estimated and a snapshot of the model weights to be taken before overfitting has substantially degraded performance.

Similarly to 2D RootPainter [193], a portion of the annotated images are assigned to the validation set instead of the training set. The first image is assigned to the training dataset, then the second image is assigned to the validation dataset. From then on images are only assigned to the validation dataset if the training dataset is at least 5 times as large as the validation dataset.

The validation dataset is used for model selection, in a way equivalent to early stopping. An epoch typically refers to a full iteration over all available training data. In this context we define epoch length in terms of the number of examples sampled from the training data and have this automatically adjusted in relation to the validation set size, which increases as more images are annotated. If no validation images have been saved then the epoch length is 128. Otherwise the length is $max(64, 2v)$ where v is the number of patches containing annotation in the validation set. Setting the length of the training period based on the validation set size ensures validation time does not overwhelm the training procedure, despite continuous addition of new cases.

At the end of each epoch the predictions for the model being trained are computed on the validation set and a dice score is computed using the available validation set annotations. If the model-in-training's dice score is higher than the most recently saved model so far then it will be saved to disk and used for generating the segmentation presented to the user in the client user-interface.

epochs_without_progress will get set to 0 in two cases. Firstly, when new data is added to the training or validation datasets. Secondly, when a new model is found which obtains a new high score on the validation set.

Network training procedures are stochastic and dependent on initial weights [13]. Therefore as the first initial weights are unlikely to be optimal, repeating the training procedure with different initial weights is a simple way to obtain better results. For this reason we introduced a restart procedure. The restart procedure was not functioning for the first 478 hearts, during which the network would stop after 60 epochs without progress.

The restarting behaviour, used for the latter portion of the experiment to further boost network accuracy, would start training the network from scratch after 60 *epochs_without_progress* were reached. If the newly trained model beat

the best model on the validation set so far then it would be saved (see Validation for more details). That means if the system is left unattended for long enough, it will likely find a new best model, and keep trying relentlessly until it does.

3.3.6 Dice evaluation

The dice score is a measure of overlap between structures. If two structures are exactly overlapping, the dice score is 1.0. For prediction p, annotation g, and voxel count N, dice score is defined as:

$$\frac{2(p \cap g)}{p \cup g} = \frac{2\sum_i^N p_i g_i}{\sum_i^N p_i + \sum_i^N g_i} \tag{3.1}$$

In order to evaluate the accuracy of the RootPainter3D contours before and after user correction, we compared the dice of both the initial predicted contour and the version after corrections with contours created manually in Eclipse for the same 20 images.

For each model generated heart contour, we also compute the dice score between the predicted heart contour and corrected version for 933 images. This approach is similar to previous studies that use manually corrected model predictions as ground-truth contours for evaluation [53, 67].

3.3.7 Interaction logging and annotation duration

To evaluate the extent to which the trained model assisted in reducing annotation time, we asked the annotator to log the specific time periods when they started and stopped annotating. We also automatically logged interaction events with the user-interface including when the user saved an image, opened a new image, moved to a different slice in the axial or sagittal views, changed zoom settings and mouse down and mouse release events.

To compute the amount of time the annotator spent on each image, we filter to interaction events inside their manually logged annotation period. For each file, we start accumulating interaction time from when the user opens that file until they open the next file. We also exclude periods of inactivity from our duration computation, which we define as no interaction events for at least 20 seconds.

3.3.8 Impact on radiation dose

We also compare the computed mean heart dose (Gy) when using the initial segmentation compared to the corrected contour.

To compare dose absorption we first obtain a 3D dose matrix with the same resolution as the associated image for each scan from the clinical dose plan. The dose matrix provides a cumulative level of radiation dose absorbed by each voxel in the planning CT scan. We compute the mean heart dose using the dose matrix combined with both the initial segmentation and corrected structure

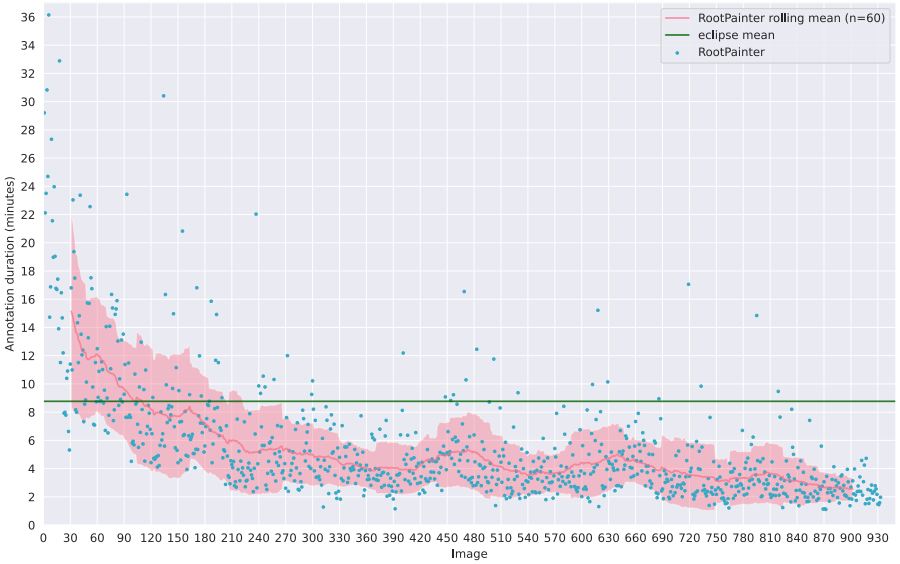


Figure 3.4: Delineation duration for each of the 933 images shown in order of delineation with the mean delineation duration of the 20 images contoured in Eclipse. RootPainter3D per-image delineation duration reduces over time.

independently. We then take the absolute difference between the two mean heart dose measurements to allow us to plot dose deviation over time as more hearts are annotated. The dose calculation is not repeated and we therefore rely on the 3D dose matrix from the clinical calculation algorithm at the time of treatment. This was the Analytical Anistropic Algorithm for earlier treatments and superseded by the Acuros XB Boltzmann equation algorithm of the Eclipse systems starting in 2014.

3.4 Results

Duration

In order to evaluate the annotation speed of RootPainter3D, we report delineation duration as a function of the number of images annotated (Figure 3.4). The first 10 and last 10 hearts are shown separately to highlight both the differences with Eclipse and RootPainter3D both before and after a period of extensive interactive annotation (Figure 3.5). RootPainter3D is initially slower and then becomes significantly faster after it has learned from more user corrections (Figure 3.5).

After around 110 images, delineation duration in RootPainter3D is less than the mean delineation of the 20 hearts contoured in Eclipse (Figure 3.4). With

Figure 3.5: RootPainter3D and Eclipse delineation duration for the first 10 and last 10 images out of the 933. RootPainter3D is initially slower than Eclipse but then becomes significantly faster. The last 10 hearts take an average of 2 minutes and 2 seconds to delineate in RootPainter3D compared to 7 minutes and 1 seconds when using Eclipse. Delineation in RootPainter3D is over three times faster than delineation in Eclipse.

Table 3.1: Dice scores between hearts delineated using RootPainter3D and Eclipse, showing strong agreement between methods. However, the RootPainter3D predicted and corrected contours are more similar to each other than either one is to the Eclipse contours. *pred* refers to RootPainter3D predicted contours and *cor* refers to the RootPainter3D corrected contours.

	Eclipse vs pred	Eclipse vs cor	pred vs cor
mean	0.945	0.952	0.991
std	0.008	0.007	0.005

the exception of temporary fluctuations, the RootPainter3D per-heart delineation duration continues to decrease as more hearts are contoured (Figure 3.4) becoming substantially faster than the comparison manual method with the last 10 hearts taking an average of 2 minutes and 2 seconds to delineate compared to 7 minutes and 1 seconds when using Eclipse (Figure 3.5). The recorded time includes both the time for the physician to evaluate the model's segmentation and assign any required corrections. The RootPainter3D delineations also maintain strong agreement with manual delineations of the same images (Table 3.1).

Accuracy

The dice score between the initial predicted heart and the corrected version are shown with running mean and standard deviation which were computed using a running average from 60 images (Figure 3.6). The dice scores of all hearts, excluding the first (which had a dice score of 0.2) are shown in figure 3.6. From

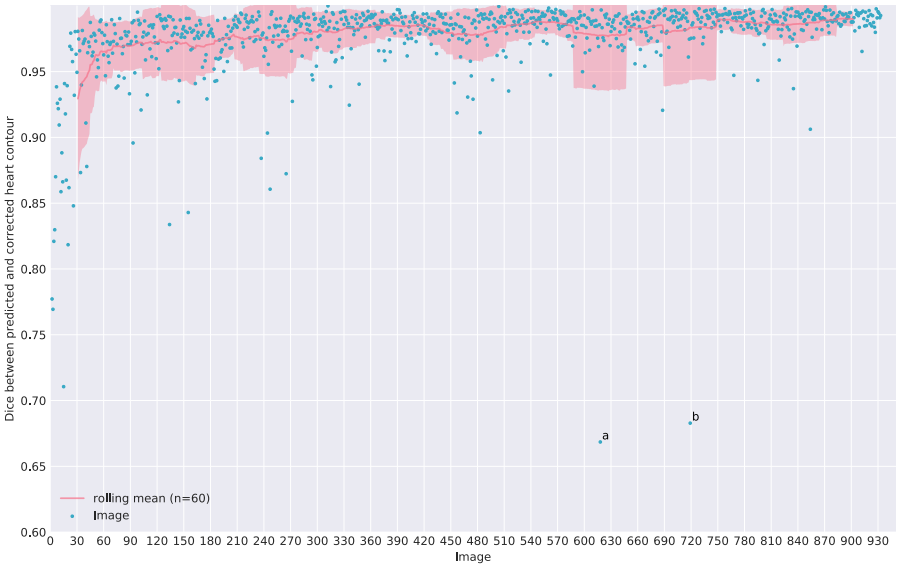


Figure 3.6: The dice score for 933 images shown in delineation order. The dice score compares the predicted vs. the corrected contour for each delineated heart. After the initial training period, the vast majority of dice scores are above 0.9, with only a few outliers dropping below. Outliers are labelled a and b.

the 300th heart onward, the vast majority of hearts have a dice score above 0.98 (Figure 3.10) with a few outliers having dice scores between 0.9 and 0.95 and just two extreme outliers having dice scores below 0.7 (Figure 3.6).

Axial slices from the outliers a and b labelled in figure 3.6 are illustrated in figure 3.7 and figure 3.8 respectively. These two hearts were the only ones out of the last 600 annotated which had a dice score lower than 0.9. In both cases the model appeared to be confused by a large tumour adjacent to the heart.

To show the model progression more clearly, in figure 3.10 the y-axis minimum is raised to 0.9, which includes most of the values. Although there are fluctuations, the mean dice score shows a trend of continuous improvement as more images are correctively delineated (Figure 3.10).

We found strong agreement between the hearts contoured in RootPainter3D and Eclipse (Table 3.1). All of the last 10 hearts in RootPainter3D had higher agreement with the manual Eclipse delineations after the corrections were assigned.

Figure 3.11 shows the mean heart dose and dose difference between predicted and corrected heart contours for all hearts where the absolute dose difference was less than 0.25 Gray. As shown in figure 3.12, only 4 of the model predicted hearts in the last 300 result in an error in dose above 1 Gray.

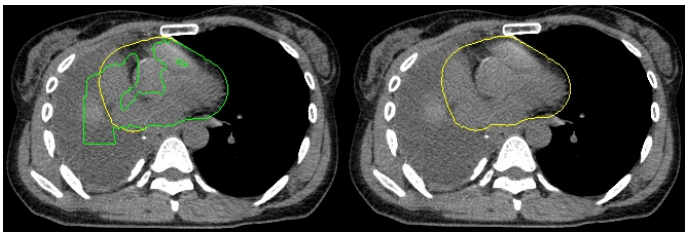


Figure 3.7: Outlier a: Dice score 0.67

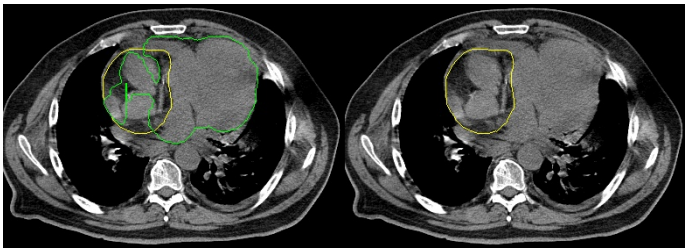


Figure 3.8: Outlier b: Dice score 0.68.

Figure 3.9: Outliers a and b. For each heart two axial slices are shown with contours overlaid. On the left the model prediction is shown in green with the user corrected heart in yellow. On the right only the corrected heart is shown. For both the outliers with low dice, the tumour was located adjacent to the heart.

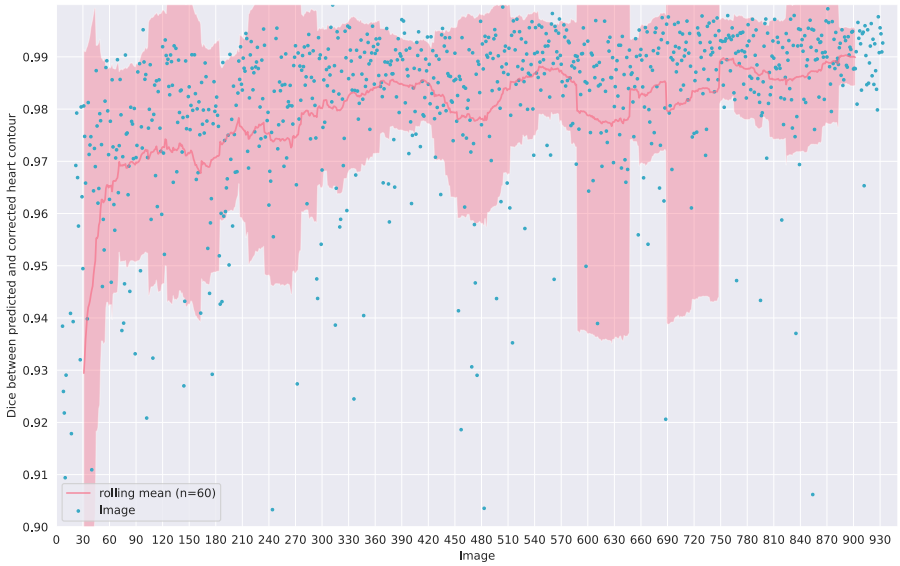


Figure 3.10: Dice score computed using the initial predicted image and the corrected version. Y-axis restricted to 0.9 to 1.0 to show improvement in mean over time. The mean dice score has fluctuations, but shows a general trend of improvement throughout the experiment.

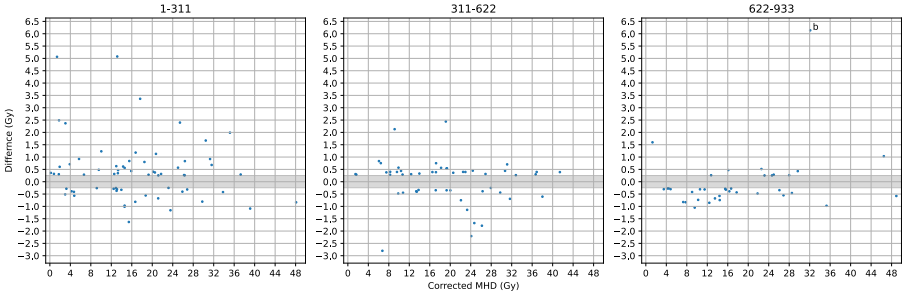


Figure 3.11: A plot of the outliers, showing dose differences between the predicted and corrected heart contours along with mean heart dose for hearts with absolute differences over 0.25 Gray for images 1-311, 312-622 and 623 to 923. The outliers became less severe in terms of dose difference as more images were annotated. Despite this, even in the later stages of interactive training (images 622-933) there was still a severe outlier with a difference of over 6 Gray. This outlier, labelled b, corresponds to the dice score outlier labelled b in figure 3.6 and shown in figure 3.8. In some cases a low corrected mean heart dose would still have a large dose difference, for example in images 622-933 one heart had less than 2 Gray mean heart dose but over 1.5 Gray difference between the predicted and corrected heart contours.

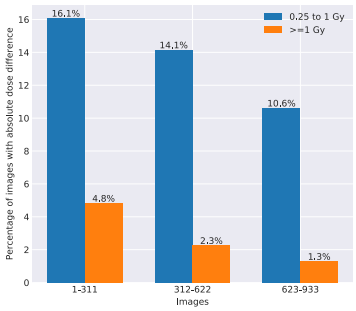


Figure 3.12: Percentage of hearts with absolute differences from 0.25 to 1 Gray and over 1 Gray for images 1-311, 312-622 and 623 to 923. The dose differences decrease as more images are annotated correctively.

3.5 Discussion

The strong agreement with the manual contours shown in table 3.1 supports our first hypothesis that RootPainter3D will result in accurate contours. Prior heart auto-contouring studies have observed a mean dice score of 0.925 between model predictions and manual delineation [53]. Disagreement between multiple heart delineators was found to give a dice score of 0.931 [251], indicating the difference between RootPainter3D delineations and our manually created delineations may be less than the difference between two annotators, even before corrections are assigned.

The decrease in per-image contouring duration as more hearts are annotated (figure 3.4) and significantly faster contouring for the last 10 hearts (Figure 3.5) support our second hypothesis that RootPainter3D will offer substantial time savings.

In Eclipse, only one in every three slices was manually delineated, with the others created using interpolation. Even though in RootPainter3D all slices were manually corrected, the contouring time was still more than three times faster than Eclipse (Figure 3.5). This discrepancy between the number of slices delineated vs interpolated also explains why RootPainter3D was slower than eclipse during the initial period of annotation (Figure 3.5).

The dice scores between the RootPainter3D corrected contours and Eclipse contours are lower than the dice scores between the RootPainter3D predicted contours and corrected contours (Table 3.1).

As the predicted and corrected contours are not done independently, it is expected that they will have higher agreement than two independently created contours. For quantifying errors in an existing delineation, there are some cases where this may be a strength. When delineating from scratch in noisy images and areas with little contrast, a drop in dice may be caused by differences in delineation along ambiguous border regions. With corrective-annotation, the annotator focuses on clear errors and the measured dice is therefore an indication

of how often the system makes such clear errors and these may be more relevant to quantify. For creating datasets for machine learning, corrective-annotation could therefore reduce label noise consisting of natural-perturbations. Although deep-learning is relatively robust to label noise [168], perturbations along boundaries are particularly problematic for U-Net model training [79].

In prior studies, correcting contours, as opposed to contouring from scratch has been shown to increase consistency and reduce inter-observer variation [40].

The two outliers with low dice scores were in patients with anatomical abnormalities that will also likely be correlated to high dose to the heart (Figure 3.9) and in one case would have resulted in an error in dose quantification of over 6 Gray had it not been corrected (Figure 3.11). This is a potential limitation of using fully-automatic contouring for dose-response modelling and shows the importance of carefully reviewing the results of such models in clinical applications.

Some of the larger errors are likely due to rare anatomical or pathological features or low contrast between the structure of interest and adjacent structures (Figure 3.9). Improvements in auto-segmentation performance are known to depend on image quality but rare artifacts or features are inevitable and will continue to require human intervention.

A typical limitation of auto-contouring studies is the small number of scans used for evaluation, with less than 100 being typical [177]. By evaluating our contouring software on over 900 hearts we were able to identify outliers that would have otherwise been missed (Figure 3.9). It is clear from figure 3.6 that smaller datasets may exclude such outliers by chance, resulting in over-optimistic characterisations of performance on unseen data.

Despite editing auto-contours being identified as a barrier to the adoption of automatic methods [251], our results, in alignment with previous studies [124, 226, 108], demonstrate that correcting auto-contours saves time compared to a standard clinical workflow. In prior work, time-savings have been demonstrated, even when the predicted structures are of particularly low quality, with a dice score as low as 0.46 [64].

The benefits of corrective-annotation to routine clinical contouring are clear as correcting inaccurate contours is already essential to optimise treatment planning [50, 127]. In addition to being usable by non-experts [48] and providing performance improvements in comparison to fully-automated systems [84], IML can provide trust and quality control benefits [133] which are of particular interest in a radiotherapy context.

The IML process (Figure 3.1) provides feedback to the annotator on how much data is necessary to train a model to a given level of accuracy and due to their extended exposure to model behaviour, provides physicians with control of model training and insight into the strengths and limitations of an AI system, attributes needed to ensure clinical adoption [28].

[160] characterise IML systems as being used for task-completion or model-building. As opposed to prior evaluations of IML with corrective-annotation for model-building [193], in this study we also demonstrate the potential for assisting in task-completion.

We used mean heart dose computation accuracy to measure model quality. Although it brings our results closer to a value familiar to the working radiation oncologist, the utility of mean heart dose for understanding cardiac toxicity has been called into question by recent studies [85]. With faster contouring capabilities, IML systems such as RootPainter3D provide capabilities to contour more structures in less time, making it possible to delineate organ sub-structures in a larger cohort of patients.

We motivated the omission of data-augmentation based on results that indicate the benefits may be dataset specific [92]. More recently, the same authors conducted a more thorough analysis, providing stronger indications that a sensibly designed data-augmentation procedure would provide benefits across datasets and organs [96]. Thus we expect the addition of data-augmentation would further improve our results.

We observed that many time consuming contour adjustments were minor changes along the boundary which are in many cases unlikely to have a large impact on dose. This aligns with previous observations [26], [26] and indicates a need for methods that guide delineators in determining which corrections are meaningful for dose-planning.

The combination of electronic health records, stored dose matrices and CT scans from record and verify systems has great potential to increase our understanding of radiation dose effects for normal tissues [232].

Prior work investigating the relationship between radiation exposure and lymphopenia for a cohort of 901 patients was completed by registering all scans to a single reference patient [1]. Our proposed method would compliment such studies by enabling efficient segmentation of structures of interest for each individual patient CT scan, potentially mitigating errors that may result from the registration process.

Our software is made available open source and we emphasise that with a restarted training process, the same application can learn to segment other organs of interest in a research setting. As we started training from random weights and use a fairly generic U-Net architecture, we expected the performance to be similar.

The risks when working with new data and multiple clinicians will need to be carefully considered before such systems can be adopted in the clinic, and new tests may need to be designed to ensure robustness and quality [233].

For technical teams including physicists and programmers, an open source solution may offer the needed flexibility to modify, extend and experiment with altered versions of the software. For less technical teams, software with existing commercial support may be a preferred choice.

Our proposed method immediately utilises all user corrections in the training procedure. Although the lack of commercial support may be a barrier to uptake, compared to commercial offerings, our method provides a way for teams to continuously adapt models to specific onsite issues without waiting for updates from third-party vendors.

3.6 Conclusion

Our results demonstrate the benefits of continual-learning with corrective-annotation, by showing how contouring time can be continually reduced whilst maintaining accuracy. The model-training was initiated and all annotation added via the graphical-user-interface, demonstrating that our proposed software provides a user-accessible way to train and use deep-learning models to semi-automatically contour large datasets.

It may be some time before such systems can be made widely available to clinics, as medical devices utilising continual-learning are yet to be approved by the FDA [115, 38].

3.6.1 Acknowledgements

We thank Thomas Carlslund and Kurt Nielsen for IT infrastructure support and Agata Wlaszczyk for proofreading. We would also like to thank Katrin Elisabet Håkansson, Mirjana Josipovic and Emmanouil Terzidi for feedback on early versions of the software and Deborah Anne Schut for feedback on the heart contouring procedure. We also thank Mikkel Skaarup for feedback on experimental design and gratefully acknowledge our financial support from Varian Medical Systems and the Danish Cancer Society (grant no R231-A13976).

3.7 Availability of source code and trained model

Source code is available at this HTTPS URL. The trained heart model is available upon request from Ivan Richter Vogelius (ivan.richter.vogelius@regionh.dk)

3.8 Conflict of interest

The authors declare no conflict of interest.

Localise to segment: crop to improve organ at risk segmentation accuracy

Abraham George Smith[1, 2, *], Denis Kutnår[1, 2], Ivan Richter Vogelius[2], Sune Darkner[1], Jens Petersen[1, 2]

1 – Department of Computer Science, University of Copenhagen
2 – Department of Oncology, Rigshospitalet, University of Copenhagen

Submitted to Medical Physics. Available as a pre-print on arxiv [197].

Abstract

Background: Increased organ at risk segmentation accuracy is required to reduce cost and complications for patients receiving radiotherapy treatment. Some deep learning methods for the segmentation of organs at risk use a two stage process where a localisation network first crops an image to the relevant region and then a locally specialised network segments the cropped organ of interest.
Purpose: We investigate the accuracy improvements brought about by such a localisation stage.
Method: We compare a two-stage approach using localisation to a single-stage baseline network trained on full resolution images using random patch sampling. We use five datasets from the Medical Segmentation Decathlon, including the heart left atrium (MRI), liver (CT), pancreas (CT), spleen (CT) and prostate (MRI). We split each dataset into training, validation and test subsets for our experiments.
Results: The two stage approach including automatic localisation provided significant increases in segmentation accuracy for the spleen, pancreas and heart. Localisation improved both training time and stability compared to the baseline method. We also observe increased benefits of localisation for smaller organs.

Conclusion: Automatic, and to a greater extent manual localisation can reduce training time and increase segmentation accuracy, especially for smaller organs.

Introduction

More than 50% of cancer patients receive radiotherapy which is associated with a range of dose dependent side effects. Delineation of organs at risk on treatment planning scans is crucial to minimise complications [50, 127]. Manual delineation is possible and still widely used but in comparison to automated methods is time consuming [208] and subject to large inter-observer variation [100]. Therefore, methods to improve the accuracy of automated methods are required. A review of auto-segmentation methods for radiotherapy is presented by Cardenas et al. [29] with deep learning methods and convolutional neural networks in particular representing the state-of-the-art.

Organ localisation has been used for a variety of tasks in image analysis and can reportedly improve segmentation accuracy whilst reducing computational memory and processing time requirements [246].

Kutnar et al. [113] found a two-stage localisation approach to be effective for the segmentation of lacunes in brain MR images and Gros et al. [67] found spine centerline localisation to provide state-of-the-art spinal cord segmentation accuracy. Feng et al. [53] proposed a two stage approach using cropped 3D images, where a similar 3D U-Net was used for both the initial organ localisation and segmentation stages. They claim their approach is more data efficient due to the use of voxel labels in the training of the localisation network. The method proposed by Feng et al. [53] is appealing as it uses the same method (segmentation with 3D-U-Net [34]) for both localisation and segmentation which simplifies both concept and implementation.

Although the method obtains competitive accuracy [251], an ablation analysis or baseline comparison method is lacking. Therefore, we conduct a more focused investigation to measure the accuracy gains brought about by such an approach to localisation. We hypothesise that localisation will improve organ at risk segmentation accuracy, demonstrated by a significant increase in dice. To the best of our knowledge, this hypothesis has not been tested in a focused investigation.

Methods

Dataset

To evaluate the effect of localisation on a diverse array of organ at risk segmentation tasks, we used the spleen, pancreas, prostate, liver and heart (left atrium) datasets [187] from the Medical Segmentation Decathlon [10]. We used only the original training sets, as this portion of the data has corresponding labels available for download. To facilitate the training and evaluation of deep

learning models for image segmentation, we split the downloaded images and labels into our own training, validation and test subsets with sizes of 60%, 20% and 20%, respectively (Table 4.1). This ratio between training, validation and test data was chosen as it is typical for deep learning model training.

Table 4.1: Number of images included in each of the training, validation and test datasets for each of the organs.

organ	training	validation	test
spleen	25	8	8
pancreas	169	56	56
prostate	19	7	6
liver	41	14	14
heart	12	4	4

Implementation

We used PyTorch [151] (Version 1.13.1) and implemented a 3D U-Net [34] which is an encoder-decoder style semantic-segmentation architecture. For all experiments we use 64GB of RAM and two NVIDIA 3090 RTX GPUs.

When performing semantic segmentation using convolutional neural networks, GPU memory is often a bottleneck. Due to this limitation there is a trade off between batch size, which is the number of images used in each training update and patch size, which is the size of the images used during training. Larger input patches allow more context to be considered for each voxel or pixel classification decision and have been found to improve accuracy [87]. Therefore we used an input patch size of 64x256x256 for all experiments as this was the largest we could fit in GPU memory. However as such large input patches take up more GPU memory they force a reduction in batch size. Therefore we used a batch size of 2 for all experiments with one instance (input patch) on each GPU, utilising a data-parallel approach, meaning the training batch is split accross the GPUs. Small batch sizes can be problematic for the commonly used batch normalisation method [91, 244]. Therefore we used group normalisation [244] after each layer as it performs well when small batch sizes are used and has been found to be effective for 3D medical image segmentation tasks [141].

We use a loss function which is a combination of dice [204] and cross-entropy as this has been found to be effective when dealing with class imbalanced datasets [194, 206]. Although in [53] the authors used cross-entropy with importance weights for their main experiments, they mentioned that they also found a combination of cross-entropy and dice loss to both stabilise and accelerate the training process. Another disadvantage of cross-entropy as opposed to dice loss is that organ specific importance weights require manual tuning. We used zero padding in the convolution operations to allow our 3D network to produce an

output segmentation with the same size as the input patch.

For all experiments we used the Adam optimiser [106] with a learning rate of 0.0001. For each training run we initialise the weights using He [77] initialisation. We used check-pointing and early stopping [139] to mitigate over-fitting. Check-pointing involves saving the model weights to disk during the training run. We computed the dice on the validation set at the end of each epoch and only saved models which obtained a new highest dice. There are various way to implement an early stopping procedure [157]. Our stopping criterion used the number of epochs since an improved dice had been found, a parameter which is a commonly referred to as patience. We set patience to 20 for all experiments, thus each training run would stop after 20 epochs had passed since a new highest dice score on the validation set had been obtained.

To mitigate the possibility that the results were due to chance, for each organ and method, training runs were repeated until 10 runs had converged, where convergence was defined as the model having at least 0.1 dice on the validation set after 20 epochs.

The methods campared include a baseline full resolution segentations approach using 3D patches, a two stage localisation approach and a method involving only organ segmentation, where the ground truth was used to localise (Figure 4.1). In the following sections we describe the three different approaches we experimented with to evaluate the benefits of localisation.

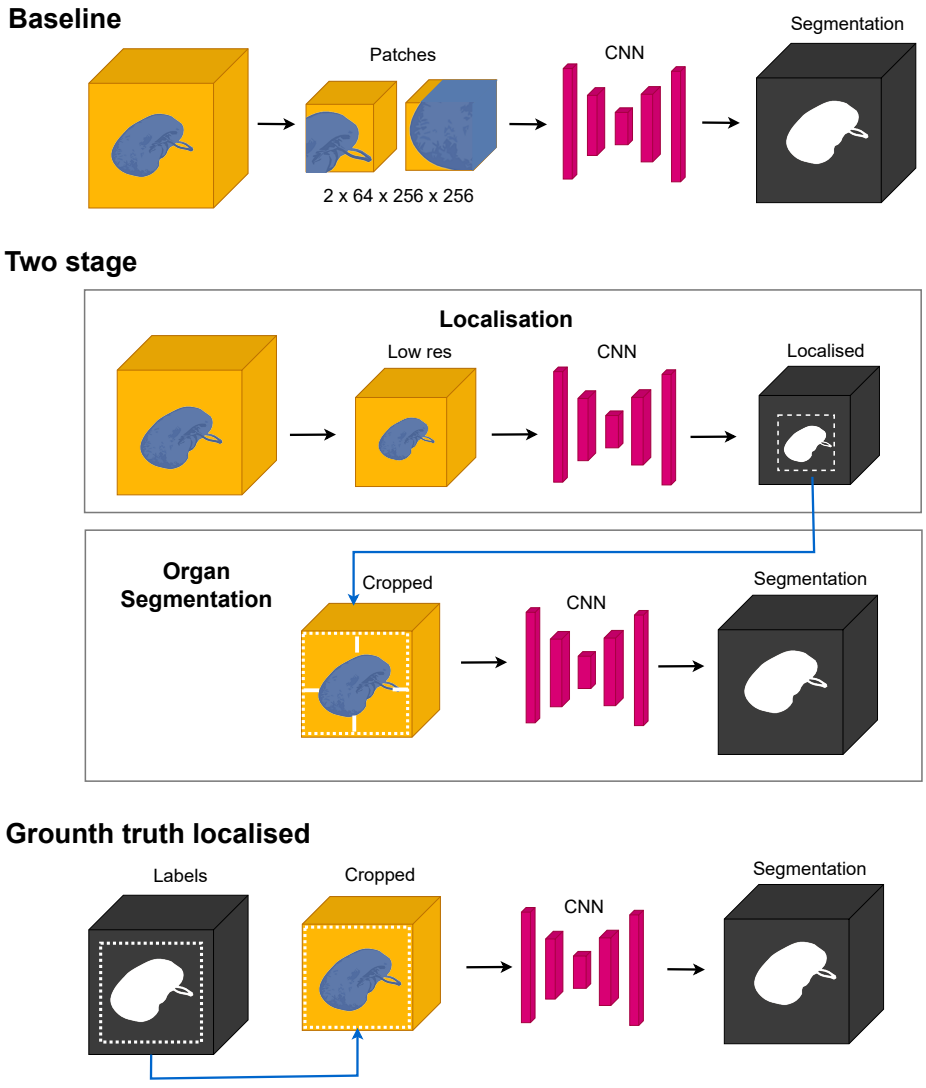


Figure 4.1: Illustration showing the three different methods compared, including the baseline, two stage involving both a localisation network and organ segmentation network and the organ segmentation network that uses the ground truth to localise.

Baseline - full resolution segmentation

In order to evaluate the advantages of the two stage localisation process we trained a single stage baseline network. For each training instance we sample a patch with random location within the image and the corresponding location

from the annotation. We enforced that at least 80% of the selected patches contained foreground annotation. Such biased instance selection is a relatively common practice as otherwise most patches would not contain foreground which can cause convergence problems.

Localisation network

In order to train the localisation network we created a low resolution version of the dataset by resizing the images and annotations down to a half their width and height and a third of their depth. We then trained the network to predict the annotations which were also resized to match the reduced resolution images. We created these low res images using the resize function from scikit-image [227] (Version 0.17.2).

Organ segmentation network

To train the organ segmentation network, we first created a dataset of images and annotations which were cropped by taking the region of the image including the organ with 15 voxels padding on each side to include some background context. To ensure enough padding was included on each side of the organ, even if the organ was at the edge of an image, the images were zero padded by 15 voxels on each side before cropping to the organ. The organ segmentation network was trained independently using these cropped versions of the original images and ground truth annotations, without regard to the output of any particular localisation network.

Ground truth localisation

We also evaluated an approach using a localisation stage which utilises the ground truth labels. We do this to access the advantages of localisation given an accurate bounding box.

Two stage localisation & segmentation pipeline

In order to segment the full resolution image with a preliminary localisation step we implemented a two stage process. We first computed a low res version of the image and then segmented it using the localisation network. We then identified the organ as the largest connected foreground region in the low res segmentation. We then segmented the corresponding region in the full resolution image with padding on each side of the organ as described in the above *Organ segmentation network* section. To perform this two stage segmentation, we pairwise couple the localisation networks with the organ segmentation networks chronologically, thus the i'th organ network trained is coupled with the i'th localisation network.

Metrics

During training we computed dice on both the training and validation data. For the final model that was automatically selected at the end of the training run, dice was computed on both the randomly selected validation and test sets using the full resolution segmentations and annotations.

The two sided t-test as implemented in SciPy [231] (Version 1.5.4) was used for testing for significant differences between the accuracy of the methods on both the validation and test datasets. We also record time for each of the training runs to converge.

Results

Training

Table 4.2: Average training time (minutes) for each of the three types of networks for each of the organs.

	baseline	low res	organ
spleen	74.9	10.8	10.1
pancreas	477.6	65.1	91.2
prostate	6.4	2.0	2.2
liver	253.2	42.8	110.0
heart	22.2	3.7	2.5

For each organ, the baseline approach is substantially slower than the other methods and for all organs takes longer to converge than both the low res and organ networks combined (Table 4.2).

Both the baseline method and low res networks had less stable performance during training compared to the organ network, with larger fluctuations in the dice (Figure 4.2). The organ segmentation network always converged (Figure 4.3a). With the baseline and low res networks, convergence is similarly likely, with 83% of the baseline training runs and 85% of the low res training runs converging. The varying rates of convergence (Figure 4.3) reflect the difference seen in training stability (Figure 4.2).

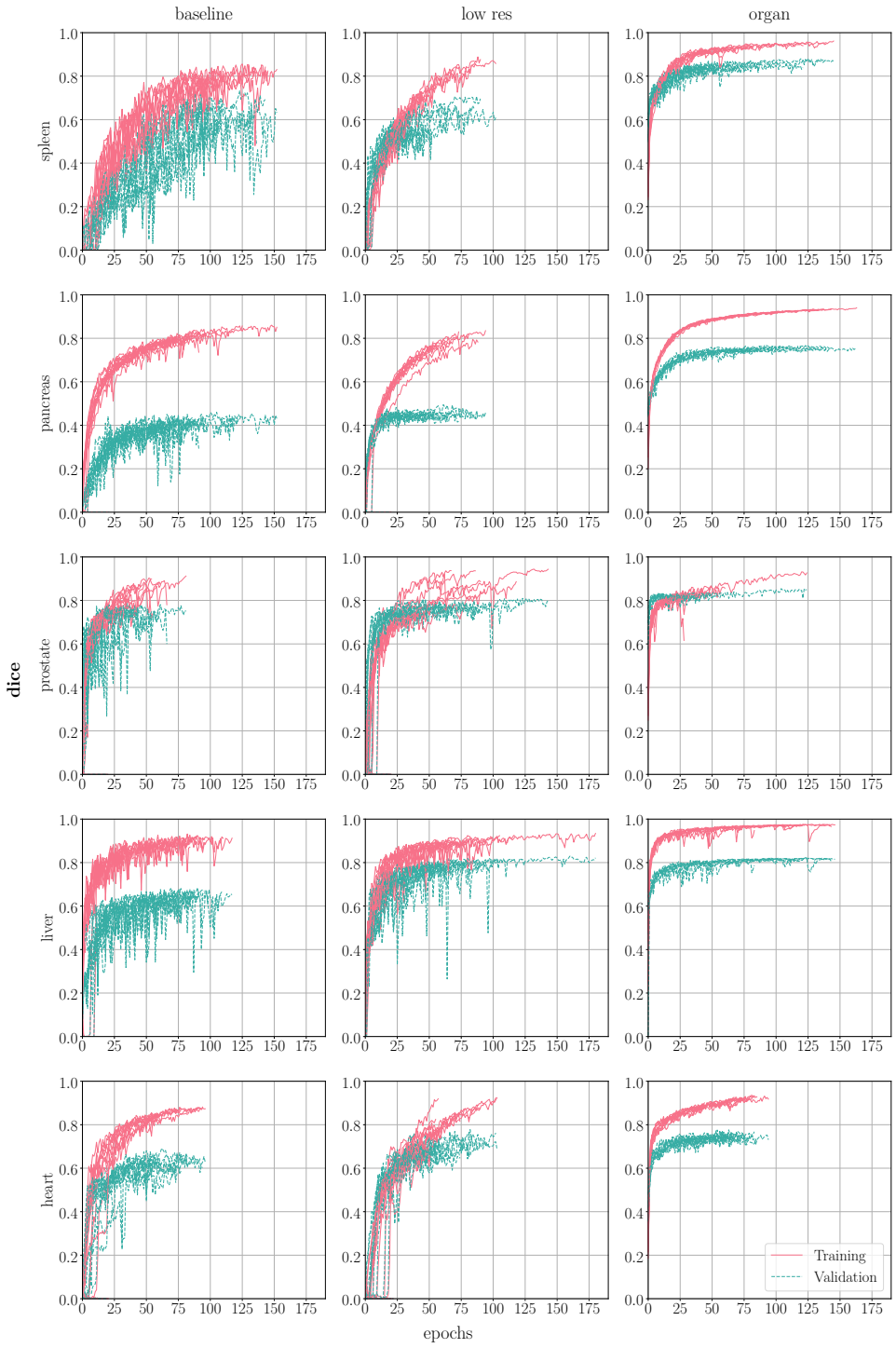


Figure 4.2: Validation and training dice are shown for each epoch for each of the 10 training runs for each organ and for each method, resulting in 150 training runs in total. Only training runs which successfully converged are shown.

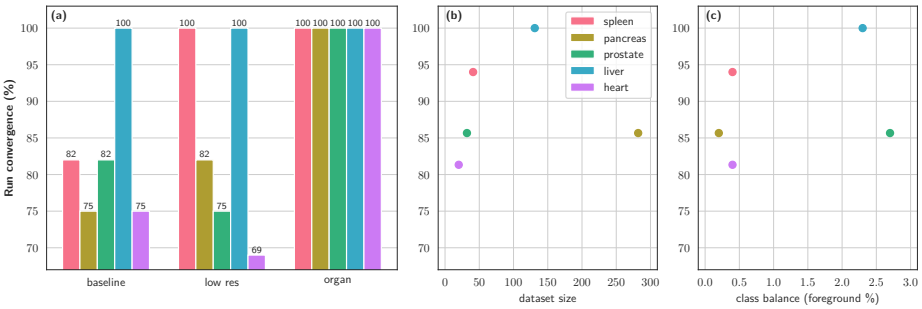


Figure 4.3: Convergence rate for (a) each method and organ, (b) as a function of dataset size and (c) as a function of class balance (average foreground percent).

Validation

Table 4.3: Average dice on the validation set for the baseline network compared to the two stage approach with both predicted and ground truth localisation. Values which are significantly higher than the baseline are shown in bold.

	baseline	two stage	ground truth localised
spleen	0.6491 ± 0.0997	**0.8142 ± 0.0221**	**0.8619 ± 0.0092**
pancreas	0.4372 ± 0.0148	**0.6674 ± 0.0146**	**0.7564 ± 0.0096**
prostate	0.7744 ± 0.0109	0.7699 ± 0.0149	**0.8323 ± 0.0076**
liver	0.6661 ± 0.0188	0.7044 ± 0.0694	**0.7807 ± 0.1061**
heart	0.6547 ± 0.0239	**0.7018 ± 0.0281**	**0.7612 ± 0.0087**

For the validation sets, the dice was significantly higher for the two stage approach compared to the baseline method for the heart ($p < 0.001$), spleen ($p < 0.001$) and pancreas ($p < 0.001$). For the liver, although the two stage approach appears it may offer some improvements, the difference was not significant ($p = 0.11$). For the prostate there was no significant difference ($p = 0.44$).

On the validation set, the difference between the ground truth localised method and baseline was significant for the liver ($p < 0.05$) and highly significant for the heart, spleen, pancreas and prostate ($p < 0.001$). For all organs except the liver ($p = 0.07$), the benefits of ground truth localization are significant compared to using the localization network to provide the cropped region ($p < 0.001$).

Test

Table 4.4: Average dice on the test set for the baseline network compared to the two stage approach with both predicted and ground truth localisation. Values which are significantly higher than the baseline are shown in bold.

	baseline	two stage	ground truth localised
spleen	0.4433 ± 0.1162	**0.6503 ± 0.0538**	**0.8255 ± 0.0086**
pancreas	0.4366 ± 0.0205	**0.6519 ± 0.0136**	**0.7397 ± 0.0063**
prostate	0.797 ± 0.0342	0.7204 ± 0.0244	**0.8361 ± 0.0145**
liver	0.6039 ± 0.0214	0.6096 ± 0.0664	**0.7156 ± 0.1028**
heart	0.5764 ± 0.0311	**0.623 ± 0.0523**	**0.7508 ± 0.0178**

On the test set, the two stage dice score was higher than the baseline for the spleen ($p < 0.001$), pancreas ($p < 0.001$) and heart ($p < 0.05$). For the liver the difference was not significant ($p = 0.8$). The test set dice was significantly higher with the baseline approach compared to the two stage method for the prostate ($p < 0.001$).

When using the ground truth labels to localise, the increase in organ network dice compared to the baseline was highly significant for the heart, spleen and pancreas ($p < 0.001$) and significant for the prostate and liver ($p < 0.05$).

The benefits of ground truth localization were also highly significant compared to using the localization network to provide the cropped region for the heart, spleen, pancreas and prostate ($p < 0.001$) and significant for the liver ($p < 0.05$). We found that smaller organs, as a percentage of scanned region tend to benefit more from localisation (Figure 4.4).

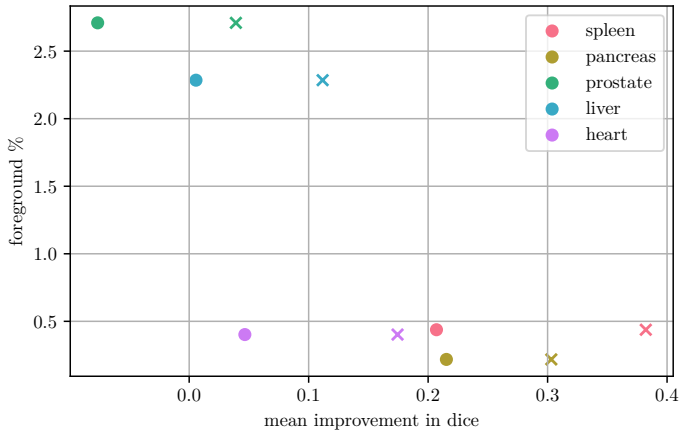


Figure 4.4: Benefit of localisation for each of the datasets for both predicted localisation region (o) and when the ground truth location is provided (x). The mean improvement in dice is calculated by subtracting the mean baseline dice from the mean localised dice. Foreground % is the percentage of the voxels in a scan that belong to the organ as opposed to the background, where background is considered as all voxels outside of that particular organ.

Discussion

Although the significant improvements in dice for the majority of datasets confirm our hypothesis that localisation improves organ at risk segmentation accuracy, the baseline performed stronger than expected in comparison to the two stage localisation approach, even out-performing the localisation approach on the prostate test set.

The mean organ volume as a percentage of total image volume ranges from 0.2% for the pancreas to 2.7% for the prostate. This represents an extreme class imbalance, particularly for the pancreas, spleen and left atrium. Class imbalance is known to have detrimental effects on the performance of machine learning models [122] and convolutional neural networks in particular [27]. If not addressed, a class imbalance problem may lead to algorithms tending to predict only the majority class [47]. Gros et al. [67] argue a two stage approach involving localisation is able to mitigate issues caused by class imbalanced data. The trend of an increased benefit of localisation for smaller organs (Figure 4.4) is expected, because for smaller organs the class balance issue becomes more severe and if the organ becomes large enough there will be negligible difference between the baseline and localisation approaches.

One explanation for the good performance of the baseline could be the random selection of patches during the baseline training procedure. This ran-

dom selection could have provided some augmentation benefits similar to random cropping. When the organ segmentation network encountered unexpected anomalies it may have been less equipped to handle them. Xu et al. [247] trained an organ segmentation network using a region containing the organ of interest but with variation in the amount of padding around it. Varying the amount of context around each organ during training may be key to a two-stage localisation network that provides consistent advantages in accuracy compared to the single stage baseline line method.

The baseline and low res network training instability, including fluctuations in dice (Figure 4.2) is likely related to the challenges with class imbalance. Although the baseline network had biassed instance selection to include foreground batches more frequently, its task was likely more complicated compared to cropped organ segmentation as the baseline network must learn to segment regions further away from the organ. For the baseline approach, the patches used in training will have also been less consistent, including varying amounts of the organ of interest or sometimes only background regions.

The heart (left atrium) had the lowest rate of convergence on average, which is likely due to it having both a relatively small dataset (Figure 4.3b) and a large class imbalance (Figure 4.3c). Our condition for convergence was based on accuracy, which typically increases with training dataset size [80]. An exception is the pancreas, with the largest dataset, yet a convergence rate of only around 85% (Figure 4.3b), which may be due to the high class imbalance in this dataset (Figure 4.3c).

Reduction in model training time is critical for both workflow optimisation and carbon footprint [9]. Slow training may also hinder novel interactive-machine-learning approaches that depend on model adaptation to support a feedback loop between annotator and model [196, 195, 241, 189]. We found that the baseline method had slower convergence and longer training time compared to the localisation approach, even when considering that the localisation approach involved training two networks (Table 4.2). The slow convergence of patch based training in comparison to other approaches has been observed in previous work evaluating methods for brain tumour segmentation [23].

A potential drawback of the two stage localisation approach is the additional complexity of training two networks. A potential limitation to the network architecture used in this study is the use of zero-padding to ensure that the network input and output had consistent size. In some cases zero-padding has been found to increases errors on the edge of a patch by as much as 35% [87].

The consistent benefits of using ground truth to localise the region for the organ segmentation network (Table 4.4) motivate the use of a manual bounding box in cases where accuracy improvements are required for smaller organs, an approach that has been used for prior studies in interactive machine learning for organ at risk segmentation [196]. Manual localisation would also be feasible with interactive segmentation methods, such as the approach proposed by Rasmussen et al. [164] where organ extremities are input to guide the predicted contour.

Conclusion

Our results show the advantages of both manual and automatic localisation for organ at risk segmentation in terms of both training time, convergence rate and segmentation accuracy, especially for smaller organs where class imbalance causes challenges for conventional approaches to segmentation model training.

Acknowledgements

This work was supported by the Danish Cancer Society (grant no R125-A7989)

Availability of data and materials

Source code is available at `https://github.com/Abe404/localise_to_segment`. The Medical Segmentation Decathlon dataset is available to download from `http://medicaldecathlon.com/`.

Conflict of interest

We declare no conflict of interest.

Corrective-annotation auto-completion enables faster organ contouring

Abraham George Smith[1, 2, *], Jens Petersen[1, 2], Isak Wahlstedt[2, 5], André Pedersen[6], Signe Lenora Risumlund[2], Mette Van Overeem Felter[3], Vibeke Nordmark Hansen[2], Ivan Richter Vogelius[2, 4]

1 – Department of Computer Science, University of Copenhagen
2 – Department of Oncology, Rigshospitalet, University of Copenhagen
3 – Department of Oncology, Herlev and Gentofte Hospital, Copenhagen
4 – Department of Health and Medical Sciences, University of Copenhagen
5 – Department of Health Technology, Technical University of Denmark
6 – Department of Health Research, SINTEF, Trondheim, Norway

Manuscript in preparation. An earlier version was presented as a digital poster at ESTRO which was published in Radiotherapy and Oncology [189].

Abstract

Background: Corrective annotation is an interactive-machine-learning method that reduces contouring time but correcting large contiguous error regions still involves repetitive work.
Purpose: To measure the change in contouring time when updating the segmentation in real time based on user annotations as input.
Methods: We compare the existing interactive-machine-learning method, RootPainter3D, to a new variant, named auto-complete, that allows segmentation for the current image being annotated to be continuously updated based on the annotation input.
Results: We find that auto-complete is slower initially and then appears to be faster after enough images have been annotated.
Conclusions: Our proposed auto-complete method illustrates the potential to

augment interactive-machine-learning methods with interactive-segmentation and, given enough data, to further accelerate auto-contouring of organs at risk in radiotherapy.

5.1 Introduction

There is a need for faster and more accurate auto-contouring to improve radiotherapy. A critical use-case is online-adaptive radiation therapy (ART) which may reduce harmful radiation dose to critical organs-art-risk [14] and is implemented at magnetic resonance linear accelerators (MR-linacs) [109, 242], and online-adaptive computerised tomography (CT) linacs [11, 186, 253]. In ART, the daily patient anatomy is taken into account in each radiotherapy treatment session, necessitating the rapid creation of accurate contours. At sites where anatomy changes rapidly due to bladder filling, rectum filling, or intestine movements, the accuracy of segmentations deteriorate with time. If online segmentation takes too long, the patient anatomy at the time of treatment will no longer be representative of the pre-treatment scan used for segmentation. Therefore, segmentation time is crucial for accurate treatment delivery in ART.

To improve annotation speed, several methods have been proposed that use preliminary annotation such as clicks or scribbles as input to a model to guide predictions. The class of annotation approaches that use user-input in the model inference stage is known as interactive-segmentation. Interactive-segmentation methods may take user input in the form of a bounding box [117], scribbles [15] or clicks at the extremities of a visible object [150]. The 3DSlicer [155] medical image annotation software supports interactive segmentation methods such as GrabCut [172]. From drawing scribbles in one or multiple 2D slices of the foreground and background classes, 3DSlicer can generate and refine an entire 3D segmentation. Imfusion Labelsis a commercial software using a GrabCut-like algorithm that has demonstrated annotation speed three times faster than a manual approach using ITK-SNAP [255] for segmentation of vestibular schwannoma in MRIs [131]. QuPath [16] enables interactive segmentation with a random forest pixel classifier. Given some initial annotations, QuPath trains a pixel classifier model, which can then be applied on the remainder of the image or new unseen images. MONAI Label [45] is used as an extension for existing annotation software such as 3DSlicer or QuPath and enables interactive segmentation both for radiology and histopathological images. Similarly to QuPath, MONAI label supports interactive segmentation through machine learning-based models. However, it extends QuPath by adding deep learning support. For ultrasound images, Annotation Web [188] provides a web interface and generates segmentations from user-defined border coordinates using cubic Hermite spline interpolation.

5.1.1 Interactive machine learning

Interactive machine learning for image segmentation was introduced by Fails and Olsen [51] in their system named Crayons which involves a corrective feedback cycle where model predictions are viewed by the human annotator, with the corrections being provided back to the model. As opposed to using user-input in the model inference procedure, interactive-machine-learning methods use user input in the training procedure to select which examples, or regions of images will be used in training. Interactive-machine-learning involves a human-in-the-loop during the model training procedure, as opposed to traditional machine learning, where models are trained from an existing annotated dataset in a fully automatic way. Thus, they are useful for applications where sufficient annotated data is not yet available and there is a desire to obtain additional annotation with the assistance of an already partially trained model.

Histo-Cloud [125] is a web-based human-in-the-loop software for histopathology images, utilising a Deeplab V3+ [33] CNN for segmentation. CVAT [180] is another open, web-based annotation tool, primarily tailored and used for natural images. CVAT is free to use and can be set up locally, similar to Annotation Web. However, CVAT offers a paid subscription to annotate data on the cloud, with GPUs, where various semi-automatic tools are available. To speed up annotation, CVAT offers various pretrained models to assist in annotation, both pure detectors and interactors, similarly to GrabCut.

5.1.2 Corrective annotation for interactive machine learning

Corrective-annotation methods have been shown to continually improve segmentation efficiency [196] and consistency [40] for routine delineation tasks. Correcting auto-generated contours may still be tedious as error regions often span multiple axial slices (Figure 5.1), requiring similar manual corrections to each slice. We propose to use the ongoing corrective annotation of the current scan as input to the model to refine predictions in subsequent slices, generating real time auto-contour updates as delineation progresses.

To the best of our knowledge, no prior work has evaluated the annotation efficiency of interactive-segmentation combined with interactive-machine-learning applied to 3D medical image data.

We evaluate the performance of our contour auto-complete method by comparing it to a baseline method where all corrections are assigned manually without updates to the model prediction during delineation (full-correction). We hypothesise that the auto-complete method will result in reductions in delineation time in comparison to the baseline method whilst maintaining high accuracy.

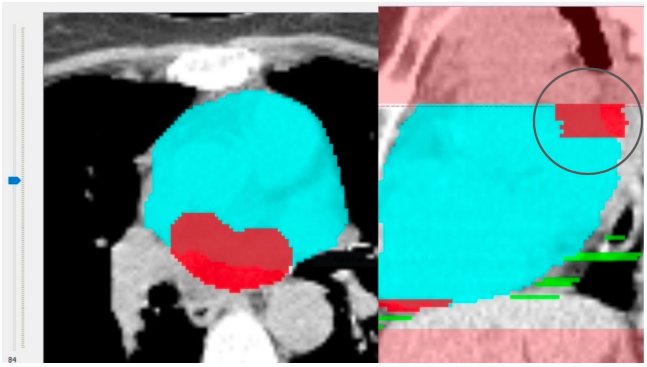


Figure 5.1: Heart scan shown with segmentation overlaid in blue in the RootPainter3D software. False negative annotations are shown in red. The grey circle indicates a contiguous error region where several similar annotations have been assigned to multiple adjacent slices.

5.2 Methods

Both corrective annotation methods (auto-complete and full-correction) were implemented as variations of the open-source RootPainter3D [196] deep learning auto-contouring software, which uses corrective-annotation in training to progressively improve auto-contouring accuracy.

We used a dataset of MRI scans which included multiple scans from 31 different patients with liver metastases that had been referred to SBRT. 177 scans were delineated in the same order using both methods. 12 images from these 177 images were also delineated by a trained clinician using the MRIdian planning system to enable consistency to be checked between the completed corrective delineations and standard clinical delineations.

To mitigate the so-called cold-start problem [120] which hampers the utility of IML methods on new datasets without an initial trained model [209], we trained an initial model to convergence on 24 dense clinician delineations, with 18 images used in a training set and 6 in a validation set used for early stopping.

The corrective delineation procedure consisted of three annotation sessions, with models left training overnight, on the newly expanded dataset of annotations, between each session. The sessions included 20, 60, and 97 images, respectively. Delineation time was automatically recorded, including time to both review and correct identified mistakes.

5.3 Results

We report time to delineate each scan as a function of the number of images delineated in figure 5.2. We found auto-complete started out slower than full-correction, but appears to be faster as more images are delineated (Figure 5.2).

For auto-complete, the delineations annotated in the third annotation session took significantly less time than the first annotation session. Although auto-complete appears faster on average, the difference between methods was not yet significant. We also report the agreement with expert clinician delineations (Figure 5.3).

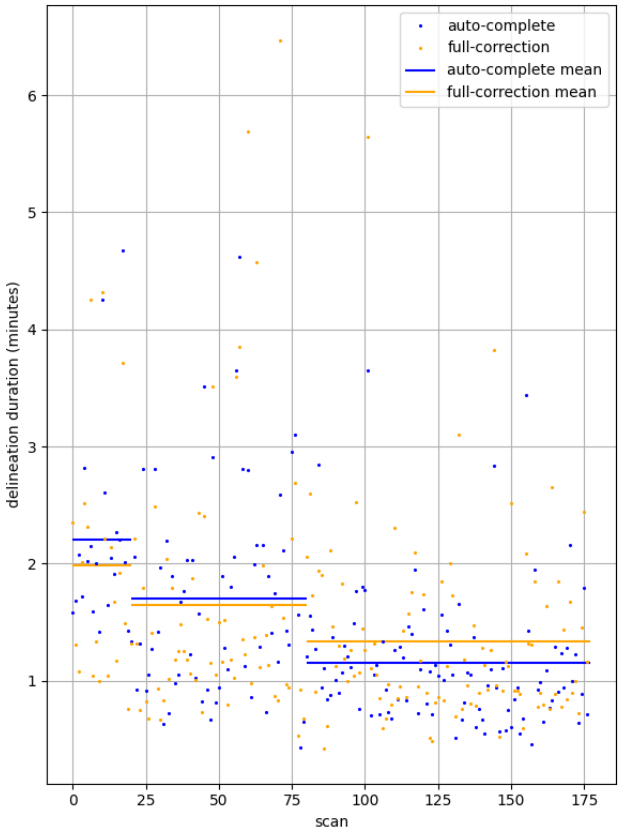


Figure 5.2: Delineation duration for each scan in order of completion time for both the auto-complete and full-correction methods.

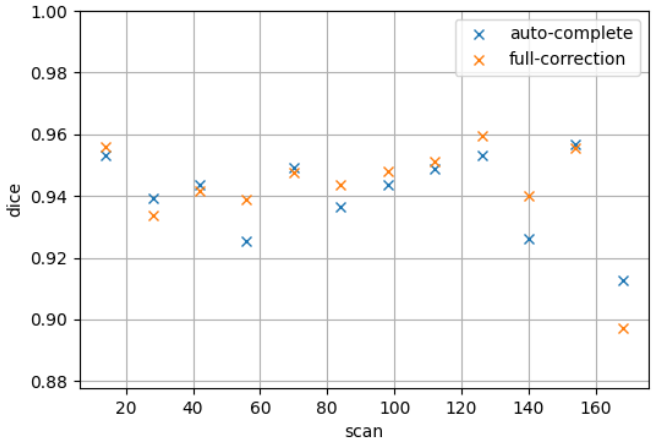


Figure 5.3: The dice with expert clinician delineations for the completed contours for both methods was similar to inter-annotator variation.

5.4 Discussion and Conclusion

During the first annotation session, the auto-complete method hampered contouring performance, as poor contour auto-completions increased the delineation workload but with sufficient training data and time, corrective annotation with auto-completion provided some reductions in contouring time, whilst maintaining high delineation accuracy. Overall both corrective-annotation methods provided rapid contouring (Figure 5.2). For comparison, manual liver delineation has been reported as taking 4-8 minutes. me. we need timings. As shown in Figure 5.3, when measuring agreement with expert clinical delineations we found an agreement between contours which was similar to inter-annotator variation (~ 0.94), indicating high accuracy and suitability for clinical use.

Discussion

6.1 2D RootPainter

The 2D RootPainter study (Chapter 2) was conducted to address the need for a more efficient and easy to use method for biologists to train their own segmentation models. IML is of broader relevance to the development of deep learning applications as it combines training and annotation in an interactive process, providing real time feedback to the annotator (Figure 3.1). To our knowledge, there were no prior methods or existing software applications that facilitate IML with a U-Net segmentation architecture.

We found corrective-annotation with the proposed RootPainter IML software to provide rapid and accessible model training for biological image segmentation. Demonstrating the versatility and accessibility of our approach, we trained segmentation models for three different structures of interest from three diverse datasets, with annotation, model training and segmentation all completed using a purpose-built graphical user interface. The accuracy of the obtained models in the limited time period demonstrates the efficiency and utility of the proposed method as a complete and accessible machine-learning pipeline for biologists.

6.2 Efficient IML model training

In chapter 2 we found the proposed IML method to be an efficient way to train segmentation models. For five out of the six models trained using corrective-annotation, the resulting measurements had a strong correlation with manual measurements within two hours of interactive training.

We also compared corrective to dense annotations in terms of their ability to train models to a given accuracy in a limited time period. We found improved performance with corrective annotation on average, but dense annotation performed surprisingly well.

Our results show that deep learning segmentation models can be trained to a high level of accuracy with annotations produced in a short period of time. Our results challenge the common misconception that deep learning methods

require vast amounts of data [65, 147]. More annotated data undoubtedly tends to improve generalisation, but the results presented in this thesis show that for some datasets and tasks in biological image analysis, useful models can be obtained with small amounts of annotation.

The difference between the corrective and dense annotation protocols studied in chapter 2 was not significant, which we expect was partially due to the restricted time for annotation. It appears that dense annotation performs well in the early stages of annotation as the value of corrections is small when the model is largely predicting incorrectly. Corrective-annotation includes an additional overhead of the user having to inspect model predictions in addition to inspecting the image. This overhead is described by [210] as the query cost. Our findings from both chapter 2 and 3 show that corrective-annotation comes into its own later in the interactive training procedure, once larger regions of the input are already predicted correctly, reducing the annotation burden and allowing the annotator to quickly move on to more problematic areas of the dataset for the model being trained. We had similar findings in our study on IML for segmentation in 3D (Chapter 3) where, in the early stages of interactive training, corrective annotation was out-performed by densely annotating in the eclipse treatment planning software (Figure 3.4). These findings from chapter 2 and 3 regarding the utility of dense and corrective annotation at various points in the training procedure informed the study design presented in chapter 5, where we opted to start training on an initial densely annotated dataset and only compared corrective-annotation procedures for subsequently annotated images. This challenge is related to the 'cold start' problem that also affects active learning [66] and is typically encountered in the literature regarding recommender systems [175, 230], where there may be a requirement to make predictions related to an entity for which there is little or no prior information. The challenges in this initial 'cold-start' period of annotation thay be alleviated by leveraging alternative annotation approaches for interactive segmentation methods, such as [150] or utilising pre-trained foundation models [107] which may reduce the user input required compared to more manual annotation approaches.

Subsequent studies have further validated the efficiency benefits brought about by our proposed segmentation methodology, an analysis using 36500 manually annotated images found RootPainter to make the phenotyping process over 50 times faster in comparison to a manual annotation method using RootFly [256], saving over 1000 hours per growth season [19].

Our findings, and the software implementations we have made publically available (Chapter 2) have helped highlight IML as a promising method to overcome the annotation requirement of deep-learning models [148]. The rapid training process has been reproduced for other datasets, with Han et al. [72] training all models in their study within 200 minutes of interactive training and Han et al. [73] using a limit of two hours for interactive training. RootPainter has also been used to further refine destructive sampling workflows, with Han et al. [72] showing how automated segmentation can be used to remove arduous manual debris cleaning steps from the phenotyping process.

Seethepalli et al. [179] were able to obtain satisfactory results using just 10

partially annotated images. For 3D X-ray CT data of soda-lime glass pellets, impressive results were obtained using just 50 annotated slices [5].

In chapter 2, we argued that corrective-annotation could be used to train accurate models in a short time period. Subsequent studies have used our methods to investigate the accuracy of models trained with corrective-annotation for various tasks. The measurements provided by RootPainter for root analysis have been shown to be highly positively correlated with root biomass data [3]. Clément et al. [36] performed validation on grid counts on 2000 images, obtaining an R-squared of 0.55. Alvarez-Borges et al. [6] observed that the mean error for porosity was below 1% and the mean error for methane gas measurements was below 0.05%. Although such evaluations of model accuracy contribute to a better understanding of the characteristics of our proposed methods for various tasks, the large scale manual validation associated with the models created in many studies using our methods, may also highlight a limitation of the corrective-annotation model training workflow, in that it is time consuming to perform additional dense annotation to validate a model trained with corrective annotation (See section 6.7.2 for a more detailed discussion).

6.3 RootPainter3D

We conducted the study presented in chapter 3 because organ-at-risk contouring is a time consuming task in radiotherapy. Although methods exist to train and use deep-learning segmentation models [95, 30], there were no studies evaluating IML with an integrated GUI for a routine radiotherapy contouring task. A method that enabled continuous training, allowing clinicians to update and train their own models would have several advantages in terms of flexibility and adaptability, allowing clinics to train their own models for use cases not anticipated by commercial software vendors. We implemented an interface enabling the navigation and annotation of 3D X-ray CT images (Figure 3.2 and 3.3), and a 3D convolutional neural network to better utilise the volumetric context (Section 3.3.2).

We evaluated our proposed method on a heart segmentation task and found it was able to converge to an accurate model, demonstrating the potential of IML software for use in routine organ-at-risk contouring. The accuracy of the model is demonstrated by the dice score of 0.945, evaluated against independently delineated heart contours using the eclipse treatment planning software. This dice score is high compared to other approaches, which we list in table 6.1 for comparison. We observed a contouring time of two minutes which is considerably faster than the more manual method used in the clinic (Section 3.4). We also found a small deviation in captured dose, with only four of the last 300 hearts having a deviation above one Gray (Figure 3.12), indicating the potential for use in dose-response studies.

Despite the high accuracy of the trained model, our study revealed challenges of using DL methods as in some cases the segmentation would fail catastrophically (Figure 3.9) indicating the risks of naively using AI models in a clinical

setting. Our observations also reinforce comments by Aznar and Marrazzo [14] that there is a continued requirement for expertise in the clinical treatment team when using AI methods to ensure supervision and correction of automatic delineations.

The focus of the RootPainter3D method is on the review and correction of segmentation errors, which lends itself to the ongoing efforts to improve the safety and standard of AI in the clinic [68]. Corrective annotation provides a way to continuously review and correct model errors but does not reveal the cause of errors such as those presented in Figure 3.9. Further insight into the visual features that cause segmentation errors may be possible with occlusion methods [65].

We used the heart contours produced in chapter 3 to investigate the relationship between mean heart dose and lymphopenia, showing the link between cardiac toxicity and immunosuppression for patients with intrathoracic tumours [212]

We used the liver contours produced in chapter 5 in a study investigating interfractional dose accumulation [236]. The annotations and RootPainter3D software created as part of chapter 3 were also used to train another model that has helped provide insights into the radiation exposure and association with observed cardiovascular toxicity in over 5000 patients [56].

Although the dice of 0.945, computed against a set of independently delineated images was high compared to other approaches, the corrective-dice of 0.991 was much higher. The corrective annotation protocol dictates that the user should only annotate regions of the predicted segmentation that they are sure are incorrect. This results in an annotation map consisting of foreground, background and unknown voxels/pixels. By default all pixels are unknown until annotated. For some of the pixels that remain unknown (not actively annotated with corrections) it may be that the annotator was satisfied with the model prediction but for others the annotator may not have been certain that the model prediction was a clear error due to noise, limitations in their own knowledge or lack of clarity in the corresponding region of the segmented image. These unknown pixels may not be captured in measures of error based on corrective annotation, leading to more optimistic indications of model performance.

We also expect that corrective-dice provides a more accurate measure of reducible clear errors. By clear error, we mean errors that, when inspecting a segmentation, annotators would agree on and by reducible we mean that could potentially be reduced by further training of the model. Examples of errors that are likely not reducible and not clear are those that will be included when comparing two independently created dense annotations that disagree on a noisy or ambiguous boundary region. Upon review it may be declared that neither dense annotation was more accurate on the boundary, but that the true boundary was ambiguous. Such regions of ambiguity in the images may lead to a drop in dice score when comparing two independent dense annotations but not necessarily when quantifying error using corrective annotation. This is because with corrective annotation the annotator is only instructed to label the regions of the image where they are certain the model has made an error. With

the dense annotation used in chapter 3 the annotator had to label each voxel as foreground or background and did not have the option to label a voxel as unknown, which may have provided a similar dice to corrective annotation.

The corrective-dice is not independent as corrective annotation is assigned in response to the segmentation and after the user has observed the model prediction. We did not investigate the extent to which observing model predictions may have biassed the annotator, which is a possibility and a more general concern when using corrective-annotation procedures for contouring [241]. We suspect this bias may in some cases also lead to advantages, as a model's predictions may highlight regions of an image that would have otherwise been missed, or by encouraging consistency.

We used a bounding box in the initial RootPainter3D study presented in chapter 3 but not in the study presented in chapter 5, in order to save time. Specifying a 3D bounding box accurately took approximately 30 seconds, which was approximately 25% of the time spent on each scan towards the end of interactive training. Using a bounding box likely provided benefits in terms of training time and stability, as shown in chapter 4. Our method for specifying the box may have been made more efficient by extreme point clicking [150] or by having boxes be automatically proposed that are verified by annotators [149].

6.4 Versatility and Accessibility

The 2D RootPainter software and proposed IML image processing pipeline presented in chapter 2 has found utility for the segmentation of a broad array of datasets and image processing tasks. These datasets include plant species in varying backgrounds and heterogeneous soils taken from minirhizotrons with RGB cameras [36, 19], multispectral cameras [130], soil-cores and profile walls [72], rhizobox facilities [181, 3, 2, 32] and the analysis of biopores in three new datasets [73].

RootPainter has also been applied successfully in more controlled phenotyping environments. It has been highlighted as an effective way to measure nodule mass per plant for legumes [44], applied to create segmentations of nodules in growth pouches [43, 138][1], and scanned roots [72], streamlining destructive phenotyping measurements.

Further validating our claims (Chapter 2) regarding flexibility and versatility, applications have been found outside of 2D biological imaging, with studies using RootPainter for granular packings and gas bubbles in X-ray CT [5, 128] and to analyse fibres [102] and voids [225] in composite materials.

One of our key aims in the study presented in chapter 2 was to make deep-learning model training more accessible to biologists without a machine learning background. To enable this we implemented a IML U-Net segmentation model training system utilising a client-server architecture, with all operations made

[1]The detailed growth and image capture setup and discussion of how our IML analysis pipeline has influenced nodule research is described in an online video created by R Ford Denison: Darwinian Agriculture [158].

available via a purpose-built cross-platform user interface. The interface was used in the study to perform all actions required for both training and using the segmentation models to complete the three different image processing tasks discussed.

Our claims regarding the accessibility of the method (Chapter 2) have been further emphasised by external research groups who have evaluated RootPainter for new tasks and datasets [181], with the approach described as straightforward to implement [5] and an easy way to train a CNN [42]. In almost all of the studies where RootPainter has been applied, models were trained interactively by scientists without a background in computer science or machine learning.

Making deep-learning segmentation model training accessible to biologists is a novel feature of RootPainter that has enabled new research questions. For example, Sell et al. [181] trained different models to analyse the same images in different ways, quantifying different regions of the root anatomy. This study specific use case was not considered in the design of the original software and would not be possible using a more specific root segmentation approach that did not facilitate model retraining by end users. RootPainter has been chosen by biologists looking to develop affordable and easy to use phenotyping platforms incorporating low cost or commodity hardware [181, 142].

RootPainter has been highlighted as an established method for CNN model training [142] that has enabled the standardisation and increased speed of image analysis tasks [18]. In 2021, RootPainter was found to be the 5th most popular root image analysis software (out of a total of 52) from a survey conducted at the 11th symposium of the International Society of Root Research [42], with none of the more popular systems providing equivalent functionality to enable segmentation from noisy complex images.

The popularity of RootPainter is still growing rapidly. The installer for the GUI client has been downloaded 1551 times [2]. The training datasets [199] used in the original paper [195] and discussed in chapter 2 have been downloaded 816 times[3]. These statistics indicate we have made progress towards achieving a stated aim of chapter 2, which was to make deep-learning model training more accessible.

Studies such as Alvarez-Borges et al. [5] were completed by scientists able to attend online RootPainter workshops[4]. These workshops have been possible to run due to the data in the original study being made available online [199], which is a strength of using plant data, as opposed to medical images. Plant images can be more easily shared to allow others to reproduce results and learn to use the proposed methods.

[2] Calculated using `https://tooomm.github.io/github-release-stats/`. Accessed on the 20th of May 2023

[3] Download statistics taken from `https://zenodo.org/record/3754046`. Accessed on the 20th of May 2023

[4] Examples of three such workshops are
`https://nordicphenotyping.org/activities/annual-nppn-workshop-2022`
`https://www.microscopy.lu.se/sites/microscopy.lu.se/files/programmemlformicroscopy_october2020_final11.pdf`
`https://www.youtube.com/watch?v=73u73tBvRO4`

The accessibility of the software has enabled biologists to ask methodological machine learning research questions with implications outside of their target domain. Han et al. [73] experimented with combining different types of datasets in the training process and found this led to improved results with less annotation.

RootPainter has enabled new insights to be discovered in existing datasets. It revealed insights on the effects of deep tillage lasting up to 50 years that were not previously discovered when using manual annotation [73]. Bauer et al. [19] used RootPainter to highlight that automated segmentation better reveals root senescence, a physiological event related to root ageing that was not easily observed with previous methods. And Chen et al. [32] used RootPainter to provide insights into the performance of deep roots under varying water and nitrogen availability. The benefits of using RootPainter in terms of revealing new biological phenomena are attributed in part to a reduction in inter-observer variation [73] and elimination of annotator specific bias [19].

6.5 Localise To Segment

The RootPainter3D paper presented in chapter 3 had a localisation stage built in. This was a manual step involving the user first specifying a bounding box around the region of interest for each image delineated. We added this step to avoid the user having to inspect errors further away from the structure of interest. This method appeared effective, but includes manual overhead. We conducted the study presented in chapter 4 to investigate the potential performance benefits of using an automated localisation stage. Although popular in the literature we could not find a direct comparison to validate the effectiveness of such a strategy. We tested the benefits of localisation on several structures. A downside of such approaches is their additional complexity. We make our own implementations available open-source to help other research groups looking to implement two-stage localisation pipelines.

Localisation appears to provide substantial advantages in terms of accuracy for most of the structures and in terms of training time for all structures investigated, which is of particular relevance in an IML setting. We suspect the benefits of localisation are due to how the approach helps mitigate issues with class balance and allows different networks to specialise on details at different scales.

6.6 Image Analysis Software Complementarity

Although RootPainter's lack of detailed trait extraction has been identified as a weakness [19], several studies [19, 248, 3] have analysed their RootPainter segmentations in the open-source root analysis software RhizoVision Explorer [179]. This combination of methods has been highlighted as the current standard for reliable high-throughput root phenotyping [19, 254, 159, 42]. To further streamline integration we have developed functionality in RootPainter to provide

improved compatibility with RhizoVision Explorer (figure 6.1) which has been applied in recent studies [248].

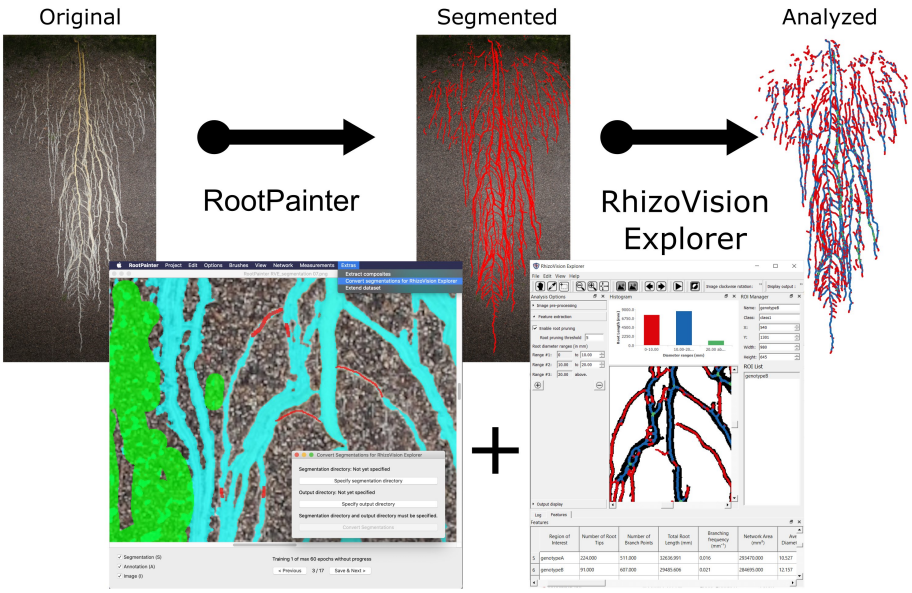


Figure 6.1: Illustration of a combined image analysis pipeline including RootPainter and Rhizovision explorer. The figure is supplied by Abraham George Smith (University of Copenhagen, Denmark) and Larry York (Oak Ridge National Laboratory, USA) and includes a rhizobox photograph from Benjamin Delory (Leuphana University of Lüneburg, Germany). The segmented root system displayed is of Lopholaena deltombei.

This complementarity between open-source image analysis software programs has created a division-of-labour enabling RootPainter to be developed as a general purpose segmentation technology, whilst RhizoVision explorer handles the more root specific trait extraction needs of the plant phenotyping community.

6.7 Remaining challenges

6.7.1 Dataset design, image order and patch size

In section 2.3.1 we describe our proposed *Create training dataset* functionality that splits each image in a dataset into a series of tiles to improve the efficiency of interactive training. This has not been used in every published study using RootPainter, for example Nair et al. [142] found that small patches did not always allow the annotator to view the necessary context to accurately annotate an image, thus they opted to annotate images with the original size of 2292 ×

1944, which we expect may have reduced the annotation efficiency. This limitation has been addressed in more recent iterations of the software by providing functionality to view images in context (Figure 6.2). But this function does not overcome limitations related to the fixed field of view of the model which, with some datasets, may not provide the most appropriate receptive field to segment the structure of interest.

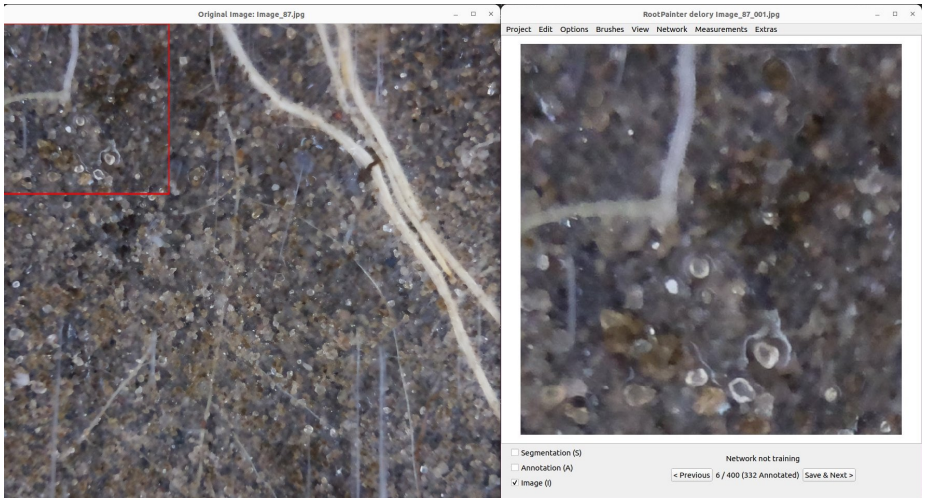


Figure 6.2: RootPainter 'view image in context' functionality allows users to inspect the original image from which a tile was taken, providing additional context to aid in annotating, otherwise ambiguous features.

In section 2.2 we motivated the need for a more user-friendly model training software addressing the existing challenges for biologists without an ML background. One challenge mentioned was dataset design.

We proposed a GUI based software combining deep-learning with corrective-annotation, which involves targeting annotation towards errors or weaknesses in the model during training (Section 2.9.3). The current implementation of both RootPainter and RootPainter3D present the images to the annotator in random order during the interactive training procedure. Our results show this approach is effective, resulting in a high accuracy in a short time period (Figure 2.7 & 2.8). However, a random image ordering, as implemented in the current versions of the software, is less optimal when training models for larger datasets, as the user becomes less likely to find the most crucial errors to be added to the training data as they progress through the images in a random order. This is because if a fixed period of time is available to annotate the images, the percentage of the dataset a user will be able to inspect and correct errors for will be smaller if the dataset is extremely large, meaning there is an increased likelihood that none of the images containing a feature of interest (for example an artefact that results in false positives) will be included in the annotated images. Thus we would not

expect a model to segment such images correctly. Although the user is able to increase the rate at which they annotate images as model performance improves (Figure 2.9(a)), the amount of pixels they annotate in each image decreases and there is still a fixed overhead for the inspection of each image viewed. For larger datasets, this inspection overhead can start to dominate the annotation time, further reducing the rate at which the user will be able to add new annotated pixels to the training data.

To compensate for this limitation in the software, subsequent works [19, 130] have taken additional steps to ensure models are suitable for the larger datasets. Bauer et al. [19] encourages users to actively curate their RootPainter training dataset to capture variation in relation to the factors present in the dataset, including facility and date of image capture. A limitation of such dataset design procedures is that users will typically not always know all factors of variation in their datasets. A randomization procedure will also randomise unknown factors of variation.

There have also been notable deviations from the corrective protocol. Malinowska et al. [130] trained a model to segment 78200 images in a few hours, skipping through many images to find the images with larger error due to artefacts and other abnormalities that they deemed more relevant to overall model performance. Such deviations from the protocol indicate challenges when using RootPainter to segment very large datasets with considerable outliers. Although additional user-instruction and protocol modification may be effective ways to address issues with larger datasets, these burden new users learning the software and introduce more possibility for user error. We believe the software can do more to optimally order the images for annotation. In future work we aim to address these challenges by improving the ordering of images using active learning to automatically suggest the next best image for the user to annotate to improve the overall performance of the model[5]. The complementarity between active learning and corrective-annotation has been recently demonstrated by [210] in a study using online learning with streaming sensor data, where a combination of IML and active learning provided improved performance in terms of both labelling and query cost.

6.7.2 The Validation Bottleneck

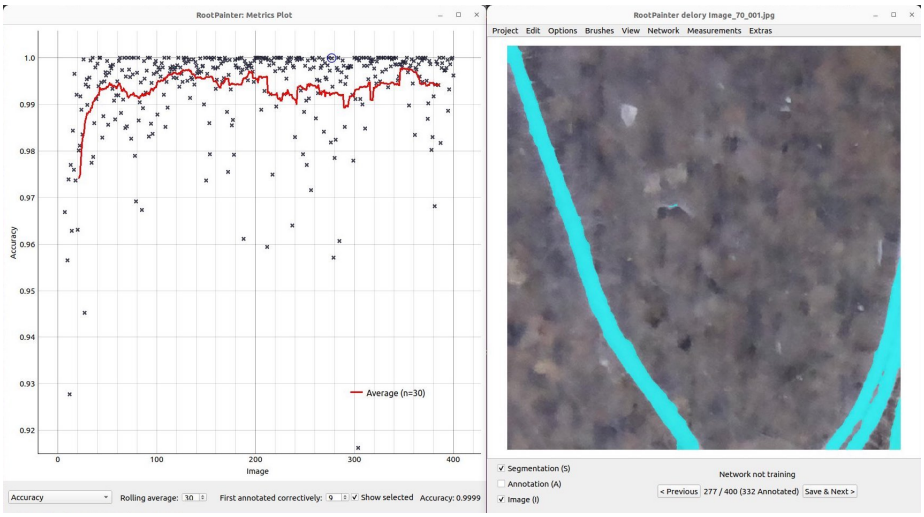
We claim in section 2.5 that the corrective-annotation protocol will enable annotators to be more informed about how much annotation is required to train a model to a suitable accuracy. Studies employing the proposed IML procedure have found the need to perform extra validation steps. The pipeline proposed by Nair et al. [142] includes a time consuming validation stage where the annotator first decides, based on a qualitative evaluation, if the model has stopped improving before performing a comparison to manually annotated data. Eval-

[5]Work package 3 of our postdoc project 'Interactive deep learning for image segmentation' which has already been awarded DKK 2662420 by the novo nordisk foundation and will be executed by assistant Professor Jens Petersen and Abraham George Smith at the University of Copenhagen over a 3 year period.

uating model performance visually in a qualitative way may not be consistent or easy for others to reproduce. Preparing additional manually annotated data for validation is typically time consuming and done using dense annotation, eliminating the benefits of using an IML procedure.

Based on our work evaluating the convergence of a heart segmentation model trained with IML (Figure 3.4) we have developed an interactive metrics plot (Figure 6.3(a)) to provide quantitative measures of model performance with the RootPainter software. The interactive metrics plot allows users to click on outliers in the plot to inspect specific examples and obtain rolling average measurements of error in terms of area, dice, accuracy, precision and recall. The metrics shown in the metrics plot are corrective-metrics, computed using the difference between the initial predicted segmentation and the corrected segmentation, which is the segmentation with the user corrections (annotations) assigned. Working with corrective-metrics could provide a way to eliminate the validation bottleneck as these are added as part of the model-training process and would not involve a time consuming additional manual annotation phase. Although the metrics plot helps, there is still a lack of clear guidelines for users to know what level of accuracy is appropriate, and thus how long to continue with their interactive model training when using an IML procedure. The use of corrective-metrics is also less established for model validation, and further research may be required to understand the strengths and weaknesses in comparison to metrics computed more conventionally using independent dense annotations.

(a)



(b)

Figure 6.3: Interactive metrics plot uses the corrected error to provide measures of model accuracy. Figure 6.3(a) includes an image from Jaeger et al. [97] and figure 6.3(b) includes an image from Alonso-Crespo et al. [4] which is a publically licensed dataset prepared for a RootPainter workshop.

6.7.3 Should we be correcting all errors?

In the RootPainter3D study (chapter 3), it can be seen in figure 3.12 that the dose did not change substantially for the vast majority of hearts when comparing the initial predicted segmentation to the corrected segmentation. The difference in dose is not just dependent on the magnitude of the error in seg-

mentation in terms of dice or incorrect voxels, but on the radiotherapy plan and the subsequent dose-gradient for the specific patient around their heart. This indicates that, for a segmentation model used to quantify dose, some segmentation errors of similar size are more important than others to correct. All of the hearts took a substantial amount of time to contour correctively (Figure 3.4), yet for only a few the corrections had a substantial impact on the dose (Figure 3.11). Automatically identifying such segmentations or regions of segmentations ahead of time would vastly reduce the necessary contouring time when producing contours to accurately quantify radiation dose within a given margin of error in terms of absolute dose.

In a clinical setting such as radiotherapy planning, it may not make sense for a dose-planner to correct organ-at-risk segmentation errors that are far away from the target and thus unlikely to lead to changes in the plan or radiation dose quantified by the contoured organ. In future work it may be possible to speed up corrective annotation by considering the predicted dose information. More could be done to guide clinicians towards corrections most critical for the end use case.

6.8 2D or 3D CNNs for 3D Image Segmentation?

An advantage of 2D CNNs, and likely one advantage of the RootPainter2D, in comparison to RootPainter3D is the faster model training time. Faster model training makes it easier to evaluate an annotation strategy for a new potential dataset or task. In chapter 2 the annotation duration for each experiment was limited to two hours, with accurate models obtained. In contrast to this, a limit of 2 hours for the annotation performed in chapter 3 would have resulted in less than 20 of the hearts being annotated and a performance that was not yet competitive with the baseline manual comparison method in terms of contouring time (Figure 3.4). The tasks evaluated in chapter 2 and chapter 3 are different and thus a direct comparison is not possible based on the works in this thesis but a considerable body of prior work exists to help inform an understanding of the tradeoffs between 2D and 3D networks for the segmentation of 3D structure

The use of networks with 3D convolutional kernels for medical image segmentation of 3D structures, as investigated in chapter 3 can be motivated by their capability of using interslice context, including spatial information that cannot be considered when using 2D convolutions on individual slices [234]. The use of individual slices would have been the only approach possible if applying the 2D system proposed in chapter 2 to the heart segmentation task investigated in chapter 3. Although there are seemingly intuitive benefits of 3D CNNs, it is not entirely clear these materialise in tangible performance advantages compared to their 2D counterparts. The use of 3D convolution comes with a computational trade-off.

In a study using a 240 CT scan dataset, model training time was found to be

twice as long for the 3D compared to a 2D approach, with all models taking over a day to train [235]. Another limitation with 3D CNNs highlighted by Lei et al. [116] is that whole volume training is increasingly computationally expensive and complex, especially as an increasing number of layers are used. These increased computational requirements, in particular GPU memory, require the use of smaller batches in training and smaller regions in the input patches with reduced input size, leading to slower training [234]. The number of voxels scales cubically with the region of interest, raising the computational burden to learn more complex visual patterns in comparison to 2D approaches [118]. Although Vu et al. [234] did not observe significant performance improvements for 3D CNNs regardless of dataset size in their investigation, they speculate that the typically higher number of parameters in 3D CNNs necessitates larger datasets to mitigate overfitting. Pre-trained 3D models and large labelled datasets for pre-training are also lacking in comparison to more established 2D approaches [249].

6.8.1 Accuracy Comparison

Table 6.1: A dice score comparison between 2D and 3D CNNs for various structures, with the higher accuracy shown in bold.
.

Reference	Structure	2D	3D
[224] (MRRN)	Left lung	0.97	**0.98**
[224] (MRRN)	Right lung	0.965	**0.97**
[224] (MRRN)	Heart	**0.92**	0.91
[224] (MRRN)	Oesophagus	**0.74**	0.64
[224] (MRRN)	Spinal cord	**0.87**	0.71
[228] (FCN)	Oesophagus	**0.835**	0.82
[228] (FCN)	Heart	**0.95**	0.94
[228] (FCN)	Trachea	0.90	0.90
[228] (FCN)	Aorta	**0.93**	0.92
[235] (U-Net)	Brain (Brats19)	0.768	**0.769**
[235] (U-Net)	Kidney (KiTS19)	**0.766**	0.763
[235] (U-Net)	Brain (IBSR 18)	0.898	**0.924**
[235] (U-Net)	Head and Neck	**0.656**	0.635
[235] (U-Net)	Pelvic (U-PRO)	0.807	**0.841**
[235] (U-Net)	Heart	0.857	**0.899**
[235] (U-Net)	Spleen	0.738	**0.828**
[235] (U-Net)	Hippocampus	0.830	**0.831**
[235] (Seg-Net)	Brain (Brats19)	**0.744**	0.726
[235] (Seg-Net)	Kidney (KiTS19)	**0.755**	0.735
[235] (Seg-Net)	Brain (IBSR 18)	**0.782**	0.744
[235] (Seg-Net)	Head and Neck	**0.643**	0.599
[235] (Seg-Net)	Pelvic (U-PRO)	0.763	**0.771**
[235] (Seg-Net)	Heart	**0.804**	0.770
[235] (Seg-Net)	Spleen	**0.691**	0.582
[235] (Seg-Net)	Hippocampus	**0.803**	0.786
[94] (U-Net)	Brain Edema [187]	0.786	**0.8071**
[94] (U-Net)	Non-enhancing brain tumour [187]	0.5865	**0.6222**
[94] (U-Net)	Enhancing brain tumour [187]	0.7742	**0.7907**
[94] (U-Net)	Heart [187]	0.9136	**0.9245**
[94] (U-Net)	Liver [187]	**0.9337**	0.5394
[94] (U-Net)	Liver tumour [187]	**0.9411**	0.6174
[94] (U-Net)	Hippocampus Anterior [187]	0.8852	**0.8987**
[94] (U-Net)	Hippocampus Posterior [187]	0.8670	**0.8820**
[94] (U-Net)	Prostate Peripheral Zone [187]	0.6198	**0.8431**
[94] (U-Net)	Prostate Transitional Zone [187]	0.6077	**0.8373**
[94] (U-Net)	Lung tumour [187]	0.5268	**0.5587**
[94] (U-Net)	Pancreas [187]	**0.7470**	0.3541
[94] (U-Net)	Pancreas tumour [187]	**0.7769**	0.4269
[93] (U-Net)	L. ventricular cavity	**0.945**	0.928
[93] (U-Net)	R. ventricular cavity	**0.902**	0.879
[93] (U-Net)	L. ventricular myocardium	**0.905**	0.872
[257] (U-Net)	Liver	**0.94**	0.93
[257] (U-Net)	R. kidney	**0.91**	0.89
[257] (U-Net)	Spleen	**0.93**	0.92
[257] (U-Net)	Pancreas	0.57	**0.59**
[196] (U-Net) (RootPainter3D)	Heart		0.945

Zettler and Mastmeyer [257] indicate significant advantages for the 2D U-Net for the kidney and liver and substantial savings when using the 2D network in terms of computation for all organs investigated.

In some cases 3D CNNs have been found to provide improved segmentation accuracy in comparison to 2D CNNs, in particular for organs with a small volume or having a tubular shape [261]. But, as shown in table 6.1, which includes a collection of comparisons from 2D and 3D from various studies in the medical image domain, the benefits of 3D are inconsistent. The 2D network presented in chapter 2 of this thesis, used as part of the proposed RootPainter IML segmentation system was found to have a higher IoU when compared to two alternative networks, including a network using 3D convolution [6]. Similar to many other studies, results from van Harten et al. [228], show only minor differences in accuracy between the 2D and 3D CNNs investigated for segmentation of oesophagus, heart, trachea and aorta.

6.8.2 Multiple adjacent slices

To alleviate issues with training time and computational resources associated with 3D networks, a variety of pseudo 3D, or 2.5D approaches have been proposed. Multislice inputs, as opposed to 3D convolution, are described as a type of pseudo 3D Network that may provide ways to capture more 3D context with a reduced computational burden in comparison to networks that perform convolution in 3D. Vu et al. [234] compared multislice inputs to 2D CNNs on five datasets, finding that the benefits in accuracy were not significant for four out of five of the datasets investigated. Increasing the number of input slices in the multislice network did not appear to provide additional benefit in segmentation accuracy in comparison to a 2D segmentation approach [234].

6.8.3 Multiple orthogonal slices

In attempting to bridge the gap between 2D and 3D Networks in terms of computational performance and accuracy, Yang et al. [249] propose an approach they named ACS (axial-coronal-sagittal) convolutions that can capture 3D context without processing the full 3D volume. Roth et al. [171] propose a 2.5D representation for lymph node detection using random sets of deep CNN observations. During training their approach samples the 3D input volume from a variety of orthogonal views, varying in scale, translation and rotation.

Roth et al. [171] argues that 3D CNNs are not possible to implement efficiently due to hardware constraints. Although there have been substantial hardware advancements in the several years since this claim was made, more recent similar multi-view 2.5D sampling approaches [154] have emerged competitive with 3D networks for a variety of 3D segmentation tasks in the medical segmentation decathlon [10].

6.8.4 Regularisation of 3D CNNs

Vu et al. [234] argued that 3D networks have a tendency to overfit and thus more data is required. A variety of regularisation methods have been employed to mitigate the overfitting of 3D CNNs. The aforementioned 2.5D approach [154] was outperformed by a purely 3D method [140] using a novel regularisation approach in the medical segmentation decathlon [10].

Myronenko [140] utilised a variational auto-encoder (VAE) to regularise a shared encoder. Myronenko [140] claim this VAE encoder regularisation both improved the accuracy and reliability of the training procedure, ensuring that the network performance was less sensitive to the initial random weights used, with each training run leading to a model with good performance.

Yang et al. [249] discuss methods for improving the performance of 3D medical image segmentation including deep supervision. One example of deep supervision is outlined by Dou et al. [46], who propose an approach where lower-level and middle-level features from their CNN encoder are upsampled using deconvolution. They then use these additional prediction paths to compute segmentations and then compute the loss via the difference between each of the generated segmentations and the ground truth. Such deeply supervised approaches are believed to provide more effective parameter updates, mitigating issues with vanishing gradients and have been found to result in faster training and better performance for heart and liver segmentation tasks [46].

UNet++[262] employs deep supervision via inclusion of skip connections to reduce the 'semantic gap' between feature maps in the decoder and encoder sections of the network to make the learning task easier. Zhou et al. [262] find their modified U-Net provides an improvement in IoU of 3.9 over a regular U-Net without deep supervision.

As a method that could speed up training, deep supervision is relevant to the IML approaches discussed in this thesis. Faster training will likely provide an improved user experience by reducing the time for the model to adapt to annotations provided by the user, further reducing the amount of annotation needed. Training stability is also of interest, as a deep-learning training process that fails is particularly complex for non-expert users to debug.

6.8.5 Ensembling both 2D and 3D networks

Given that 2D and 3D CNNs likely make different types of errors, combining an ensemble of 3D and 2D U-Nets may prove to be a more optimal approach compared to using either in isolation. This approach was taken by nnU-Net[94], which won the medical segmentation decathlon using an ensemble of both 2D and 3D U-Nets.

Similarly van Harten et al. [228] compare both 2D CNNs, 3D CNNs and an ensemble combining both methods. Although there are minor differences between approaches they are likely negligible when considering the additional complexity of combining 2D and 3D methods and the additional hardware requirements and training cost of training a 3D network. Of the approaches

compared by van Harten et al. [228], the 2D network used 73 thousand parameters, a fraction of the 3.7 million used for their 3D network, which brings into doubt the fairness of the comparison. A fairer comparison would allocate similar resources to both architectures. The extra parameters could have been utilised in the 2D network to enable a deeper network with more channels or layers, a larger receptive field or even an ensemble of 2D networks taking similar resources to the single 3D network.

6.8.6 Issues specific to using 3D CNNs with Interactive Machine Learning

One issue to consider is how the user navigates through data when providing annotation to create a model using IML within a limited time. When annotating 3D images sequentially (as opposed to 2D images) and completely, there is likely a reduced rate at which variance is captured in the annotated portion of the data. It may be the case that annotating similar adjacent slices will result in an annotated dataset that includes less variation than a dataset including slices that are more varied from regions further away in the same patient or taken from many different patients.

Such annotation procedures that are designed to maximise variation may pose challenging in practice as accurate annotation of a single slice may involve navigation and inspection of the surrounding 3D region. It may also be necessary to annotate each patient completely and sequentially due to clinical workflow requirements.

There are also considerable challenges in utilising sparse 3D annotations in training in comparison to sparse 2D annotations. Assuming a single slice is corrected, in the 3D context this drastically increases the amount of information that is provided as input to the network. 3D volumes are inherently much bigger and more memory intensive, with the input increasing cubically with the size of the region of interest. This poses limitations in the training efficiency of 3D IML methods. Training time itself will also slow the feedback loop between annotator input and model adaptation.

6.8.7 New possibilities when processing data in 3D

In chapter 5 we explored a solution not possible if working in a slice-by-slice way. As 3D volumes are segmented in a single pass, sparse annotation can be used to guide the network inference and assist in filling in the blanks in the missing slices between the slices annotated. This approach combining IML with interactive-segmentation was found to further improve contouring efficiency in comparison to an IML approach involving fully manual correction. (Chapter 5).

6.9 Auto-complete

We conducted the auto-complete study because, in chapter 3 we found that corrective-annotation has limitations when applied to 3D due to the user having to navigate through many adjacent slices, correcting similar model errors repetitively.

We hypothesised that if we allowed inference to repeat, but with the users work-in-progress annotation as input, then the network would be able to learn to utilise the annotation to refine the segmentation, providing speed-ups in the overall annotation process. This could potentially provide a way for the network to learn to interpolate between annotated slices or to extrapolate from the region of the structure corrected so far.

We found the proposed auto-complete method to be slower initially, but then it appeared, after a substantial period of interactive training, to become competitive and eventually faster than interactive training, confirming our hypothesis.

The auto-complete method indicates the type of advancements possible when integrating 3D context in a method combining aspects of both IML and interactive segmentation that would not be possible if processing the input data in a slice-by-slice manner.

6.10 Conclusion

The flexibility offered by the IML systems presented in this thesis has resulted in an expansion in the applications and use of novel computer science AI methods. Our findings in this thesis shine light on a path forward to enable more accessible and flexible deep-learning model training for a plethora of biological image analysis tasks.

Bibliography

[1] Azadeh Abravan, Corinne Faivre-Finn, Jason Kennedy, Alan McWilliam, and Marcel van Herk. Radiotherapy-Related Lymphopenia Affects Overall Survival in Patients With Lung Cancer. *Journal of Thoracic Oncology: Official Publication of the International Association for the Study of Lung Cancer*, 15(10):1624–1635, October 2020. ISSN 1556-1380. doi: 10.1016/j.jtho.2020.06.008.

[2] Kyriaki Adelais Boulata. Root growth in different soil depths and sap flow of barley (Hordeum vulgare L.) and pea (Pisum sativum L.) under drought. Master's thesis, University of Copenhagen, November 2020.

[3] Inés M. Alonso-Crespo, Emanuela W. A. Weidlich, Vicky M. Temperton, and Benjamin M. Delory. Assembly history modulates vertical root distribution in a grassland experiment. *Oikos*, n/a(n/a):e08886. ISSN 1600-0706. doi: 10.1111/oik.08886.

[4] Inés M. Alonso-Crespo, Vicky M. Temperton, and Benjamin M. Delory. Minirhizotron images for RootPainter demo, October 2022.

[5] Fernando Alvarez-Borges, Sharif Ahmed, and Robert C. Atwood. On Acquisition Parameters and Processing Techniques for Interparticle Contact Detection in Granular Packings Using Synchrotron Computed Tomography. *Journal of Imaging*, 8(5):135, May 2022. ISSN 2313-433X. doi: 10.3390/jimaging8050135.

[6] Fernando J. Alvarez-Borges, Oliver N. F. King, Bangalore N. Madhusudhan, Thomas Connolley, Mark Basham, and Sharif I. Ahmed. Comparison of Methods to Segment Variable-Contrast XCT Images of Methane-Bearing Sand Using U-Nets Trained on Single Dataset Sub-Volumes. *Methane*, 2(1):1–23, March 2023. ISSN 2674-0389. doi: 10.3390/methane2010001.

[7] Fernando Jesus Alvarez-Borges, Oliver N. F. King, B.N Madhusudhan, Thomas Connolley, Mark Basham, and Sharif I. Ahmed. U-Net Segmentation Methods for Variable-Contrast XCT Images of Methane-Bearing Sand. Preprint, Soil Science, April 2021.

[8] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. Power to the People: The Role of Humans in Interactive Machine Learning. *AI Magazine*, 35(4):105–120, December 2014. ISSN 2371-9621, 0738-4602. doi: 10.1609/aimag.v35i4.2513.

[9] Lasse F. Wolff Anthony, Benjamin Kanding, and Raghavendra Selvan. Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models. 2020. doi: 10.48550/ARXIV.2007.03051.

[10] Michela Antonelli, Annika Reinke, Spyridon Bakas, Keyvan Farahani, AnnetteKopp-Schneider, Bennett A. Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M. Summers, Bram van Ginneken, Michel Bilello, Patrick Bilic, Patrick F. Christ, Richard K. G. Do, Marc J. Gollub, Stephan H. Heckers, Henkjan Huisman, William R. Jarnagin, Maureen K. McHugo, Sandy Napel, Jennifer S. Goli Pernicka, Kawal Rhode, Catalina Tobon-Gomez, Eugene Vorontsov, Henkjan Huisman, James A. Meakin, Sebastien Ourselin, Manuel Wiesenfarth, Pablo Arbelaez, Byeonguk Bae, Sihong Chen, Laura Daza, Jianjiang Feng, Baochun He, Fabian Isensee, Yuanfeng Ji, Fucang Jia, Namkug Kim, Ildoo Kim, Dorit Merhof, Akshay Pai, Beomhee Park, Mathias Perslev, Ramin Rezaiifar, Oliver Rippel, Ignacio Sarasua, Wei Shen, Jaemin Son, Christian Wachinger, Liansheng Wang, Yan Wang, Yingda Xia, Daguang Xu, Zhanwei Xu, Yefeng Zheng, Amber L. Simpson, Lena Maier-Hein, and M. Jorge Cardoso. The Medical Segmentation Decathlon. *Nature Communications*, 13(1):4128, July 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-30695-9.

[11] Yves Archambault, Christopher Boylan, Drew Bullock, Tomasz Morgas, Jarkko Peltola, Emmi Ruokokoski, Angelo Genghi, Benjamin Haas, Pauli Suhonen, and Stephen Thompson. MAKING ON-LINE ADAPTIVE RADIOTHERAPY POSSIBLE USING ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING FOR EFFICIENT DAILY REPLANNING. page 10.

[12] Hassan Ashraf, Asim Waris, Muhammad Fazeel Ghafoor, Syed Omer Gilani, and Imran Khan Niazi. Melanoma segmentation using deep learning with test-time augmentations and conditional random fields. *Scientific Reports*, 12(1):3948, March 2022. ISSN 2045-2322. doi: 10.1038/s41598-022-07885-y.

[13] A. Atiya and Chuanyi Ji. How initial conditions affect generalization performance in large networks. *IEEE Transactions on Neural Networks*, 8(2):448–451, March 1997. ISSN 1941-0093. doi: 10.1109/72.557701.

[14] Marianne Camille Aznar and Livia Marrazzo. Techniques to Reduce Dose to Organs at Risk. In Orit Kaidar-Person, Icro Meattini, and Philip Poortmans, editors, *Breast Cancer Radiation Therapy: A Practical Guide for Technical Applications*, pages 287–295. Springer International Publishing, Cham, 2022. ISBN 978-3-030-91170-6. doi: 10.1007/978-3-030-91170-6_38.

[15] Xue Bai and Guillermo Sapiro. A Geodesic Framework for Fast Interactive Image and Video Segmentation and Matting. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, October 2007. doi: 10.1109/ICCV.2007.4408931.

[16] Peter Bankhead, Maurice Loughrey, Jose Fernandez, Yvonne Dombrowski, Darragh Mcart, Philip Dunne, Stephen Mcquaid, Ronan Gray, Liam Murray, Helen Coleman, Jacqueline James, Manuel Salto-Tellez, and Peter Hamilton. QuPath: Open source software for digital pathology image analysis. *Scientific Reports*, 7, 12 2017. doi: 10.1038/s41598-017-17204-5.

[17] Lara Barazzuol, Rob P. Coppes, and Peter van Luijk. Prevention and treatment of radiotherapy-induced side effects. *Molecular Oncology*, 14 (7):1538–1554, 2020. ISSN 1878-0261. doi: 10.1002/1878-0261.12750.

[18] Caroline Baudson, Benjamin M. Delory, Patrick du Jardin, and Pierre Delaplace. Triggering root system plasticity in a changing environment with bacterial bioinoculants – Focus on plant P nutrition. *Plant and Soil*, December 2022. ISSN 1573-5036. doi: 10.1007/s11104-022-05809-3.

[19] Felix Maximilian Bauer, Lena Lärm, Shehan Morandage, Guillaume Lobet, Jan Vanderborght, Harry Vereecken, and Andrea Schnepf. Development and Validation of a Deep Learning Based Automated Minirhizotron Image Analysis Pipeline. *Plant Phenomics*, 2022, May 2022. doi: 10.34133/2022/9758532.

[20] Arnaud Benard and Michael Gygli. Interactive Video Object Segmentation in the Wild. *arXiv:1801.00269 [cs]*, December 2017.

[21] Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. Large-scale interactive object segmentation with human annotators. *arXiv:1903.10830 [cs]*, March 2019.

[22] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. *arXiv:1206.5533 [cs]*, September 2012.

[23] David Bouget, André Pedersen, Sayied Abdol Mohieb Hosainey, Ole Solheim, and Ingerid Reinertsen. Meningioma Segmentation in T1-Weighted MRI Leveraging Global Context and Attention Mechanisms. *Frontiers in Radiology*, 1, 2021. ISSN 2673-8740.

[24] Jordão Bragantini, Bruno Moura, Alexandre X. Falcão, and Fábio A. M. Cappabianco. Grabber: A tool to improve convergence in interactive image segmentation. *Pattern Recognition Letters*, 140:267–273, December 2020. ISSN 0167-8655. doi: 10.1016/j.patrec.2020.10.012.

[25] Matthew Brett, Christopher J. Markiewicz, Michael Hanke, Marc-Alexandre Côté, Ben Cipollini, Paul McCarthy, Dorota Jarecka, Christopher P. Cheng, Yaroslav O. Halchenko, Michiel Cottaar, Eric Larson,

Satrajit Ghosh, Demian Wassermann, Stephan Gerhard, Gregory R. Lee, Hao-Ting Wang, Erik Kastman, Jakub Kaczmarzyk, Roberto Guidotti, Or Duek, Jonathan Daniel, Ariel Rokem, Cindee Madison, Brendan Moloney, Félix C. Morency, Mathias Goncalves, Ross Markello, Cameron Riddell, Christopher Burns, Jarrod Millman, Alexandre Gramfort, Jaakko Leppäkangas, Anibal Sólon, Jasper J.F. van den Bosch, Robert D. Vincent, Henry Braun, Krish Subramaniam, Krzysztof J. Gorgolewski, Pradeep Reddy Raamana, Julian Klug, B. Nolan Nichols, Eric M. Baker, Soichi Hayashi, Basile Pinsard, Christian Haselgrove, Mark Hymers, Oscar Esteban, Serge Koudoro, Fernando Pérez-García, Nikolaas N. Oosterhof, Bago Amirbekian, Ian Nimmo-Smith, Ly Nguyen, Samir Reddigari, Samuel St-Jean, Egor Panfilov, Eleftherios Garyfallidis, Gael Varoquaux, Jon Haitz Legarreta, Kevin S. Hahn, Oliver P. Hinds, Bennet Fauber, Jean-Baptiste Poline, Jon Stutters, Kesshi Jordan, Matthew Cieslak, Miguel Estevan Moreno, Valentin Haenel, Yannick Schwartz, Zvi Baratz, Benjamin C Darwin, Bertrand Thirion, Carl Gauthier, Dimitri Papadopoulos Orfanos, Igor Solovey, Ivan Gonzalez, Jath Palasubramaniam, Justin Lecher, Katrin Leinweber, Konstantinos Raktivan, Markéta Calábková, Peter Fischer, Philippe Gervais, Syam Gadde, Thomas Ballinger, Thomas Roos, Venkateswara Reddy Reddam, and freec84. Nipy/nibabel: 3.2.1. Zenodo, November 2020.

[26] Charlotte L. Brouwer, Djamal Boukerroui, Jorge Oliveira, Padraig Looney, Roel J. H. M. Steenbakkers, Johannes A. Langendijk, Stefan Both, and Mark J. Gooding. Assessment of manual adjustment performed in clinical practice following deep learning contouring for head and neck organs at risk in radiotherapy. *Physics and Imaging in Radiation Oncology*, 16:54–60, October 2020. ISSN 2405-6316. doi: 10.1016/j.phro.2020.10.001.

[27] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, October 2018. ISSN 08936080. doi: 10.1016/j.neunet.2018.07.011.

[28] Carrie J. Cai, Samantha Winter, David Steiner, Lauren Wilcox, and Michael Terry. "Hello AI": Uncovering the Onboarding Needs of Medical Practitioners for Human-AI Collaborative Decision-Making. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):104:1–104:24, November 2019. doi: 10.1145/3359206.

[29] Carlos E. Cardenas, Jinzhong Yang, Brian M. Anderson, Laurence E. Court, and Kristy B. Brock. Advances in Auto-Segmentation. *Seminars in Radiation Oncology*, 29(3):185–197, July 2019. ISSN 1053-4296. doi: 10.1016/j.semradonc.2019.02.001.

[30] M. Jorge Cardoso, Wenqi Li, Richard Brown, Nic Ma, Eric Kerfoot, Yiheng Wang, Benjamin Murrey, Andriy Myronenko, Can Zhao, Dong Yang, Vishwesh Nath, Yufan He, Ziyue Xu, Ali Hatamizadeh, Andriy

Myronenko, Wentao Zhu, Yun Liu, Mingxin Zheng, Yucheng Tang, Isaac Yang, Michael Zephyr, Behrooz Hashemian, Sachidanand Alle, Mohammad Zalbagi Darestani, Charlie Budd, Marc Modat, Tom Vercauteren, Guotai Wang, Yiwen Li, Yipeng Hu, Yunguan Fu, Benjamin Gorman, Hans Johnson, Brad Genereaux, Barbaros S. Erdal, Vikash Gupta, Andres Diaz-Pinto, Andre Dourson, Lena Maier-Hein, Paul F. Jaeger, Michael Baumgartner, Jayashree Kalpathy-Cramer, Mona Flores, Justin Kirby, Lee A. D. Cooper, Holger R. Roth, Daguang Xu, David Bericat, Ralf Floca, S. Kevin Zhou, Haris Shuaib, Keyvan Farahani, Klaus H. Maier-Hein, Stephen Aylward, Prerna Dogra, Sebastien Ourselin, and Andrew Feng. MONAI: An open-source framework for deep learning in healthcare, November 2022.

[31] Rich Caruana, Steve Lawrence, and C Lee Giles. Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping. page 7.

[32] Guanying Chen, Camilla Ruø Rasmussen, Dorte Bodin Dresbøll, Abraham George Smith, and Kristian Thorup-Kristensen. Dynamics of Deep Water and N Uptake of Oilseed Rape (Brassica napus L.) Under Varied N and Water Supply. *Frontiers in Plant Science*, 13:866288, 2022. ISSN 1664-462X. doi: 10.3389/fpls.2022.866288.

[33] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, volume 11211, pages 833–851. Springer International Publishing, Cham, 2018. ISBN 978-3-030-01233-5 978-3-030-01234-2. doi: 10.1007/978-3-030-01234-2_49.

[34] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. *arXiv:1606.06650 [cs]*, June 2016.

[35] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. In Sebastien Ourselin, Leo Joskowicz, Mert R. Sabuncu, Gozde Unal, and William Wells, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, Lecture Notes in Computer Science, pages 424–432, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46723-8. doi: 10.1007/978-3-319-46723-8_49.

[36] Corentin Clément, Joost Sleiderink, Simon Fiil Svane, Abraham George Smith, Efstathios Diamantopoulos, Dorte Bodin Desbrøll, and Kristian Thorup-Kristensen. Comparing the deep root growth and water uptake of intermediate wheatgrass (Kernza®) to alfalfa. *Plant and Soil*, 472(1): 369–390, March 2022. ISSN 1573-5036. doi: 10.1007/s11104-021-05248-6.

[37] Ronan Collobert and Samy Bengio. Links between perceptrons, MLPs and SVMs. In *Twenty-First International Conference on Machine Learning - ICML '04*, page 23, Banff, Alberta, Canada, 2004. ACM Press. doi: 10.1145/1015330.1015415.

[38] Samantha Cruz Rivera, Xiaoxuan Liu, An-Wen Chan, Alastair K. Denniston, Melanie J. Calvert, SPIRIT-AI and CONSORT-AI Working Group, SPIRIT-AI and CONSORT-AI Steering Group, and SPIRIT-AI and CONSORT-AI Consensus Group. Guidelines for clinical trial protocols for interventions involving artificial intelligence: The SPIRIT-AI extension. *Nature Medicine*, 26(9):1351–1363, September 2020. ISSN 1546-170X. doi: 10.1038/s41591-020-1037-7.

[39] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc'aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, Quoc V Le, and Andrew Y Ng. Large Scale Distributed Deep Networks. page 9.

[40] M. A. Deeley, A. Chen, R. D. Datteri, J. Noble, A. Cmelak, E. Donnelly, A. Malcolm, L. Moretti, J. Jaboin, K. Niermann, Eddy S. Yang, David S. Yu, and B. M. Dawant. Segmentation editing improves efficiency while reducing inter-expert variation and maintaining accuracy for normal brain tissues in the presence of space-occupying lesions. *Physics in Medicine and Biology*, 58(12):4071–4097, June 2013. ISSN 1361-6560. doi: 10.1088/0031-9155/58/12/4071.

[41] Geoff Delaney, Susannah Jacob, Carolyn Featherstone, and Michael Barton. The role of radiotherapy in cancer treatment. *Cancer*, 104(6):1129–1137, 2005. ISSN 1097-0142. doi: 10.1002/cncr.21324.

[42] Benjamin M. Delory, Maria C. Hernandez-Soriano, Tomke S. Wacker, Anastazija Dimitrova, Yiyang Ding, Laura A. Greeley, Jason Liang Pin Ng, Jennifer Mesa-Marín, Limeng Xie, Congcong Zheng, and Larry M. York. A snapshot of the root phenotyping landscape in 2021, January 2022.

[43] R. Ford Denison. Legume-imposed selection for more-efficient symbiotic rhizobia. *Proceedings of the National Academy of Sciences*, 118(22):e2107033118, June 2021. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.2107033118.

[44] R. Ford Denison and Katherine E. Muller. An evolutionary perspective on increasing net benefits to crops from symbiotic microbes. *Evolutionary Applications*, 15(10):1490–1504, 2022. ISSN 1752-4571. doi: 10.1111/eva.13384.

[45] Andres Diaz-Pinto, Sachidanand Alle, Alvin Ihsani, Muhammad Asad, Vishwesh Nath, Fernando Pérez-García, Pritesh Mehta, Wenqi Li, Holger R. Roth, Tom Vercauteren, Daguang Xu, Prerna Dogra, Sebastien

Ourselin, Andrew Feng, and M. Jorge Cardoso. MONAI Label: A framework for AI-assisted Interactive Labeling of 3D Medical Images. *arXiv e-prints*, 2022. URL `https://arxiv.org/pdf/2203.12362.pdf`.

[46] Qi Dou, Lequan Yu, Hao Chen, Yueming Jin, Xin Yang, Jing Qin, and Pheng-Ann Heng. 3D deeply supervised network for automated segmentation of volumetric medical images. *Medical Image Analysis*, 41:40–54, October 2017. ISSN 1361-8423. doi: 10.1016/j.media.2017.05.001.

[47] Chris Drummond and Robert C. Holte. Severe Class Imbalance: Why Better Algorithms Aren't the Answer. In João Gama, Rui Camacho, Pavel B. Brazdil, Alípio Mário Jorge, and Luís Torgo, editors, *Machine Learning: ECML 2005*, Lecture Notes in Computer Science, pages 539–546, Berlin, Heidelberg, 2005. Springer. ISBN 978-3-540-31692-3. doi: 10.1007/11564096_52.

[48] John J. Dudley and Per Ola Kristensson. A Review of User Interface Design for Interactive Machine Learning. *ACM Transactions on Interactive Intelligent Systems*, 8(2):8:1–8:37, June 2018. ISSN 2160-6455. doi: 10.1145/3185517.

[49] Berkeley Elias, Asheesh Lanba, Jose Costa Netto, Felix Fritschi, and Abraham Smith. Interactive Machine Learning Methods for the Quantification of Vascular Features in Soybean Images Obtained via Laser Ablation Tomography (LATscan). *Thinking Matters Symposium*, April 2021.

[50] Gary A. Ezzell, James M. Galvin, Daniel Low, Jatinder R. Palta, Isaac Rosen, Michael B. Sharpe, Ping Xia, Ying Xiao, Lei Xing, Cedric X. Yu, IMRT subcommitte, and AAPM Radiation Therapy committee. Guidance document on delivery, treatment planning, and clinical implementation of IMRT: Report of the IMRT Subcommittee of the AAPM Radiation Therapy Committee. *Medical Physics*, 30(8):2089–2115, August 2003. ISSN 0094-2405. doi: 10.1118/1.1591194.

[51] Jerry Alan Fails and Dan R. Olsen. Interactive machine learning. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, IUI '03, pages 39–45, Miami, Florida, USA, January 2003. Association for Computing Machinery. ISBN 978-1-58113-586-2. doi: 10.1145/604045.604056.

[52] Di Feng, Christian Haase-Schutz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, March 2021. ISSN 1524-9050, 1558-0016. doi: 10.1109/TITS.2020.2972974.

[53] Xue Feng, Kun Qing, Nicholas J. Tustison, Craig H. Meyer, and Quan Chen. Deep convolutional neural network for segmentation of thoracic

organs-at-risk using cropped 3D images. *Medical Physics*, 46(5):2169–2180, 2019. ISSN 2473-4209. doi: 10.1002/mp.13466.

[54] Xue Feng, Mark E. Bernard, Thomas Hunter, and Quan Chen. Improving accuracy and robustness of deep convolutional neural network based thoracic OAR segmentation. *Physics in Medicine & Biology*, 65(7):07NT01, March 2020. ISSN 0031-9155. doi: 10.1088/1361-6560/ab7877.

[55] N. Forbes, A. Smith, J. Petersen, C. Terrones-Campos, J. Reekie, S. Darkner, L. Specht, and I. Vogelius. MO-0716 Radiotherapy exposure and association with observed cardiovascular toxicity in over 5000 patients. *Radiotherapy and Oncology*, 170:S627–S628, 2022.

[56] N. Forbes, A. Smith, J. Petersen, C. Terrones-Campos, J. Reekie, S. Darkner, L. Specht, and I. Vogelius. MO-0716 Radiotherapy exposure and association with observed cardiovascular toxicity in over 5000 patients. *Radiotherapy and Oncology*, 170:S627–S628, 2022.

[57] Nicolás Gaggion, Federico Ariel, Vladimir Daric, Éric Lambert, Simon Legendre, Thomas Roulé, Alejandra Camoirano, Diego H Milone, Martin Crespi, Thomas Blein, and Enzo Ferrante. ChronoRoot: High-throughput phenotyping by deep segmentation networks reveals novel temporal parameters of plant root system architecture. *GigaScience*, 10(7):giab052, July 2021. ISSN 2047-217X. doi: 10.1093/gigascience/giab052.

[58] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):44:1–44:37, March 2014. ISSN 0360-0300. doi: 10.1145/2523813.

[59] Mohsen Ghafoorian, Alireza Mehrtash, Tina Kapur, Nico Karssemeijer, Elena Marchiori, Mehran Pesteie, Charles R. G. Guttmann, Frank-Erik de Leeuw, Clare M. Tempany, Bram van Ginneken, Andriy Fedorov, Purang Abolmaesumi, Bram Platel, and William M. Wells III. Transfer Learning for Domain Adaptation in MRI: Application in Brain Lesion Segmentation. *arXiv:1702.07841 [cs]*, 10435:516–524, 2017. doi: 10.1007/978-3-319-66179-7_59.

[60] Eli Gibson, Yipeng Hu, Nooshin Ghavami, Hashim U. Ahmed, Caroline Moore, Mark Emberton, Henkjan J. Huisman, and Dean C. Barratt. Inter-site Variability in Prostate Segmentation Accuracy Using Deep Learning. In Alejandro F. Frangi, Julia A. Schnabel, Christos Davatzikos, Carlos Alberola-López, and Gabor Fichtinger, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, Lecture Notes in Computer Science, pages 506–514, Cham, 2018. Springer International Publishing. ISBN 978-3-030-00937-3. doi: 10.1007/978-3-030-00937-3_58.

[61] Estibaliz Gómez-de-Mariscal, Carlos García-López-de-Haro, Wei Ouyang, Laurène Donati, Emma Lundberg, Michael Unser, Arrate Muñoz-Barrutia, and Daniel Sage. DeepImageJ: A user-friendly environment to run deep learning models in ImageJ. *Nature Methods*, 18(10):1192–1195, October 2021. ISSN 1548-7105. doi: 10.1038/s41592-021-01262-9.

[62] F. Gonda, V. Kaynig, Thouis R. Jones, D. Haehn, J. W. Lichtman, T. Parag, and H. Pfister. ICON: An interactive approach to train deep neural networks for segmentation of neuronal structures. In *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pages 327–331, Melbourne, Australia, April 2017. doi: 10.1109/ISBI.2017.7950530.

[63] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. Adaptive Computation and Machine Learning Series. MIT Press, Cambridge, MA, USA, November 2016. ISBN 978-0-262-03561-3.

[64] Mark J. Gooding, Annamarie J. Smith, Maira Tariq, Paul Aljabar, Devis Peressutti, Judith van der Stoep, Bart Reymen, Daisy Emans, Djoya Hattu, Judith van Loon, Maud de Rooy, Rinus Wanders, Stephanie Peeters, Tim Lustberg, Johan van Soest, Andre Dekker, and Wouter van Elmpt. Comparative evaluation of autocontouring in clinical practice: A practical method using the Turing test. *Medical Physics*, 45(11):5105–5115, 2018. ISSN 2473-4209. doi: 10.1002/mp.13200.

[65] A. Green, M. C. Aznar, R. Muirhead, and E. M. Vasquez Osorio. Reading the Mind of a Machine: Hopes and Hypes of Artificial Intelligence for Clinical Oncology Imaging. *Clinical Oncology*, 34(3):e130–e134, March 2022. ISSN 0936-6555, 1433-2981. doi: 10.1016/j.clon.2021.11.008.

[66] Nela Grimova, Martin Macas, and Vaclav Gerla. Addressing the Cold Start Problem in Active Learning Approach Used For Semi-automated Sleep Stages Classification. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 2249–2253, December 2018. doi: 10.1109/BIBM.2018.8621434.

[67] Charley Gros, Benjamin De Leener, Atef Badji, Josefina Maranzano, Dominique Eden, Sara M. Dupont, Jason Talbott, Ren Zhuoquiong, Yaou Liu, Tobias Granberg, Russell Ouellette, Yasuhiko Tachibana, Masaaki Hori, Kouhei Kamiya, Lydia Chougar, Leszek Stawiarz, Jan Hillert, Elise Bannier, Anne Kerbrat, Gilles Edan, Pierre Labauge, Virginie Callot, Jean Pelletier, Bertrand Audoin, Henitsoa Rasoanandrianina, Jean-Christophe Brisset, Paola Valsasina, Maria A. Rocca, Massimo Filippi, Rohit Bakshi, Shahamat Tauhid, Ferran Prados, Marios Yiannakas, Hugh Kearney, Olga Ciccarelli, Seth Smith, Constantina Andrada Treaba, Caterina Mainero, Jennifer Lefeuvre, Daniel S. Reich, Govind Nair, Vincent Auclair, Donald G. McLaren, Allan R. Martin, Michael G. Fehlings, Shahabeddin Vahdat, Ali Khatibi, Julien Doyon, Timothy Shepherd, Erik Charlson,

Sridar Narayanan, and Julien Cohen-Adad. Automatic segmentation of the spinal cord and intramedullary multiple sclerosis lesions with convolutional neural networks. *NeuroImage*, 184:901–915, January 2019. ISSN 1053-8119. doi: 10.1016/j.neuroimage.2018.09.081.

[68] R. Hallows, L. Glazier, M. S. Katz, M. Aznar, and M. Williams. Safe and Ethical Artificial Intelligence in Radiotherapy – Lessons Learned From the Aviation Industry. *Clinical Oncology*, 34(2):99–101, February 2022. ISSN 0936-6555. doi: 10.1016/j.clon.2021.11.019.

[69] Eusun Han, Timo Kautz, Ute Perkons, Marcel Lüsebrink, Ralf Pude, and Ulrich Köpke. Quantification of soil biopore density after perennial fodder cropping. *Plant and Soil*, 394(1):73–85, September 2015. ISSN 1573-5036. doi: 10.1007/s11104-015-2488-3.

[70] Eusun Han, Timo Kautz, Ute Perkons, Daniel Uteau, Stephan Peth, Ning Huang, Rainer Horn, and Ulrich Köpke. Root growth dynamics inside and outside of soil biopores as affected by crop sequence determined with the profile wall method. *Biology and Fertility of Soils*, 51(7):847–856, October 2015. ISSN 1432-0789. doi: 10.1007/s00374-015-1032-1.

[71] Eusun Han, Timo Kautz, Ning Huang, and Ulrich Köpke. Dynamics of plant nutrient uptake as affected by biopore-associated root growth in arable subsoil. *Plant and Soil*, 415(1):145–160, June 2017. ISSN 1573-5036. doi: 10.1007/s11104-016-3150-4.

[72] Eusun Han, Abraham George Smith, Roman Kemper, Rosemary White, John Kirkegaard, Kristian Thorup-Kristensen, and Miriam Athmann. Digging roots is easier with AI. *bioRxiv*, page 2020.12.01.397034, December 2020. doi: 10.1101/2020.12.01.397034.

[73] Eusun Han, John A. Kirkegaard, Rosemary White, Abraham George Smith, Kristian Thorup-Kristensen, Timo Kautz, and Miriam Athmann. Deep learning with multisite data reveals the lasting effects of soil type, tillage and vegetation history on biopore genesis. *Geoderma*, 425:116072, November 2022. ISSN 0016-7061. doi: 10.1016/j.geoderma.2022.116072.

[74] Ali Hatamizadeh, Yucheng Tang, Vishwesh Nath, Dong Yang, Andriy Myronenko, Bennett Landman, Holger R. Roth, and Daguang Xu. UNETR: Transformers for 3D Medical Image Segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 574–584, 2022.

[75] H. Hauggaard-Nielsen, P. Ambus, and E.S. Jensen. Temporal and spatial distribution of roots and competition for nitrogen in pea-barley intercrops – a field study employing 32P technique. *Plant and Soil*, 236(1):63–74, September 2001. ISSN 1573-5036. doi: 10.1023/A:1011909414400.

[76] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, December 2015.

[77] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *arXiv:1502.01852 [cs]*, February 2015.

[78] Zhuoxun He, Lingxi Xie, Xin Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Data Augmentation Revisited: Rethinking the Distribution Gap between Clean and Augmented Data. *arXiv:1909.09148 [cs, stat]*, September 2019.

[79] Nicholas Heller, Joshua Dean, and Nikolaos Papanikolopoulos. Imperfect Segmentation Labels: How Much Do They Matter? *arXiv:1806.04618 [cs]*, June 2018.

[80] Edward G. A. Henderson, Marcel van Herk, and Eliana M. Vasquez Osorio. The impact of training dataset size and ensemble inference strategies on head and neck auto-segmentation. 2023. doi: 10.48550/ARXIV.2303.17318.

[81] Alex Hernández-García and Peter König. Data augmentation instead of explicit regularization. *arXiv:1806.03852 [cs]*, June 2019.

[82] David Joon Ho, Narasimhan P. Agaram, Peter J. Schüffler, Chad M. Vanderbilt, Marc-Henri Jean, Meera R. Hameed, and Thomas J. Fuchs. Deep Interactive Learning: An Efficient Labeling Approach for Deep Learning-Based Osteosarcoma Treatment Response Assessment. In Anne L. Martel, Purang Abolmaesumi, Danail Stoyanov, Diana Mateus, Maria A. Zuluaga, S. Kevin Zhou, Daniel Racoceanu, and Leo Joskowicz, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, Lecture Notes in Computer Science, pages 540–549, Cham, 2020. Springer International Publishing. ISBN 978-3-030-59722-1. doi: 10.1007/978-3-030-59722-1_52.

[83] Andreas Holzinger. Interactive machine learning for health informatics: When do we need the human-in-the-loop? *Brain Informatics*, 3(2):119–131, June 2016. ISSN 2198-4026. doi: 10.1007/s40708-016-0042-6.

[84] Andreas Holzinger, Markus Plass, Michael Kickmeier-Rust, Katharina Holzinger, Gloria Cerasela Crişan, Camelia-M. Pintea, and Vasile Palade. Interactive machine learning: Experimental evidence for the human in the algorithmic loop. *Applied Intelligence*, 49(7):2401–2414, July 2019. ISSN 1573-7497. doi: 10.1007/s10489-018-1361-5.

[85] Bradford S. Hoppe, James E. Bates, Nancy P. Mendenhall, Christopher G. Morris, Debbie Louis, Meng Wei Ho, Richard T. Hoppe, Marwan Shaikh, Zuofeng Li, and Stella Flampouri. The Meaningless Meaning of Mean Heart Dose in Mediastinal Lymphoma in the Modern Radiation Therapy

Era. *Practical Radiation Oncology*, 10(3):e147–e154, May 2020. ISSN 1879-8500. doi: 10.1016/j.prro.2019.09.015.

[86] Yang Hu, Andrea Soltoggio, Russell Lock, and Steve Carter. A Fully Convolutional Two-Stream Fusion Network for Interactive Image Segmentation. *arXiv:1807.02480 [cs]*, July 2018.

[87] Bohao Huang, Daniel Reichman, Leslie M. Collins, Kyle Bradbury, and Jordan M. Malof. Tiling and Stitching Segmentation Output for Remote Sensing: Basic Challenges and Recommendations. *arXiv:1805.12219 [cs]*, February 2019.

[88] Hyun Do Huh and Seonghoon Kim. History of Radiation Therapy Technology. *Progress in Medical Physics*, 31(3):124–134, September 2020. ISSN 2508-4445, 2508-4453. doi: 10.14316/pmp.2020.31.3.124.

[89] Elizabeth Huynh, Ahmed Hosny, Christian Guthier, Danielle S. Bitterman, Steven F. Petit, Daphne A. Haas-Kogan, Benjamin Kann, Hugo J. W. L. Aerts, and Raymond H. Mak. Artificial intelligence in radiation oncology. *Nature Reviews Clinical Oncology*, 17(12):771–781, December 2020. ISSN 1759-4782. doi: 10.1038/s41571-020-0417-8.

[90] Bulat Ibragimov and Lei Xing. Segmentation of organs-at-risks in head and neck CT images using convolutional neural networks. *Medical Physics*, 44(2):547–557, 2017. ISSN 2473-4209. doi: 10.1002/mp.12045.

[91] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv:1502.03167 [cs]*, March 2015.

[92] Fabian Isensee, Jens Petersen, Simon A A Kohl, Paul F Jager, and Klaus H Maier-Hein. nnU-Net: Breaking the Spell on Successful Medical Image Segmentation. page 9.

[93] Fabian Isensee, Paul Jaeger, Peter M. Full, Ivo Wolf, Sandy Engelhardt, and Klaus H. Maier-Hein. Automatic Cardiac Disease Assessment on cine-MRI via Time-Series Segmentation and Domain Specific Features. *arXiv:1707.00587 [cs]*, 10663, 2018. doi: 10.1007/978-3-319-75541-0.

[94] Fabian Isensee, Jens Petersen, Andre Klein, David Zimmerer, Paul F. Jaeger, Simon Kohl, Jakob Wasserthal, Gregor Koehler, Tobias Norajitra, Sebastian Wirkert, and Klaus H. Maier-Hein. nnU-Net: Self-adapting Framework for U-Net-Based Medical Image Segmentation. *arXiv:1809.10486 [cs]*, September 2018.

[95] Fabian Isensee, Paul F. Jaeger, Simon A. A. Kohl, Jens Petersen, and Klaus H. Maier-Hein. nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, (2):1–9, December 2020. ISSN 1548-7105. doi: 10.1038/s41592-020-01008-z.

[96] Fabian Isensee, Paul F. Jaeger, Simon A. A. Kohl, Jens Petersen, and Klaus H. Maier-Hein. nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18(2):203–211, February 2021. ISSN 1548-7105. doi: 10.1038/s41592-020-01008-z. URL `https://www.nature.com/articles/s41592-020-01008-z`. Number: 2 Publisher: Nature Publishing Group.

[97] Stefan Jaeger, Sema Candemir, Sameer Antani, Yì-Xiáng J. Wáng, Pu-Xuan Lu, and George Thoma. Two public chest X-ray datasets for computer-aided screening of pulmonary diseases. *Quantitative Imaging in Medicine and Surgery*, 4(6):475–477, December 2014. ISSN 2223-4292. doi: 10.3978/j.issn.2223-4292.2014.11.20.

[98] Stasa Jelercic and Mirjana Rajer. The role of PET-CT in radiotherapy planning of solid tumours. *Radiology and Oncology*, 49(1):1–9, March 2015. ISSN 1318-2099. doi: 10.2478/raon-2013-0071.

[99] Yu Jiang and Changying Li. Convolutional Neural Networks for Image-Based High-Throughput Plant Phenotyping: A Review. *Plant Phenomics*, 2020:1–22, April 2020. ISSN 2643-6515. doi: 10.34133/2020/4152816. URL `https://spj.sciencemag.org/journals/plantphenomics/2020/4152816/`.

[100] Leo Joskowicz, D. Cohen, N. Caplan, and J. Sosna. Inter-observer variability of manual contour delineation of structures in CT. *European Radiology*, 29(3):1391–1399, March 2019. ISSN 1432-1084. doi: 10.1007/s00330-018-5695-5.

[101] A. Kamilaris and F. X. Prenafeta-Boldú. A review of the use of convolutional neural networks in agriculture. *The Journal of Agricultural Science*, 156(3):312–322, April 2018. ISSN 0021-8596, 1469-5146. doi: 10.1017/S0021859618000436.

[102] Radmir Karamov, Christian Breite, Stepan V. Lomov, Ivan Sergeichev, and Yentl Swolfs. Super-Resolution Processing of Synchrotron CT Images for Automated Fibre Break Analysis of Unidirectional Composites. *Polymers*, 15(9):2206, January 2023. ISSN 2073-4360. doi: 10.3390/polym15092206.

[103] Timo Kautz. Research on subsoil biopores and their functions in organically managed soils: A review. *Renewable Agriculture and Food Systems*, 30(4):318–327, August 2015. ISSN 1742-1705, 1742-1713. doi: 10.1017/S1742170513000549.

[104] Benjamin Kellenberger, Diego Marcos, Sylvain Lobry, and Devis Tuia. Half a Percent of Labels is Enough: Efficient Animal Detection in UAV Imagery using Deep CNNs and Active Learning. *IEEE Transactions on Geoscience and Remote Sensing*, 57:1–10, 2019. ISSN 0196-2892, 1558-0644. doi: 10.1109/TGRS.2019.2927393.

[105] Chris Van Kessel, William R. Horwath, Ueli Hartwig, David Harris, and Andreas LÜscher. Net soil carbon input under ambient and elevated CO2 concentrations: Isotopic evidence after 4 years. *Global Change Biology*, 6 (4):435–444, 2000. ISSN 1365-2486. doi: 10.1046/j.1365-2486.2000.00318.x.

[106] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, January 2017.

[107] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment Anything, April 2023.

[108] Kendall Kiser, Arko Barman, Sonja Stieb, Clifton D Fuller, and Luca Giancardo. Novel Autosegmentation Spatial Similarity Metrics Capture the Time Required to Correct Segmentations Better than Traditional Metrics in a Thoracic Cavity Segmentation Workflow. preprint, Radiology and Imaging, May 2020. URL `http://medrxiv.org/lookup/doi/10.1101/2020.05.14.20102103`.

[109] Sebastian Klüter, Sonja Katayama, C. Katharina Spindeldreier, Stefan A. Koerber, Gerald Major, Markus Alber, Sati Akbaba, Jürgen Debus, and Juliane Hörner-Rieber. First prospective clinical evaluation of feasibility and patient acceptance of magnetic resonance-guided radiotherapy in Germany. *Strahlentherapie Und Onkologie: Organ Der Deutschen Rontgengesellschaft ... [et Al]*, 196(8):691–698, August 2020. ISSN 1439-099X. doi: 10.1007/s00066-020-01578-z.

[110] Theodora Kontogianni, Michael Gygli, Jasper Uijlings, and Vittorio Ferrari. Continuous Adaptation for Interactive Object Segmentation by Learning from Corrections. *arXiv:1911.12709 [cs]*, November 2019.

[111] Theodora Kontogianni, Michael Gygli, Jasper Uijlings, and Vittorio Ferrari. Continuous Adaptation for Interactive Object Segmentation by Learning from Corrections. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, Lecture Notes in Computer Science, pages 579–596, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58517-4. doi: 10.1007/978-3-030-58517-4_34.

[112] Ulrich Kopke, Miriam Athmann, Eusun Han, Timo Kautz, Ulrich Kopke, Miriam Athmann, Eusun Han, and Timo Kautz. Optimising Cropping Techniques for Nutrient and Environmental Management in Organic Agriculture. 2015. doi: 10.22004/AG.ECON.230376.

[113] Denis Kutnar, Bas H. M. van der Velden, Marta Girones Sanguesa, Mirjam I. Geerlings, J. Matthijs Biesbroek, and Hugo J. Kuijf. MixLacune: Segmentation of lacunes of presumed vascular origin. 2021. doi: 10.48550/ARXIV.2108.02483.

[114] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature14539.

[115] Cecilia S Lee and Aaron Y Lee. Clinical applications of continual learning machine learning. *The Lancet Digital Health*, 2(6):e279–e281, June 2020. ISSN 25897500. doi: 10.1016/S2589-7500(20)30102-3.

[116] Yang Lei, Yabo Fu, Tonghe Wang, Richard L. J. Qiu, Walter J. Curran, Tian Liu, and Xiaofeng Yang. Deep Learning in Multi-organ Segmentation. 2020. doi: 10.48550/ARXIV.2001.10619.

[117] Victor Lempitsky, Pushmeet Kohli, Carsten Rother, and Toby Sharp. Image segmentation with a bounding box prior. In *2009 IEEE 12th International Conference on Computer Vision*, pages 277–284, September 2009. doi: 10.1109/ICCV.2009.5459262.

[118] Wenqi Li, Guotai Wang, Lucas Fidon, Sebastien Ourselin, M. Jorge Cardoso, and Tom Vercauteren. On the Compactness, Efficiency, and Representation of 3D Convolutional Networks: Brain Parcellation as a Pretext Task. *arXiv:1707.01992 [cs]*, 10265:348–360, 2017. doi: 10.1007/978-3-319-59050-9_28.

[119] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Interactive Image Segmentation with Latent Diversity. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 577–585, Salt Lake City, UT, June 2018. IEEE. ISBN 978-1-5386-6420-9. doi: 10.1109/CVPR.2018.00067.

[120] Blerina Lika, Kostas Kolomvatsos, and Stathes Hadjiefthymiades. Facing the cold start problem in recommender systems. *Expert Systems with Applications*, 41(4, Part 2):2065–2073, March 2014. ISSN 0957-4174. doi: 10.1016/j.eswa.2013.09.005.

[121] Hubert Lin, Paul Upchurch, and Kavita Bala. Block Annotation: Better Image Annotation for Semantic Segmentation with Sub-Image Decomposition. *arXiv:2002.06626 [cs, eess]*, February 2020.

[122] Charles X. Ling and Victor S. Sheng. Class Imbalance Problem. In Claude Sammut and Geoffrey I. Webb, editors, *Encyclopedia of Machine Learning*, pages 171–171. Springer US, Boston, MA, 2010. ISBN 978-0-387-30164-8. doi: 10.1007/978-0-387-30164-8_110.

[123] Guillaume Lobet, Xavier Draye, and Claire Périlleux. An online database for plant image analysis software tools. *Plant Methods*, 9(1):38, October 2013. ISSN 1746-4811. doi: 10.1186/1746-4811-9-38.

[124] Tim Lustberg, Johan van Soest, Mark Gooding, Devis Peressutti, Paul Aljabar, Judith van der Stoep, Wouter van Elmpt, and Andre Dekker. Clinical evaluation of atlas and deep learning based automatic contouring

for lung cancer. *Radiotherapy and Oncology*, 126(2):312–317, February 2018. ISSN 01678140. doi: 10.1016/j.radonc.2017.11.012.

[125] Brendon Lutnick, David Manthey, Jan Becker, Brandon Ginley, Katharina Moos, Jonathan Zuckerman, Luís Rodrigues, Alexander Gallan, Laura Barisoni, Charles Alpers, Xiaoxin Wang, Komuraiah Myakala, Bryce Jones, Moshe Levi, Jeffrey Kopp, Teruhiko Yoshida, Jarcy Zee, Sanjay Jain, and Angela Victoria-Castro. A user-friendly tool for cloud-based whole slide image segmentation with examples from renal histopathology. *Communications Medicine*, 2:105, 08 2022. doi: 10.1038/s43856-022-00138-z.

[126] Jonathan P. Lynch. TURNER REVIEW No. 14. Roots of the Second Green Revolution. *Australian Journal of Botany*, 55(5):493, 2007. ISSN 0067-1924. doi: 10.1071/BT06118.

[127] Thomas Rockwell Mackie, Jeff Kapatoes, Ken Ruchala, Weiguo Lu, Chuan Wu, Gustavo Olivera, Lisa Forrest, Wolfgang Tome, Jim Welsh, Robert Jeraj, Paul Harari, Paul Reckwerdt, Bhudatt Paliwal, Mark Ritter, Harry Keller, Jack Fowler, and Minesh Mehta. Image guidance for precise conformal radiotherapy. *International Journal of Radiation Oncology, Biology, Physics*, 56(1):89–105, May 2003. ISSN 0360-3016. doi: 10.1016/S0360-3016(03)00090-7.

[128] B. N. Madhusudhan, S. K. Sahoo, F. Alvarez-Borges, S. Ahmed, L. J. North, and A. I. Best. Gas Bubble Dynamics During Methane Hydrate Formation and its Influence on Geophysical Properties of Sediment Using High-Resolution Synchrotron Imaging and Rock Physics Modeling. *Frontiers in Earth Science*, 10, 2022. ISSN 2296-6463.

[129] Sabarinath Mahadevan, Paul Voigtlaender, and Bastian Leibe. Iteratively Trained Interactive Segmentation. *arXiv:1805.04398 [cs]*, May 2018.

[130] Marta Malinowska, Anja Karine Ruud, Just Jensen, Simon Fiil Svane, Abraham George Smith, Andrea Bellucci, Ingo Lenk, Istvan Nagy, Mattia Fois, Thomas Didion, Kristian Thorup-Kristensen, Christian Sig Jensen, and Torben Asp. Relative importance of genotype, gene expression, and DNA methylation on complex traits in perennial ryegrass. *The Plant Genome*, 15(4):e20253, 2022. ISSN 1940-3372. doi: 10.1002/tpg2.20253.

[131] Hari McGrath, Peichao Li, Reuben Dorent, Robert Bradford, Shakeel Saeed, Sotirios Bisdas, Sebastien Ourselin, Jonathan Shapey, and Tom Vercauteren. Manual segmentation versus semi-automated segmentation for quantifying vestibular schwannoma volume on mri. *International Journal of Computer Assisted Radiology and Surgery*, 15, 07 2020. doi: 10.1007/s11548-020-02222-y.

[132] Philippe Meyer, Vincent Noblet, Christophe Mazzara, and Alex Lallement. Survey on deep learning for radiotherapy. *Computers in Biology and Medicine*, 98:126–146, July 2018. ISSN 1879-0534. doi: 10.1016/j.compbiomed.2018.05.018.

[133] Chris J. Michael, Dina Acklin, and Jaelle Scheuerman. On Interactive Machine Learning and the Potential of Cognitive Feedback. *arXiv:2003.10365 [cs]*, March 2020.

[134] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. *arXiv:1606.04797 [cs]*, June 2016.

[135] Marie Louise Holm Milo, Birgitte Vrou Offersen, Troels Bechmann, Axel Cosmus Pyndt Diederichsen, Christian Rønn Hansen, Eva Holtved, Mirjana Josipovic, Tamás Lörincz, Maja Vestmø Maraldo, Mette Holck Nielsen, Marianne Nordsmark, Petra Witt Nyström, Mette Pøhl, Hanne Krogh Rose, Tine Schytte, Esben Svitzer Yates, and Ebbe Laugaard Lorenzen. Delineation of whole heart and substructures in thoracic radiation therapy: National guidelines and contouring atlas by the Danish Multidisciplinary Cancer Groups. *Radiotherapy and Oncology*, 150:121–127, September 2020. ISSN 0167-8140. doi: 10.1016/j.radonc.2020.06.015.

[136] Massimo Minervini, Hanno Scharr, and Sotirios A. Tsaftaris. Image Analysis: The New Bottleneck in Plant Phenotyping [Applications Corner]. *IEEE Signal Processing Magazine*, 32(4):126–131, July 2015. ISSN 1558-0792. doi: 10.1109/MSP.2015.2405111.

[137] Arezoo Modiri, Ivan Vogelius, Laura Ann Rechner, Lotte Nygård, Søren M Bentzen, and Lena Specht. Outcome-based multiobjective optimization of lymphoma radiation therapy plans. *The British Journal of Radiology*, 94(1127):20210303, November 2021. ISSN 0007-1285. doi: 10.1259/bjr.20210303.

[138] Daniel Monnens, R. Ford Denison, and Walid Sadok. Rising vapor-pressure deficit increases nitrogen fixation in a legume crop. *New Phytologist*, n/a(n/a). ISSN 1469-8137. doi: 10.1111/nph.18929.

[139] Nelson Morgan and Hervé Bourlard. Generalization and parameter estimation in feedforward nets: Some experiments. In *Advances in Neural Information Processing Systems*, pages 630–637, 1990.

[140] Andriy Myronenko. 3D MRI brain tumor segmentation using autoencoder regularization, November 2018.

[141] Andriy Myronenko. 3D MRI Brain Tumor Segmentation Using Autoencoder Regularization. In Alessandro Crimi, Spyridon Bakas, Hugo Kuijf, Farahani Keyvan, Mauricio Reyes, and Theo van Walsum, editors, *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain*

Injuries, Lecture Notes in Computer Science, pages 311–320, Cham, 2019. Springer International Publishing. ISBN 978-3-030-11726-9. doi: 10.1007/978-3-030-11726-9_28.

[142] Richard Nair, Martin Strube, Martin Hertel, Olaf Kolle, Victor Rolo, and Mirco Migliavacca. High Frequency Root Dynamics: Sampling And Interpretation Using Replicated Robotic Minirhizotrons. *Journal of Experimental Botany*, page erac427, October 2022. ISSN 1460-2431. doi: 10.1093/jxb/erac427.

[143] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep Double Descent: Where Bigger Models and More Data Hurt. *arXiv:1912.02292 [cs, stat]*, December 2019.

[144] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt*. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, December 2021. ISSN 1742-5468. doi: 10.1088/1742-5468/ac3a74.

[145] Narendra Narisetti, Michael Henke, Christiane Seiler, Rongli Shi, Astrid Junker, Thomas Altmann, and Evgeny Gladilin. Semi-automated Root Image Analysis (saRIA). *Scientific Reports*, 9(1):1–10, December 2019. ISSN 2045-2322. doi: 10.1038/s41598-019-55876-3.

[146] Narendra Narisetti, Michael Henke, Christiane Seiler, Astrid Junker, Jörn Ostermann, Thomas Altmann, and Evgeny Gladilin. Fully-automated root image analysis (faRIA). *Scientific Reports*, 11(1):16047, August 2021. ISSN 2045-2322. doi: 10.1038/s41598-021-95480-y.

[147] Mauricio Orbes-Arteaga, Thomas Varsavsky, Lauge Sorensen, Mads Nielsen, Akshay Pai, Sebastien Ourselin, Marc Modat, and M. Jorge Cardoso. Augmentation based unsupervised domain adaptation, February 2022.

[148] Isabella Østerlund, Staffan Persson, and Zoran Nikoloski. Tracing and tracking filamentous structures across scales: A systematic review. *Computational and Structural Biotechnology Journal*, 21:452–462, January 2023. ISSN 2001-0370. doi: 10.1016/j.csbj.2022.12.023.

[149] Dim P. Papadopoulos, Jasper R. R. Uijlings, Frank Keller, and Vittorio Ferrari. We Don't Need No Bounding-Boxes: Training Object Class Detectors Using Only Human Verification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 854–863, Las Vegas, NV, USA, June 2016. IEEE. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.99.

[150] Dim P. Papadopoulos, Jasper R. R. Uijlings, Frank Keller, and Vittorio Ferrari. Extreme Clicking for Efficient Object Annotation. In *2017 IEEE*

International Conference on Computer Vision (ICCV), pages 4940–4949, Venice, October 2017. IEEE. ISBN 978-1-5386-1032-9. doi: 10.1109/ICCV.2017.528.

[151] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. page 4.

[152] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. October 2017. URL `https://openreview.net/forum?id=BJJsrmfCZ`.

[153] Fábio Perez, Cristina Vasconcelos, Sandra Avila, and Eduardo Valle. Data Augmentation for Skin Lesion Analysis. *arXiv:1809.01442 [cs]*, 11041: 303–311, 2018. doi: 10.1007/978-3-030-01201-4_33.

[154] Mathias Perslev, Erik Bjørnager Dam, Akshay Pai, and Christian Igel. One Network to Segment Them All: A General, Lightweight System for Accurate 3D Medical Image Segmentation. In Dinggang Shen, Tianming Liu, Terry M. Peters, Lawrence H. Staib, Caroline Essert, Sean Zhou, Pew-Thian Yap, and Ali Khan, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, Lecture Notes in Computer Science, pages 30–38, Cham, 2019. Springer International Publishing. ISBN 978-3-030-32245-8. doi: 10.1007/978-3-030-32245-8_4.

[155] Steve Pieper, Michael Halle, and Ron Kikinis. 3d slicer. In *2004 2nd IEEE international symposium on biomedical imaging: nano to macro (IEEE Cat No. 04EX821)*, pages 632–635. IEEE, 2004.

[156] Michael P. Pound, Jonathan A. Atkinson, Darren M. Wells, Tony P. Pridmore, and Andrew P. French. Deep Learning for Multi-Task Plant Phenotyping. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2055–2063, 2017.

[157] Lutz Prechelt. Early Stopping — but when? page 15.

[158] R Ford Denison: Darwinian Agriculture. Using artificial intelligence to help crops guide the evolution of nitrogen-fixing bacteria, January 2022.

[159] Charlotte Rambla. Development of phenotyping and selection tools to support breeding of future wheat varieties with improved root systems. September 2022. doi: 10.14264/a7508ff.

[160] Gonzalo Ramos, Christopher Meek, Patrice Simard, Jina Suh, and Soroush Ghorashi. Interactive machine teaching: A human-centered approach to building machine-learned models. *Human–Computer Interaction*, 35(5-6):413–451, November 2020. ISSN 0737-0024, 1532-7051. doi: 10.1080/07370024.2020.1734931.

[161] Camilla Ruø Rasmussen, Kristian Thorup-Kristensen, and Dorte Bodin Dresbøll. Uptake of subsoil water below 2 m fails to alleviate drought response in deep-rooted Chicory (Cichorium intybus L.). *Plant and Soil*, 446(1):275–290, January 2020. ISSN 1573-5036. doi: 10.1007/s11104-019-04349-7.

[162] Irene Skovby Rasmussen and Kristian Thorup-Kristensen. Does earlier sowing of winter wheat improve root growth and N uptake? *Field Crops Research*, 196:10–21, September 2016. ISSN 0378-4290. doi: 10.1016/j.fcr.2016.05.009.

[163] Irene Skovby Rasmussen, Dorte Bodin Dresbøll, and Kristian Thorup-Kristensen. Winter wheat cultivars and nitrogen (N) fertilization—Effects on root growth, N uptake efficiency and N use efficiency. *European Journal of Agronomy*, 68:38–49, August 2015. ISSN 1161-0301. doi: 10.1016/j.eja.2015.04.003.

[164] Mathis Ersted Rasmussen, Jasper Albertus Nijkamp, Jesper Grau Eriksen, and Stine Sofia Korreman. A simple single-cycle interactive strategy to improve deep learning-based segmentation of organs-at-risk in head-and-neck cancer. *Physics and Imaging in Radiation Oncology*, 26:100426, April 2023. ISSN 24056316. doi: 10.1016/j.phro.2023.100426.

[165] Maria C. Rebolledo, Alexandra L. Peña, Jorge Duitama, Daniel F. Cruz, Michael Dingkuhn, Cecile Grenier, and Joe Tohme. Combining Image Analysis, Genome Wide Association Studies and Different Field Trials to Reveal Stable Genetic Regions Related to Panicle Architecture and the Number of Spikelets per Panicle in Rice. *Frontiers in Plant Science*, 7, 2016. ISSN 1664-462X. doi: 10.3389/fpls.2016.01384.

[166] Boris Rewald, Catharina Meinen, Michael Trockenbrodt, Jhonathan E. Ephrath, and Shimon Rachmilevitch. Root taxa identification in plant mixtures – current techniques and future challenges. *Plant and Soil*, 359 (1-2):165–182, October 2012. ISSN 0032-079X, 1573-5036. doi: 10.1007/s11104-012-1164-0.

[167] Sarah M. Rich, Jack Christopher, Richard Richards, and Michelle Watt. Root phenotypes of young wheat plants grown in controlled environments show inconsistent correlation with mature root traits in the field. *Journal of Experimental Botany*. doi: 10.1093/jxb/eraa201.

[168] David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. Deep Learning is Robust to Massive Label Noise. *arXiv:1705.10694 [cs]*, February 2018.

[169] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention –*

MICCAI 2015, volume 9351, pages 234–241. Springer International Publishing, Cham, 2015. ISBN 978-3-319-24573-7 978-3-319-24574-4. doi: 10.1007/978-3-319-24574-4_28.

[170] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4. doi: 10.1007/978-3-319-24574-4_28.

[171] Holger R. Roth, Le Lu, Ari Seff, Kevin M. Cherry, Joanne Hoffman, Shijun Wang, Jiamin Liu, Evrim Turkbey, and Ronald M. Summers. A New 2.5D Representation for Lymph Node Detection using Random Sets of Deep Convolutional Neural Network Observations. *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, 17(0 1):520–527, 2014.

[172] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 23(3):309–314, 2004.

[173] Tomas Sakinis, Fausto Milletari, Holger Roth, Panagiotis Korfiatis, Petro Kostandy, Kenneth Philbrick, Zeynettin Akkus, Ziyue Xu, Daguang Xu, and Bradley J. Erickson. Interactive segmentation of medical images through fully convolutional neural networks. March 2019.

[174] Luís Santos, Filipe N. Santos, Paulo Moura Oliveira, and Pranjali Shinde. Deep Learning Applications in Agriculture: A Short Review. In Manuel F. Silva, José Luís Lima, Luís Paulo Reis, Alberto Sanfeliu, and Danilo Tardioli, editors, *Robot 2019: Fourth Iberian Robotics Conference*, Advances in Intelligent Systems and Computing, pages 139–151, Cham, 2020. Springer International Publishing. ISBN 978-3-030-35990-4. doi: 10.1007/978-3-030-35990-4_12.

[175] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 253–260, New York, NY, USA, August 2002. Association for Computing Machinery. ISBN 978-1-58113-561-9. doi: 10.1145/564376.564421.

[176] Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, Benjamin Schmid, Jean-Yves Tinevez, Daniel James White, Volker Hartenstein, Kevin Eliceiri, Pavel Tomancak,

and Albert Cardona. Fiji: An open-source platform for biological-image analysis. *Nature Methods*, 9(7):676–682, July 2012. ISSN 1548-7105. doi: 10.1038/nmeth.2019.

[177] Jan Schreier, Francesca Attanasi, and Hannu Laaksonen. Generalization vs. Specificity: In Which Cases Should a Clinic Train its Own Segmentation Models? *Frontiers in Oncology*, 10, 2020. ISSN 2234-943X. doi: 10.3389/fonc.2020.00675.

[178] Pola Schwöbel, Frederik Warburg, Martin Jørgensen, Kristoffer H. Madsen, and Søren Hauberg. Probabilistic Spatial Transformers for Bayesian Data Augmentation. *arXiv:2004.03637 [cs, stat]*, April 2020.

[179] Anand Seethepalli, Kundan Dhakal, Marcus Griffiths, Haichao Guo, Gregoire T Freschet, and Larry M York. RhizoVision Explorer: Open-source software for root image analysis and measurement standardization. *AoB PLANTS*, 13(6):plab056, December 2021. ISSN 2041-2851. doi: 10.1093/aobpla/plab056.

[180] Boris Sekachev, Nikita Manovich, Maxim Zhiltsov, Andrey Zhavoronkov, Dmitry Kalinin, Ben Hoff, TOsmanov, Dmitry Kruchinin, Artyom Zankevich, DmitriySidnev, Maksim Markelov, Johannes222, Mathis Chenuet, a andre, telenachos, Aleksandr Melnikov, Jijoong Kim, Liron Ilouz, Nikita Glazov, Priya4607, Rush Tehrani, Seungwon Jeong, Vladimir Skubriev, Sebastian Yonekura, vugia truong, zliang7, lizhming, and Tritin Truong. opencv/cvat: v1.1.0, August 2020. URL `https://doi.org/10.5281/zenodo.4009388`.

[181] Marili Sell, Abraham George Smith, Iuliia Burdun, Gristin Rohula-Okunev, Priit Kupper, and Ivika Ostonen. Assessing the fine root growth dynamics of Norway spruce manipulated by air humidity and soil nitrogen with deep learning segmentation of smartphone images. *Plant and Soil*, 480(1):135–150, November 2022. ISSN 1573-5036. doi: 10.1007/s11104-022-05565-4.

[182] Raghavendra Selvan, Erik Dam, Nicki Skafte Detlefsen, Sofus Rischel, Kaining Sheng, Mads Nielsen, and Akshay Pai. Lung Segmentation from Chest X-rays using Variational Data Imputation. In *ICML Workshop on the Art of Learning with Missing Values (Artemiss)*, June 2022.

[183] Rico Sennrich, Barry Haddow, and Alexandra Birch. Edinburgh Neural Machine Translation Systems for WMT 16. *arXiv:1606.02891 [cs]*, June 2016.

[184] Burr Settles. Active Learning Literature Survey. Technical Report, University of Wisconsin-Madison Department of Computer Sciences, 2009. URL `https://minds.wisconsin.edu/handle/1793/60660`.

[185] Fahad Shamshad, Salman Khan, Syed Waqas Zamir, Muhammad Haris Khan, Munawar Hayat, Fahad Shahbaz Khan, and Huazhu Fu. Transformers in Medical Imaging: A Survey, January 2022.

[186] Patrik Sibolt, Lina M. Andersson, Lucie Calmels, David Sjöström, Ulf Bjelkengren, Poul Geertsen, and Claus F. Behrens. Clinical implementation of artificial intelligence-driven cone-beam computed tomography-guided online adaptive radiotherapy in the pelvic region. 17:1–7. ISSN 24056316. doi: 10.1016/j.phro.2020.12.004. URL `https://linkinghub.elsevier.com/retrieve/pii/S2405631620300816`.

[187] Amber L. Simpson, Michela Antonelli, Spyridon Bakas, Michel Bilello, Keyvan Farahani, Bram van Ginneken, Annette Kopp-Schneider, Bennett A. Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M. Summers, Patrick Bilic, Patrick F. Christ, Richard K. G. Do, Marc Gollub, Jennifer Golia-Pernicka, Stephan H. Heckers, William R. Jarnagin, Maureen K. McHugo, Sandy Napel, Eugene Vorontsov, Lena Maier-Hein, and M. Jorge Cardoso. A large annotated medical image dataset for the development and evaluation of segmentation algorithms, February 2019.

[188] Erik Smistad, Andreas Østvik, and Lasse Løvstakken. Annotation web - an open-source web-based annotation tool for ultrasound images. In *2021 IEEE International Ultrasonics Symposium (IUS)*, pages 1–4, 2021. doi: 10.1109/IUS52206.2021.9593336.

[189] A. Smith, J. Petersen, I. Wahlstedt, S.L. Risumlund, M.V.O. Felter, V.N. Hansen, and I.R. Vogelius. PD-0065 Corrective-annotation autocompletion enables faster organ contouring. *Radiotherapy and Oncology*, 170:S38–S39, May 2022. ISSN 01678140. doi: 10.1016/S0167-8140(22)02735-9.

[190] Abraham George Smith, Jens Petersen, Raghavendra Selvan, and Camilla Ruø Rasmussen. Data for paper 'Segmentation of Roots in Soil with U-Net', November 2019.

[191] Abraham George Smith, Jens Petersen, Raghavendra Selvan, and Camilla Ruø Rasmussen. Trained U-Net Model for paper 'Segmentation of Roots in Soil with U-Net'. October 2019. doi: 10.5281/zenodo.3484015.

[192] Abraham George Smith, Eusun Han, Jens Petersen, Niels Alvin Faircloth Olsen, Christian Giese, Miriam Athmann, Dorte Bodin Dresbøll, and Kristian Thorup-Kristensen. Counted Nodules dataset used in 'RootPainter: Deep Learning Segmentation of Biological Images with Corrective Annotation', April 2020.

[193] Abraham George Smith, Eusun Han, Jens Petersen, Niels Alvin Faircloth Olsen, Christian Giese, Miriam Athmann, Dorte Bodin Dresbøll,

and Kristian Thorup-Kristensen. RootPainter: Deep Learning Segmentation of Biological Images with Corrective Annotation. *bioRxiv*, page 2020.04.16.044461, May 2020. doi: 10.1101/2020.04.16.044461.

[194] Abraham George Smith, Jens Petersen, Raghavendra Selvan, and Camilla Ruø Rasmussen. Segmentation of roots in soil with U-Net. *Plant Methods*, 16(1):13, February 2020. ISSN 1746-4811. doi: 10.1186/s13007-020-0563-0.

[195] Abraham George Smith, Eusun Han, Jens Petersen, Niels Alvin Faircloth Olsen, Christian Giese, Miriam Athmann, Dorte Bodin Dresbøll, and Kristian Thorup-Kristensen. RootPainter: Deep learning segmentation of biological images with corrective annotation. *New Phytologist*, 236 (2):774–791, 2022. ISSN 1469-8137. doi: 10.1111/nph.18387.

[196] Abraham George Smith, Jens Petersen, Cynthia Terrones-Campos, Anne Kiil Berthelsen, Nora Jarrett Forbes, Sune Darkner, Lena Specht, and Ivan Richter Vogelius. RootPainter3D: Interactive-machine-learning enables rapid and accurate contouring for radiotherapy. *Medical Physics*, 49(1):461–473, 2022. ISSN 2473-4209. doi: 10.1002/mp.15353.

[197] Abraham George Smith, Denis Kutnár, Ivan Richter Vogelius, Sune Darkner, and Jens Petersen. Localise to segment: Crop to improve organ at risk segmentation accuracy. 2023. doi: 10.48550/ARXIV.2304.04606.

[198] Abraham Goerge Smith, Eusun Han, Jens Petersen, Niels Alvin Faircloth Olsen, Christian Giese, Miriam Athmann, Dorte Bodin Dresbøll, and Kristian Thorup-Kristensen. Counted Biopores dataset used in 'RootPainter: Deep Learning Segmentation of Biological Images with Corrective Annotation', April 2020.

[199] Abraham Goerge Smith, Eusun Han, Jens Petersen, Niels Alvin Faircloth Olsen, Christian Giese, Miriam Athmann, Dorte Bodin Dresbøll, and Kristian Thorup-Kristensen. Training datasets and final models from paper "RootPainter: Deep Learning Segmentation of Biological Images with Corrective Annotation', April 2020.

[200] Leslie N. Smith. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay. *arXiv:1803.09820 [cs, stat]*, April 2018.

[201] Nicholas Sofroniew, Talley Lambert, Kira Evans, Juan Nunez-Iglesias, Grzegorz Bokota, Philip Winston, Gonzalo Peña-Castellanos, Kevin Yamauchi, Matthias Bussonnier, Draga Doncila Pop, Ahmet Can Solak, Ziyang Liu, Pam Wadhwa, Alister Burt, Genevieve Buckley, Andrew Sweet, Lukasz Migas, Volker Hilsenstein, Lorenzo Gaifas, Jordão Bragantini, Jaime Rodríguez-Guerra, Hector Muñoz, Jeremy Freeman, Peter

Boone, Alan Lowe, Christoph Gohlke, Loic Royer, Andrea PIERRÉ, Hagai Har-Gil, and Abigail McGovern. Napari: A multi-dimensional image viewer for Python. Zenodo, May 2022.

[202] Mohammadreza Soltaninejad, Craig J. Sturrock, Marcus Griffiths, Tony P. Pridmore, and Michael P. Pound. Three Dimensional Root CT Segmentation using Multi-Resolution Encoder-Decoder Networks. *bioRxiv*, page 713859, July 2019. doi: 10.1101/713859.

[203] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The Implicit Bias of Gradient Descent on Separable Data. *arXiv:1710.10345 [cs, stat]*, December 2018.

[204] Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sébastien Ourselin, and M. Jorge Cardoso. Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations. *arXiv:1707.03237 [cs]*, 10553: 240–248, 2017. doi: 10.1007/978-3-319-67558-9_28.

[205] Simon Fiil Svane, Christian Sig Jensen, and Kristian Thorup-Kristensen. Construction of a large-scale semi-field facility to study genotypic differences in deep root growth and resources acquisition. *Plant Methods*, 15 (1):26, March 2019. ISSN 1746-4811. doi: 10.1186/s13007-019-0409-9.

[206] Saeid Asgari Taghanaki, Yefeng Zheng, S. Kevin Zhou, Bogdan Georgescu, Puneet Sharma, Daguang Xu, Dorin Comaniciu, and Ghassan Hamarneh. Combo Loss: Handling Input and Output Imbalance in Multi-Organ Segmentation. *arXiv:1805.02798 [cs]*, October 2018.

[207] Sarah Taghavi Namin, Mohammad Esmaeilzadeh, Mohammad Najafi, Tim B. Brown, and Justin O. Borevitz. Deep phenotyping: Deep learning for temporal phenotype/genotype classification. *Plant Methods*, 14(1):66, August 2018. ISSN 1746-4811. doi: 10.1186/s13007-018-0333-4.

[208] Hao Tang, Xuming Chen, Yang Liu, Zhipeng Lu, Junhua You, Mingzhou Yang, Shengyu Yao, Guoqi Zhao, Yi Xu, Tingfeng Chen, Yong Liu, and Xiaohui Xie. Clinically applicable deep learning framework for organs at risk delineation in CT images. *Nature Machine Intelligence*, 1(10):480–491, October 2019. ISSN 2522-5839. doi: 10.1038/s42256-019-0099-z.

[209] Agnes Tegen, Paul Davidsson, and Jan A. Persson. Activity recognition through interactive machine learning in a dynamic sensor setting. *Personal and Ubiquitous Computing*, June 2020. ISSN 1617-4917. doi: 10.1007/s00779-020-01414-2.

[210] Agnes Tegen, Paul Davidsson, and Jan A. Persson. Active Learning and Machine Teaching for Online Learning: A Study of Attention and Labelling Cost. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1215–1220, December 2021. doi: 10.1109/ICMLA52953.2021.00197.

[211] Javier Tello, María Ignacia Montemayor, Astrid Forneck, and Javier Ibáñez. A new image-based tool for the high throughput phenotyping of pollen viability: Evaluation of inter- and intra-cultivar diversity in grapevine. *Plant Methods*, 14(1):3, January 2018. ISSN 1746-4811. doi: 10.1186/s13007-017-0267-2.

[212] C. Terrones-Campos, B. Ledergerber, N. Forbes, A. G. Smith, J. Petersen, M. Helleberg, J. Lundgren, L. Specht, and I. R. Vogelius. Prediction of Radiation-induced Lymphopenia following Exposure of the Thoracic Region and Associated Risk of Infections and Mortality. *Clinical Oncology*, April 2023. ISSN 0936-6555. doi: 10.1016/j.clon.2023.04.003.

[213] Cynthia Terrones-Campos, Bruno Ledergerber, Nora Forbes, Abraham George Smith, Marie Helleberg, Jens Lundgren, Lena Specht, and Ivan Richter Vogelius. Prediction of radiation-induced lymphopenia following exposure of the thoracic region and associated risk of infections and mortality. *submitted*.

[214] Cynthia Terrones-Campos, Bruno Ledergerber, Ivan Richter Vogelius, Marie Helleberg, Lena Specht, and Jens Lundgren. Hematological toxicity in patients with solid malignant tumors treated with radiation – Temporal analysis, dose response and impact on survival. *Radiotherapy and Oncology*, 158:175–183, May 2021. ISSN 01678140. doi: 10.1016/j.radonc.2021.02.029.

[215] Kristian Thorup-Kristensen. Are differences in root growth of nitrogen catch crops important for their ability to reduce soil nitrate-N content, and how can this be measured? *Plant and Soil*, 230(2):185–195, March 2001. ISSN 1573-5036. doi: 10.1023/A:1010306425468.

[216] Kristian Thorup-Kristensen. Effect of deep and shallow root systems on the dynamics of soil inorganic N during 3-year crop rotations. *Plant and Soil*, 288(1-2):233–248, October 2006. ISSN 0032-079X, 1573-5036. doi: 10.1007/s11104-006-9110-7.

[217] Kristian Thorup-Kristensen and John Kirkegaard. Root system-based limits to agricultural productivity and efficiency: The farming systems context. *Annals of Botany*, 118(4):573–592, October 2016. ISSN 0305-7364. doi: 10.1093/aob/mcw122.

[218] Kristian Thorup-Kristensen, Dorte Bodin Dresbøll, and Hanne L. Kristensen. Crop yield, root growth, and nutrient dynamics in a conventional and three organic cropping systems with different levels of external inputs and N re-cycling through fertility building crops. *European Journal of Agronomy*, 37(1):66–82, February 2012. ISSN 1161-0301. doi: 10.1016/j.eja.2011.11.004.

[219] Kristian Thorup-Kristensen, Niels Halberg, Mette Nicolaisen, Jørgen Eivind Olesen, Timothy E. Crews, Philippe Hinsinger, John

Kirkegaard, Alain Pierret, and Dorte Bodin Dresbøll. Digging Deeper for Agricultural Resources, the Value of Deep Rooting. *Trends in Plant Science*, 25(4):406–417, April 2020. ISSN 1360-1385. doi: 10.1016/j.tplants.2019.12.007.

[220] Kristian Thorup-Kristensen, Niels Halberg, Mette H. Nicolaisen, Jørgen E. Olesen, and Dorte Bodin Dresbøll. Exposing Deep Roots: A Rhizobox Laboratory. *Trends in Plant Science*, 25(4):418–419, April 2020. ISSN 1360-1385. doi: 10.1016/j.tplants.2019.12.006.

[221] Mariya Toneva and Alessandro Sordoni. AN EMPIRICAL STUDY OF EXAMPLE FORGETTING DURING DEEP NEURAL NETWORK LEARNING. In *International Conference on Learning Representations*, page 18, New Orleans, Louisiana, United States, 2019.

[222] Jordan Ubbens, Mikolaj Cieslak, Przemyslaw Prusinkiewicz, and Ian Stavness. The use of plant models in deep learning: An application to leaf counting in rosette plants. *Plant Methods*, 14(1):1–10, January 2018. ISSN 1746-4811. doi: 10.1186/s13007-018-0273-z.

[223] Jordan R. Ubbens and Ian Stavness. Deep Plant Phenomics: A Deep Learning Platform for Complex Plant Phenotyping Tasks. *Frontiers in Plant Science*, 8, 2017. ISSN 1664-462X. doi: 10.3389/fpls.2017.01190.

[224] Hyemin Um, Jue Jiang, Maria Thor, Andreas Rimner, Leo Luo, Joseph O. Deasy, and Harini Veeraraghavan. Multiple resolution residual network for automatic thoracic organs-at-risk segmentation from CT. *arXiv:2005.13690 [cs, eess]*, May 2020.

[225] Shailee Upadhyay, Abraham George Smith, Dirk Vandepitte, Stepan Vladimirovitch Lomov, Yentl Swolfs, and Mahoor Mehdikhani. Analysis of voids in filament wound composites using a machine-learning based segmentation tool. In *Proceedings of the 20th European Conference on Composite Materials (ECCM20)*, February 2022.

[226] Femke Vaassen, Colien Hazelaar, Ana Vaniqui, Mark Gooding, Brent van der Heyden, Richard Canters, and Wouter van Elmpt. Evaluation of measures for assessing time-saving of automatic organ-at-risk segmentation in radiotherapy. *Physics and Imaging in Radiation Oncology*, 13:1–6, January 2020. ISSN 2405-6316. doi: 10.1016/j.phro.2019.12.001.

[227] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. Scikit-image: Image processing in Python. *PeerJ*, 2: e453, June 2014. ISSN 2167-8359. doi: 10.7717/peerj.453.

[228] Louis D van Harten, Julia M H Noothout, Joost J C Verhoeff, Jelmer M Wolterink, and Ivana Is˘gum. AUTOMATIC SEGMENTATION OF ORGANS AT RISK IN THORACIC CT SCANS BY COMBINING 2D AND 3D CONVOLUTIONAL NEURAL NETWORKS.

[229] Vladimir N. Vapnik. Bounds on the rate of convergence of learning processes. In *The Nature of Statistical Learning Theory*, pages 69–91. Springer New York, New York, NY, 2000. ISBN 978-1-4757-3264-1. doi: 10.1007/978-1-4757-3264-1_4.

[230] Manasi Vartak, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, and Hugo Larochelle. A Meta-Learning Perspective on Cold-Start Recommendations for Items. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[231] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, March 2020. ISSN 1548-7105. doi: 10.1038/s41592-019-0686-2.

[232] Ivan R. Vogelius, Jens Petersen, and Søren M. Bentzen. Harnessing data science to advance radiation oncology. *Molecular Oncology*, 14(7):1514–1528, 2020. ISSN 1878-0261. doi: 10.1002/1878-0261.12685.

[233] Kerstin N. Vokinger, Stefan Feuerriegel, and Aaron S. Kesselheim. Continual learning in medical devices: FDA's action plan and beyond. *The Lancet Digital Health*, 0(0), April 2021. ISSN 2589-7500. doi: 10.1016/S2589-7500(21)00076-5.

[234] Minh H. Vu, Guus Grimbergen, Tufve Nyholm, and Tommy Löfstedt. Evaluation of Multi-Slice Inputs to Convolutional Neural Networks for Medical Image Segmentation. *arXiv:1912.09287 [cs, eess, stat]*, December 2019.

[235] Minh H. Vu, Guus Grimbergen, Tufve Nyholm, and Tommy Löfstedt. Evaluation of multislice inputs to convolutional neural networks for medical image segmentation. *Medical Physics*, 47(12):6216–6231, 2020. ISSN 2473-4209. doi: 10.1002/mp.14391.

[236] Isak Wahlstedt, Abraham George Smith, Claus E. Andersen, Claus P. Behrens, Susanne Nørring Bekke, Kristian Boye, Mette van Overeem Felter, Mirjana Josipovic, Jens Petersen, Signe Lenora Risumlund, José D. Tascón-Vidarte, Janita E. van Timmeren, and Ivan R. Vogelius. Title: Interfractional Dose Accumulation for MR-guided Liver SBRT: Variation Among Algorithms is Highly Patient- and Fraction-dependent. *Radiotherapy and Oncology*, 0(0), December 2022. ISSN 0167-8140, 1879-0887. doi: 10.1016/j.radonc.2022.109448.

[237] James Walter, James Edwards, Jinhai Cai, Glenn McDonald, Stanley J. Miklavcic, and Haydn Kuchel. High-Throughput Field Imaging and Basic Image Analysis in a Wheat Breeding Programme. *Frontiers in Plant Science*, 10, 2019. ISSN 1664-462X. doi: 10.3389/fpls.2019.00449.

[238] Haoyu Wang, Haiyu Song, Haiyan Wu, Zhiqiang Zhang, Shengchun Deng, Xiaoqing Feng, and Yanhong Chen. Multilayer feature fusion and attention-based network for crops and weeds segmentation. *Journal of Plant Diseases and Protection*, 129(6):1475–1489, December 2022. ISSN 1861-3829, 1861-3837. doi: 10.1007/s41348-022-00663-y.

[239] Kyle Wang and Joel E. Tepper. Radiation therapy-associated toxicity: Etiology, management, and prevention. *CA: A Cancer Journal for Clinicians*, 71(5):437–454, 2021. ISSN 1542-4863. doi: 10.3322/caac.21689.

[240] Jakob Wasserthal, Manfred Meyer, Hanns-Christian Breit, Joshy Cyriac, Shan Yang, and Martin Segeroth. TotalSegmentator: Robust segmentation of 104 anatomical structures in CT images, August 2022.

[241] Zixiang Wei, Jintao Ren, Stine Sofia Korreman, and Jasper Nijkamp. Towards interactive deep-learning for tumour segmentation in head and neck cancer radiotherapy. *Physics and Imaging in Radiation Oncology*, 25: 100408, January 2023. ISSN 2405-6316. doi: 10.1016/j.phro.2022.12.005.

[242] Dennis Winkel, Gijsbert H. Bol, Petra S. Kroon, Bram van Asselen, Sara S. Hackett, Anita M. Werensteijn-Honingh, Martijn P.W. Intven, Wietse S.C. Eppinga, Rob H.N. Tijssen, Linda G.W. Kerkmeijer, Hans C.J. de Boer, Stella Mook, Gert J. Meijer, Jochem Hes, Mirjam Willemsen-Bosman, Eline N. de Groot-van Breugel, Ina M. Jürgenliemk-Schulz, and Bas W. Raaymakers. Adaptive radiotherapy: The elekta unity MR-linac concept. 18:54–59. ISSN 24056308. doi: 10.1016/j.ctro.2019.04.001. URL `https://linkinghub.elsevier.com/retrieve/pii/S2405630819300631`.

[243] Yinhao Wu, Bin Chen, An Zeng, Dan Pan, Ruixuan Wang, and Shen Zhao. Skin Cancer Classification With Deep Learning: A Systematic Review. *Frontiers in Oncology*, 12, 2022. ISSN 2234-943X.

[244] Yuxin Wu and Kaiming He. Group Normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.

[245] Weihuang Xu, Guohao Yu, Alina Zare, Brendan Zurweller, Diane Rowland, Joel Reyes-Cabrera, Felix B. Fritschi, Roser Matamala, and Thomas E. Juenger. Overcoming Small Minirhizotron Datasets Using Transfer Learning. *arXiv:1903.09344 [cs]*, March 2019.

[246] X. Xu, F. Zhou, B. Liu, and X. Bai. Multiple Organ Localization in CT Image Using Triple-Branch Fully Convolutional Networks. *IEEE Access*, 7:98083–98093, 2019. ISSN 2169-3536. doi: 10.1109/ACCESS.2019.2930417.

[247] Xuanang Xu, Fugen Zhou, Bo Liu, Dongshan Fu, and Xiangzhi Bai. Efficient Multiple Organ Localization in CT Image using 3D Region Proposal Network. *IEEE transactions on medical imaging*, January 2019. ISSN 1558-254X. doi: 10.1109/TMI.2019.2894854.

[248] Zhanyou Xu, Larry M. York, Anand Seethepalli, Bruna Bucciarelli, Hao Cheng, and Deborah A. Samac. Objective Phenotyping of Root System Architecture Using Image Augmentation and Machine Learning in Alfalfa (Medicago sativa L.). *Plant Phenomics*, 2022, April 2022. doi: 10.34133/2022/9879610.

[249] Jiancheng Yang, Xiaoyang Huang, Bingbing Ni, Jingwei Xu, Canqian Yang, and Guozheng Xu. Reinventing 2D Convolutions for 3D Images. *arXiv:1911.10477 [cs, eess]*, March 2020.

[250] Jinzhong Yang, Chuanming Wei, Lifei Zhang, Yongbin Zhang, Rick S. Blum, and Lei Dong. A statistical modeling approach for evaluating auto-segmentation methods for image-guided radiotherapy. *Computerized Medical Imaging and Graphics*, 36(6):492–500, September 2012. ISSN 0895-6111. doi: 10.1016/j.compmedimag.2012.05.001.

[251] Jinzhong Yang, Harini Veeraraghavan, Samuel G. Armato, Keyvan Farahani, Justin S. Kirby, Jayashree Kalpathy-Kramer, Wouter van Elmpt, Andre Dekker, Xiao Han, Xue Feng, Paul Aljabar, Bruno Oliveira, Brent van der Heyden, Leonid Zamdborg, Dao Lam, Mark Gooding, and Gregory C. Sharp. Autosegmentation for thoracic radiation treatment planning: A grand challenge at AAPM 2017. *Medical Physics*, 45(10):4568–4581, October 2018. ISSN 2473-4209. doi: 10.1002/mp.13141.

[252] Robail Yasrab, Jonathan A Atkinson, Darren M Wells, Andrew P French, Tony P Pridmore, and Michael P Pound. RootNav 2.0: Deep learning for automatic navigation of complex plant root architectures. *GigaScience*, 8 (11):giz123, November 2019. ISSN 2047-217X. doi: 10.1093/gigascience/giz123.

[253] Suk Whan Yoon, Hui Lin, Michelle Alonso-Basanta, Nate Anderson, Ontida Apinorasethkul, Karima Cooper, Lei Dong, Brian Kempsey, Jaclyn Marcel, James Metz, Ryan Scheuermann, and Taoran Li. Initial evaluation of a novel cone-beam CT-based semi-automated online adaptive radiotherapy system for head and neck cancer treatment – a timing and automation quality study. ISSN 2168-8184. doi: 10.7759/cureus.9660. URL https://www.cureus.com/articles/35411-initial-evaluation-of-a-novel-cone-beam-ct-based-semi-automated-online-adaptive-

[254] Larry M. York, Jonathan R. Cumming, Adrianna Trusiak, Gregory Bonito, Adam C. von Haden, Udaya C. Kalluri, Lisa K. Tiemann, Peter F. Andeer, Elena Blanc-Betes, Jonathan H. Diab, Alonso Favela, Amandine Germon, Nuria Gomez-Casanovas, Charles A. Hyde, Angela D. Kent,

Dae Kwan Ko, Austin Lamb, Ali M. Missaoui, Trent R. Northen, Yunqiao Pu, Arthur J. Ragauskas, Sierra Raglin, Henrik V. Scheller, Lorenzo Washington, and Wendy H. Yang. Bioenergy Underground: Challenges and opportunities for phenotyping roots and the microbiome for sustainable bioenergy crop production. *The Plant Phenome Journal*, 5(1):e20028, 2022. ISSN 2578-2703. doi: 10.1002/ppj2.20028.

[255] Paul A. Yushkevich, Joseph Piven, Heather Cody Hazlett, Rachel Gimpel Smith, Sean Ho, James C. Gee, and Guido Gerig. User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *Neuroimage*, 31(3):1116–1128, 2006.

[256] Guang Zeng, Stanley T. Birchfield, and Christina E. Wells. Automatic discrimination of fine roots in minirhizotron images. *New Phytologist*, 177 (2):549–557, January 2008. ISSN 1469-8137. doi: 10.1111/j.1469-8137.2007.02271.x.

[257] Nico Zettler and Andre Mastmeyer. Comparison of 2D vs. 3D U-Net Organ Segmentation in abdominal 3D CT images, July 2021.

[258] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv:1611.03530 [cs]*, February 2017.

[259] Zhengxin Zhang, Qingjie Liu, and Yunhong Wang. Road Extraction by Deep Residual U-Net. *IEEE Geoscience and Remote Sensing Letters*, 15 (5):749–753, May 2018. ISSN 1545-598X, 1558-0571. doi: 10.1109/LGRS.2018.2802944.

[260] Hao Zheng, Yizhe Zhang, Lin Yang, Chaoli Wang, and Danny Z. Chen. An Annotation Sparsification Strategy for 3D Medical Image Segmentation via Representative Selection and Self-Training. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):6925–6932, April 2020. ISSN 2374-3468. doi: 10.1609/aaai.v34i04.6175.

[261] Xiangrong Zhou, Kuzuma Yamada, Ryosuke Takayama, Xinxin Zhou, Takeshi Hara, Hiroshi Fujita, Song Wang, and Takuya Kojima. Performance evaluation of 2D and 3D deep learning approaches for automatic segmentation of multiple organs on CT images. In Kensaku Mori and Nicholas Petrick, editors, *Medical Imaging 2018: Computer-Aided Diagnosis*, page 83, Houston, United States, February 2018. SPIE. ISBN 978-1-5106-1639-4 978-1-5106-1640-0. doi: 10.1117/12.2295178.

[262] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. UNet++: A Nested U-Net Architecture for Medical Image Segmentation, July 2018.

[263] Kunlin Zou, Xin Chen, Yonglin Wang, Chunlong Zhang, and Fan Zhang. A modified U-Net with a specific data argumentation method for semantic

segmentation of weed images in the field. *Computers and Electronics in Agriculture*, 187:106242, August 2021. ISSN 01681699. doi: 10.1016/j.compag.2021.106242.