MONAI Label: A framework for AI-assisted Interactive Labeling of 3D Medical Images

Andres Diaz-Pinto[a,b,*], Sachidanand Alle[b], Alvin Ihsani[b], Muhammad Asad[a], Vishwesh Nath[b], Fernando Pérez-García[a,c], Pritesh Mehta[a,c], Wenqi Li[b], Holger R. Roth[b], Tom Vercauteren[a], Daguang Xu[b], Prerna Dogra[b], Sebastien Ourselin[a], Andrew Feng[b], M. Jorge Cardoso[a]

[a]*School of Biomedical Engineering & Imaging Sciences, King's College London. London, UK.*
[b]*NVIDIA Santa Clara, CA, USA.*
[c]*Department of Medical Physics and Biomedical Engineering, University College London. London, UK.*

Abstract

The lack of annotated datasets is a major challenge in training new task-specific supervised AI algorithms as manual annotation is expensive and time-consuming. To address this problem, we present MONAI Label, a free and open-source platform that facilitates the development of AI-based applications that aim at reducing the time required to annotate 3D medical image datasets. Through MONAI Label researchers can develop annotation applications focusing on their domain of expertise. It allows researchers to readily deploy their apps as services, which can be made available to clinicians via their preferred user-interface. Currently, MONAI Label readily supports locally installed (3DSlicer) and web-based (OHIF) frontends, and offers two Active learning strategies to facilitate and speed up the training of segmentation algorithms. MONAI Label allows researchers to make incremental improvements to their labeling apps by making them available to other researchers and clinicians alike. Lastly, MONAI Label provides sample labeling apps, namely DeepEdit and DeepGrow, demonstrating dramatically reduced annotation times.

Keywords: 3D Medical imaging, Interactive 3D image segmentation, Active Learning, Deep Learning

Introduction

Image segmentation and quantitative analysis of medical data, especially of 3D volumes, are tedious and time-consuming tasks. The lack of expertly annotated datasets is one of the primary bottlenecks when developing new supervised segmentation deep learning algorithms. Non-expert manual annotation is time-consuming and cost-ineffective, and even with the training of relatively large groups of annotators, resulting labels are often less than ideal. Conversely, expert-led annotations (e.g. clinicians, anatomists) result in higher quality labels, but due to time availability and cost, the number and variety of annotated samples suffer.

Deep learning algorithms based on CNN are the current state of the art for automatic 2D and 3D medical image segmentation [1, 2, 3], inspired by the landmark contributions of [4] (2D U-Net), [5] (3D U-Net), and [6] (V-Net). The *MSD* [7] is driving methodological innovations and performance improvements. At the time of writing, second position on the live leaderboard[1] is held by nnU-Net [1], a segmentation pipeline based on U-Net that automatically configures to any new medical image segmentation task, and first position is held by the DiNTS scheme [2], which allows architectures not confined to pre-defined topologies. A recent work by [3] introduces UNETR, which uses a transformer [8] as the encoder, achieving favorable benchmarks on the *MSD* brain tumor and spleen segmentation tasks. Despite state-of-the-art performance on several medical image segmentation tasks, automatic segmentation algorithms have not yet reached the desired robustness to allow clinical use [9]. In par-

---

*Corresponding author
Email address: andres.diaz-pinto@kcl.ac.uk (Andres Diaz-Pinto)

[1]https://decathlon-10.grand-challenge.org/evaluation/challenge/leaderboard/

(a) Input Volume   **(b) Initial Segmentation**   **(c) Scribbles-based Interaction**   **(d) Clicks-based Interaction**   **(e) Final Segmentation**
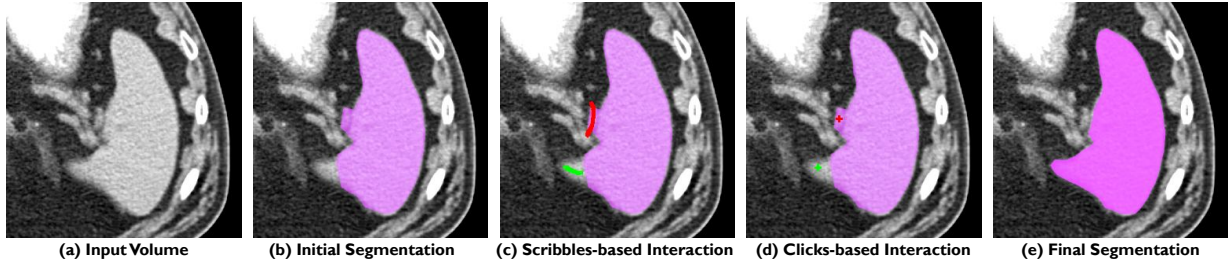
Figure 1: **Different interactive segmentation approaches offered by MONAI Label:** Shows (a) input volume, (b) initial automatic segmentation using AI, (c) scribbles-based interaction, (d) click-based interaction and (e) final segmentation incorporating user-interactions using AI-assisted annotation methods in MONAI Label.

ticular, segmentation accuracy can be impacted by patient variation, acquisition differences, and image artifacts [10].

On the other hand, interactive segmentation methods based on deep learning have been proposed for more robust natural image segmentation [11, 12]. In [11], user foreground and background clicks were converted into euclidean distance maps, and subsequently added as additional input channels to a CNN. Inspired by the aforementioned studies and other incremental works, interactive methods for medical image segmentation based on deep learning have been recently proposed [13, 14, 9, 15]. In [14], a bounding-box and scribble-based CNN segmentation pipeline was proposed, whereby an initial segmentation is obtained within a user-provided bounding-box, followed by image-specific fine-tuning using user-provided scribbles. In contrast, [13, 9] proposed a click-based method, motivated in part by the work of [11]. In their work, Gaussian-smoothed foreground and background clicks were added as input channels to an encoder-decoder CNN. Experiments on multiple-organ segmentation on CT showed that their method delivers 2D segmentations in a fast and reliable manner, generalizes well to unseen structures, and produces accurate results with few clicks. An alternate method that first performs an automatic CNN segmentation step, followed by an optional refinement through user clicks or scribbles, was proposed by [15]. Their method, named DeepI-GeoS, achieved substantially improved performance compared to automatic CNN on 2D placenta and 3D brain tumor segmentation, and higher accuracy with fewer interactions compared to traditional interactive methods.

Automatic and semi-automatic segmentation methods are available as part of open-source soft-

ware packages for medical image and biomedical image analysis. For instance, ITK-SNAP [16] offers semi-automatic active contour segmentation [17]; 3D Slicer [18] and MITK [19] offer automatic, boundary-points-based [20, 21], and Deep-Grow [9] segmentation through the NVIDIA Clara AI-Assisted Annotation Extension [22], as well as other semi-automatic segmentation methods such as region growing [23] and level sets [24].

On the other hand, biomedical image analysis tools such as ilastik [25] that facilitates the use of machine learning algorithms on tasks such as segmentation, object detection, counting and tracking. However, ilastik does not include the option of training deep convolutional networks, which puts this system in disadvantage when compared to other tools that do use deep learning algorithms. Another popular open-source platform for biomedical image analysis is BioMedisa [26, 27], an online platform developed for semi-automatic segmentation of large volumetric image. It also offers different GPU-based algorithms and weighted random walks for smart interpolation of pre-segmented slices. Biomedisa has significant advantages over CPU-based semi-automatic segmentation tools for biomedical image analysis. It was specifically developed to work on a cluster or parallel computer architectures. This means, using Biomedisa with a single and small GPU is not always easy.

In addition to open-source software packages, automatic and semi-automatic methods for annotating images are available in commercial solutions such as SuperAnnotate[2], V7 Labs[3] and Segments[4]. However, the details of the algorithms behind these

---

[2]https://superannotate.com/
[3]https://www.v7labs.com/
[4]https://segments.ai

platforms are not open-source, which makes them less attractive to the scientific community.

MONAI Label is a free and intelligent open-source image labeling and learning tool that enables users to create annotated datasets and build AI-based annotation models for clinical evaluation. It enables researchers to build, train and perform inference using labeling apps in a serverless way, where custom labeling apps are exposed as a service through the MONAI Label Server. MONAI Label currently offers three segmentation approaches: two interactive (DeepEdit and DeepGrow) and one non-interactive (Automatic Segmentation) which is based on any type of UNet. In addition to this, it provides scribbles-based interactive segmentation methods that rely on energy-based optimization to propose or refine segmentation labels [28, 14, 29]. MONAI Label also includes a heuristic planner, an algorithm that enables better utilization of available GPU hardware by proposing the best configurations possible for training/inferring from an AI-based model. The heuristic planner can be used in conjunction with any available AI-based annotation techniques. An overview of the different interactive segmentation techniques offered by MONAI Label is shown in Figure 1.

Similar to popular annotation platforms for 2D images (Visual Object Tagging Tool (VoTT)[5] and Computer Vision Annotation Tool (CVAT)[6]), MONAI Label offers Active learning strategies to facilitate the training of deep learning algorithm for 3D medical image segmentation. Specifically, it computes aleatoric and epistemic uncertainty values to rank unlabeled images, allowing the clinician/user to segment the harder samples first. It also supports the DICOMweb Standard for web-based medical imaging, enabling users to integrate MONAI Label in their Picture Archiving and Communication System (PACS), XNAT [30], Image Data Commons (IDC) [31] or any other DICOM system. This feature allows web developers to unlock the power of healthcare images using industry-standard toolsets. Additionally, MONAI Label support two popular graphical user interfaces (GUIs): 3DSlicer and OHIF. See Figure 2.

Documentation and MONAI Label application examples can be found at `https://pypi.org/project/monailabel/`. It can be installed from the Python Package Index (PyPI) by running **pip install monailabel**, which also installs the pre-built OHIF viewer [32].

1. Annotation Approaches

MONAI Label currently employs four annotation approaches: three interactive (DeepGrow, DeepEdit, Scribbles-based) and one non-interactive (Automatic Segmentation). Both DeepGrow and DeepEdit guide the segmentation with positive and negative clicks provided by the user. Similarly, the scribbles-based methods utilize foreground and background scribbles provided by the user. On the other hand, the automatic approach allows the researcher to create a segmentation pipeline using a segmentation network implemented in MONAI or the underlying PyTorch framework (i.e. UNet, Highresnet, ResNet [33], DynUnet [34], etc) to automatically segment images.

1.1. DeepGrow

DeepGrow is an interactive segmentation model where the user guides the segmentation with positive and negative clicks. The positive clicks, inside the region of interest, expand the segmentation to include that location, while the negative clicks are used for contracting the segmentation to exclude the clicked region from the region of interest [9].

The Training process of a DeepGrow model differs from traditional deep learning segmentation due to a simulation process of positive and negative guidance (clicks) involved in the training process. The positive and negative guidance maps are generated based on the false negatives and false positives which are dependent on the predictions. Both DeepGrow 2D and 3D allow the user to annotate only one label at a time. DeepGrow 2D allows the user to annotate the image one slice at a time, whereas DeepGrow 3D can annotate whole volumes.

1.2. DeepEdit

DeepEdit extends DeepGrow's click-based segmentation to allow for click-free segmentation inference and click-based segmentation editing. More specifically, it allows the user to perform inference, as a standard segmentation method (i.e. UNet), and also to interactively segment part of an image using clicks as DeepGrow does [9]. DeepEdit aims to facilitate the user experience and to facilitate the
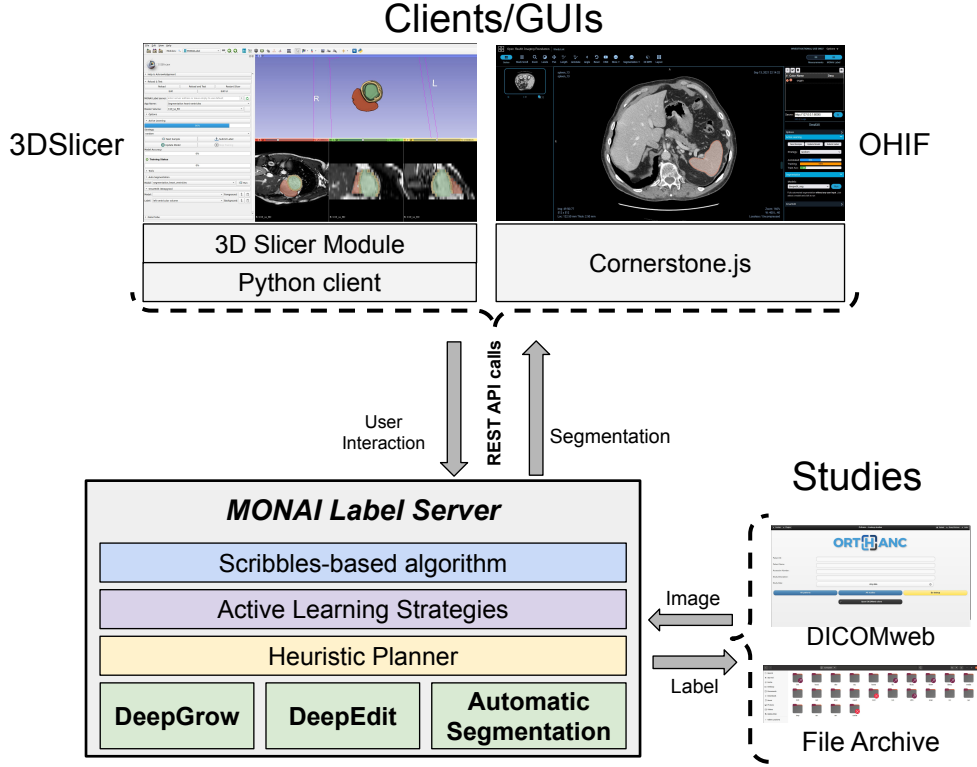

Figure 2: **Overview of the MONAI Label platform:** MONAI Label consists of three main high-level modules, namely (i) client, (ii) server, and (iii) studies. On the client side, MONAI Label supports different graphical user interfaces (GUIs) (3DSlicer/OHIF) for data viewing and annotation. On the server-side, a range of AI-assisted interactive annotation methods are provided, including click-based DeepEdit and DeepGrow and scribbles-based label refinement algorithms. The server also includes Active learning strategies and a heuristic planner for improving the efficiency of the underlying annotation methods. Studies provide a database of the dataset that is to be annotated, as well as to store the annotations.
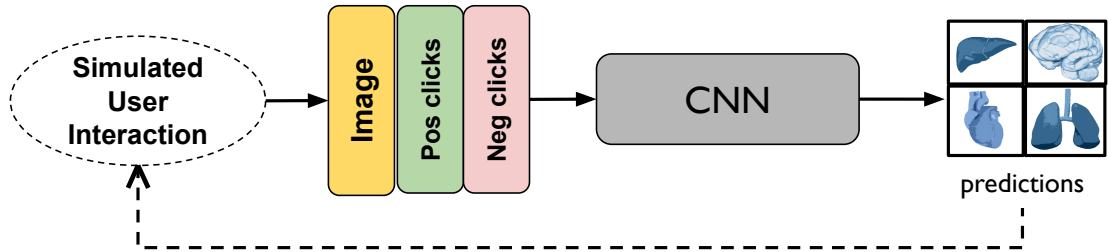

Figure 3: **Schema of the DeepGrow approach:** The input tensor consists of the image and two tensor representing positive (Pos) and negative (Neg) clicks provided by the user.

development of new active learning techniques as no user interaction (i.e. clicks) is needed to obtain an initial predicted mask, thus allowing the integration of Active Learning strategies to prioritize the labeling process.

The training process of the DeepEdit approach involves a combination of simulated clicks and standard non-interactive training. As shown in Figure 4 (Training Mode), the input of the network is a concatenation of three tensors: image, positive clicks representing the foreground and negative clicks representing the background. This model has two training stages: For half of the iterations, tensors representing the foreground and background points are zeros and for the other half, positive and negative clicks are simulated following the process presented by [9].

For automatic inference, the tensors representing positive and negative clicks are replaced by zeros. However, for interactive segmentation mode, positive and negative points/clicks provided by the user are placed in the channels accordingly. See Figure 4 (Inference Mode)

1.3. Scribbles-based Segmentation

Scribbles are free-hand drawings, such as drawing with a pen on paper, which have been widely employed to propose a range of interactive segmentation methods [14, 35, 36]. Scribbles provide natural interaction, which most annotators are already familiar with. These interactions introduce flexibility in annotators' workload, i.e. it can be as involved as required; providing both minimal interactions for simpler delineation tasks and detailed interactions for more difficult segmentations.

MONAI Label provides APIs for implementing scribbles-based interactive segmentation workflows. An overview of such workflows is presented in Figure 5.

In the figure, it can be seen that the scribbles-based interactive segmentation can be used in two different modes which are:

1. Scribbles-based online likelihood segmentation: uses scribbles to generate segmentation labels
2. Scribbles-based CNN segmentation refinement: refines segmentations from a deep learning model using user-scribbles

MONAI Label provides sample applications for both 1. and 2.

1.3.1. Scribbles-based Online Likelihood Segmentation

This approach relies on an annotator to provide scribbles to indicate regions belonging to both foreground and background objects. These scribbles are used to build an online likelihood model that enables delineation of foreground objects through likelihood prediction [29]. An energy optimization technique then refines these likelihood-based segmentations [35, 36]. The process can be repeated to provide additional user-scribbles to further refine the initial segmentations, after which the label is saved into a dataset.

In addition to scribbles, the online likelihood model can be built using a bounding box that provides the region of interest [14]. In this approach, anything outside the bounding box is selected as background scribbles, while foreground scribbles are added inside the bounding box as necessary. MONAI Label APIs provides this functionality through ROI selection tool, which can be used in conjunction with user-scribbles.

By using scribbles as interactions in an online likelihood model, this method provides a balance between fully-automatic and fully-manual segmentation methods. It is suitable for scenarios where a pre-trained deep learning model is not available, e.g. in cold start situations, as well as when starting to label a new dataset.

1.3.2. Scribbles-based CNN Segmentation Refinement

Segmentation label refinement is especially useful in cases where an unseen dataset is to be delineated using a pre-trained CNN model. In such cases, the segmentation model may not fully work in delineating all foreground regions, especially if the unseen dataset contains new variations. Scribbles-based interaction enables refining these segmentations through minimal interaction from an annotator. Hence, this approach enables significant reduction of the workload for a given annotation task.

1.4. Automated Segmentation

This MONAI Label annotation approach is essentially a non-interactive algorithm based on a standard convolutional neural network (CNN) (i.e. UNet). Researchers can use any of the available networks created in MONAI Core for their purposes. In terms of classes and interfaces, this application follows the same structure as the Deep-
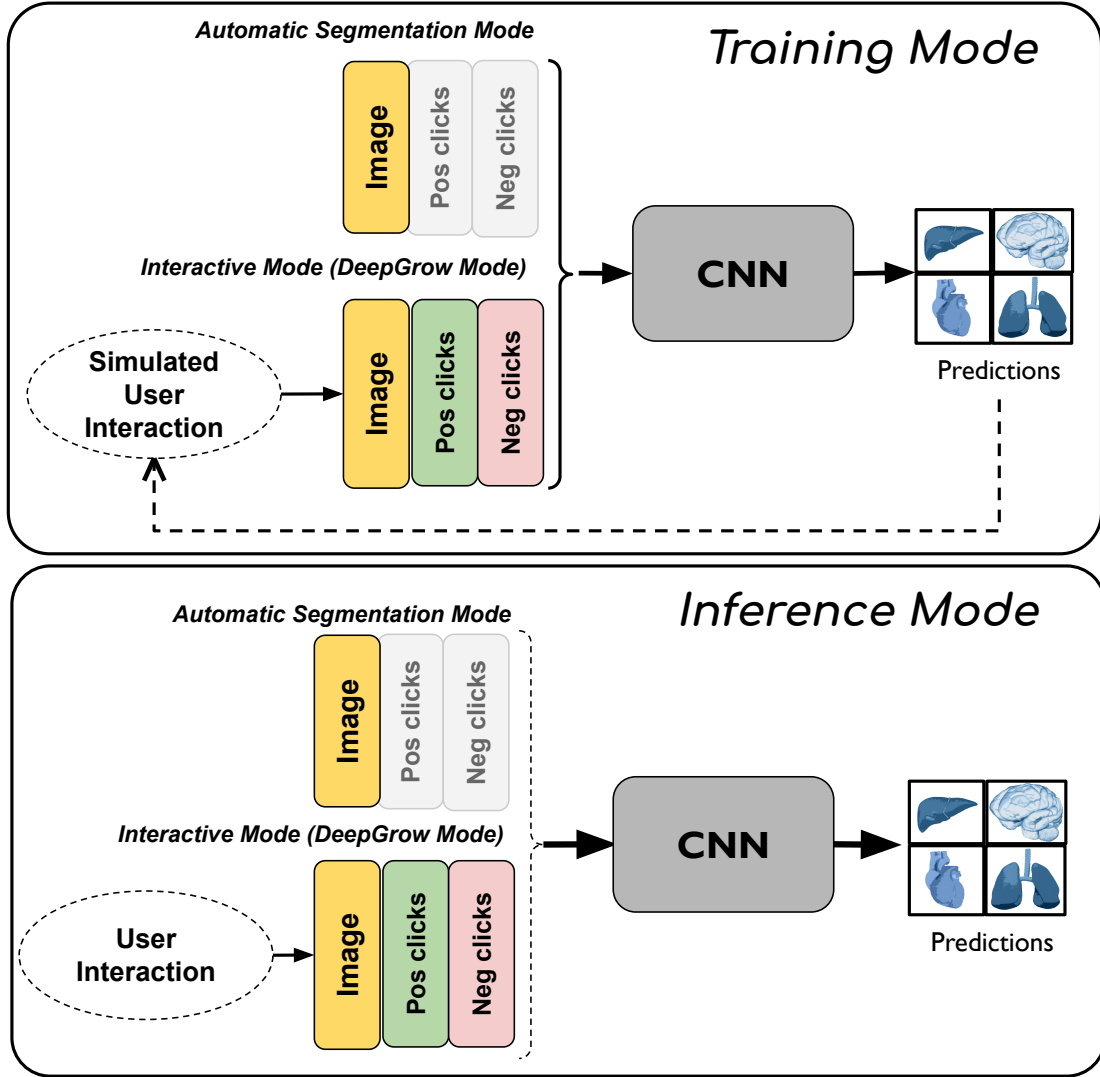
Figure 4: **General schema of the DeepEdit approach:** DeepEdit training process consist of a combination of two modes: the automatic segmentation mode and DeepGrow mode. For inference, input tensor could be either the image with two zero-tensor (automatic segmentation mode) or an image with two tensor representing positive and negative clicks provided by the user (interactive mode or DeepGrow mode).
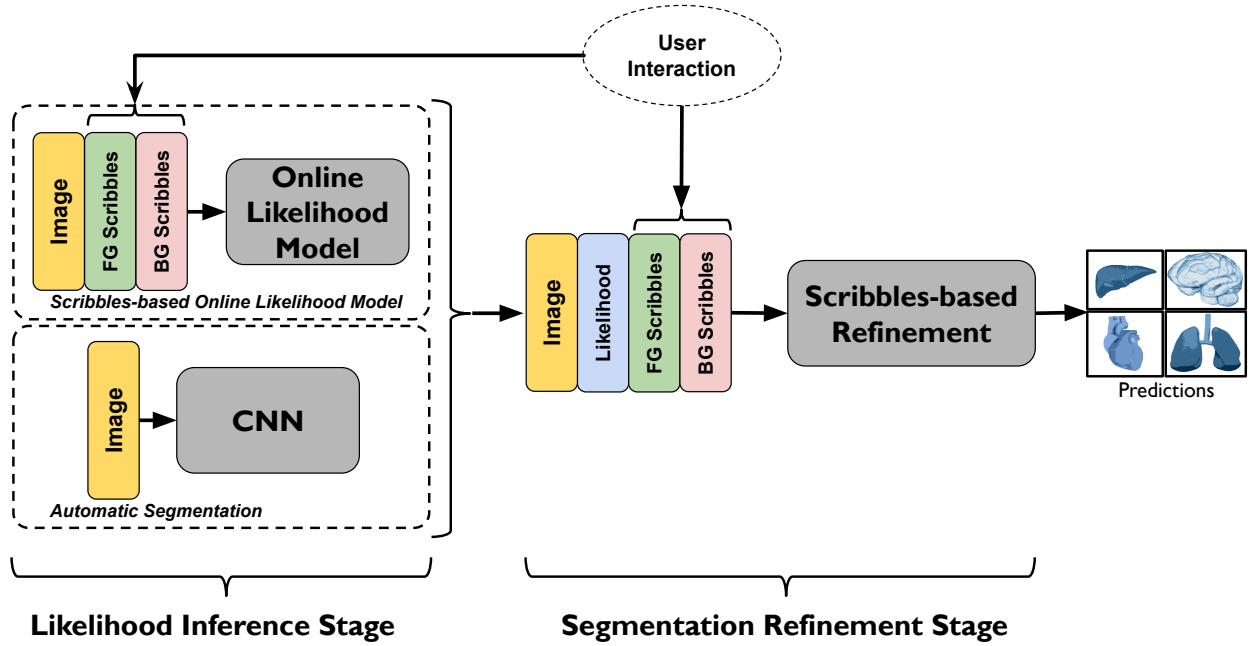
Figure 5: **Scribbles-based interactive segmentation in MONAI Label:** Scribbles-based methods consists of two stages, namely (i) likelihood inference stage and, (ii) segmentation refinement stage. The likelihood can come from either an online model built using the image volume and user-scribbles, or using a pre-trained CNN on image volume alone. The interactions are provided as foreground (FG) or background (BG) scribbles. Image volume, likelihood and scribbles are then used in a refinement stage to refine the initial segmentation using an energy optimization approach, e.g. using GraphCut [35].

Grow and DeepEdit approaches. The most important difference between this annotation app and the interactive ones is the extra information added to the input images. In this app, extra channels representing foreground and background clicks are not added to the input image. Note that the automated segmentation can be then further edited manually using standard segmentation tools provided in the clients.

2. Graphical User Interfaces

2.1. Locally-installed: 3DSlicer

3DSlicer is an open source multi-platform software package widely used for biomedical and medical imaging research [18]. The module implemented to work with MONAI Label handles calls/events created by the user interaction. Current version supports click interaction and allows the user to upload images and labels. Additional interactions such as closed curves, ROI or any other are supported by the MONAI Label server. Researcher can modify this module to make it more dynamic or customised to their MONAI Label segmentation applications.

2.2. Web-based: Open Health Imaging Foundation (OHIF)

The OHIF Viewer is an open-source and web-based viewer [32]. It is based on Cornerstone.js and works out-of-the-box with Image Archives that support DICOMweb. MONAI Label has OHIF embedded and it works with the DICOMweb server support. It also allows the user to create their own label mask and interact with the MONAI Label server. Also important to note that OHIF has been integrated into XNAT and other large initiatives/centers, meaning that it would be easier to integrate into complex workflows.

3. Developing and Deploying MONAI Label Apps

The MONAI Label framework empowers researchers with the ability to develop novel annotation methods and make them readily available to other researchers and clinical staff for continuous evaluation of user experience and annotation performance. Developing a MONAI Label App requires the implementation of a simple Python API

that defines what models are used during annotation, how they learn from user interactions, and how active learning is employed to shorten annotation time as work progresses.

MONAI Label Apps are exposed as a service using the MONAI Label Server. The MONAI Label server exposes the functionality in the MONAI Label App via a RESTful API.

MONAI Label Server enables data management for all MONAI Label Apps, allowing app developers to track various metadata related to training, AI-based annotation, and active learning, and provides caching for images and labels. Data management in MONAI Label Server supports both locally stored datasets and data residing in DICOM servers (accessible via DICOMweb).

3.1. Application Call-flow

MONAI Label App developers may choose to implement part or all of the following functionality in their apps.

- **Active Learning:** MONAI Label offers all the necessary callbacks needed to implement custom active learning techniques. Natively, MONAI Label supports simple active learning techniques such as in-sequence and random sampling, and more highly performant active learning techniques such as Test Time Augmentation [37] (aleatoric uncertainty) and Montecarlo computation using Dropout (epistemic uncertainty).

- **Inference:** This interface allows the developer to use or define the inference task in the MONAI Label App which will define the behaviour of the user interaction (i.e. clicks, scribbles, ROI, etc). The developer will need to define the inference models and inferrers (simple inferrer, sliding window inferrer, etc), and pre- and post-transform.

- **Training:** This interface allows the developer to define the training task for their own MONAI Label App which may include one or more models with the appropriate pre- and post-processing in place. Here, the developer also decides the best data splitting strategy and validation metrics.

- **Custom transforms:** In this file developers can create their custom transforms to apply on the dataset before, after, or during training the deep learning algorithm.

- **Main Module:** Developers define the core structure of the App and declare the modules they defined above, providing IDs for each so they may be used independently and together as required.

- **Requirements File:** Here, researchers have the option of specifying the external libraries required by their app (e.g. TensorFlow, Catalyst, Kornia [38], etc)

MONAI Label enables researchers to build labeling applications in a serverless way, which means that MONAI Label Apps are always ready to deploy via MONAL Label server. To develop a new MONAI labeling app, developers must inherit the `MONAILabelApp` interface and implement the methods in the interface that are relevant to their application. Typically a labeling applications will consist of:

- Inferencing tasks to allow end-users to invoke select pre-trained or actively trained models.

- Training tasks used to train a set of models in the background and perhaps without the users participation to actively improve annotation as work progresses,

- Image selection strategies that choose the unannotated image that is least represented in the already labelled images.

Figure 6 shows the base interfaces that a developer may use to implement their app and the various tasks their app may perform. For example, in the figure the user app `MyApp` employs: two inferencing tasks, namely `MyInfer`, which is a custom implementation of `InferTask`, and `InferDeepGrow2D`, which is a ready-to-use utility included with MONAI Label. There is also one training task, `TrainDeepGrow` which is an extension of the `BasicTrainTask` utility, and two "next image selection" strategies, `TTA` and `Epistemic` included with MONAL Label. These two strategies allows the user to select the next image based on the aleatoric or epistemic uncertainty values. Finally, there is also `MyStrategy` which implements the interface `Strategy` which the end user may select as a custom alternative for next image selection.

MONAI Label currently provides template apps and segmentation approaches which developers may start using out of the box or modify to achieve
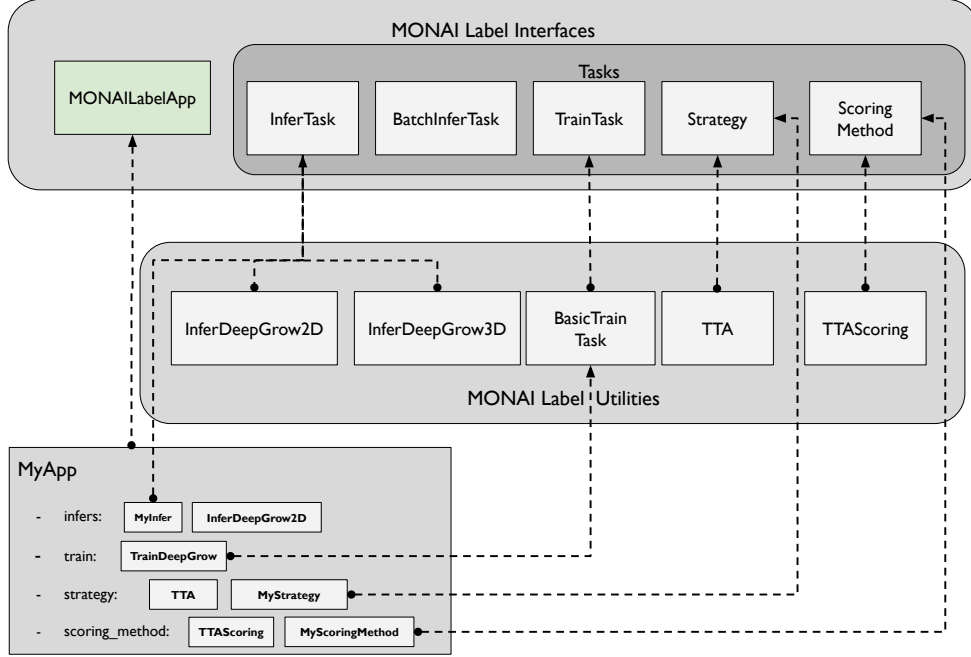
Figure 6: **Modules Overview:** MONAI Label provides interfaces which can be implemented by the label app developer for custom functionality as well as utilities which are readily usable in the labeling app.

the desired behaviour. Template applications currently include annotations based on DeepGrow [9], DeepEdit, and automated segmentation. All template apps with support for Scribbles-based segmentation.

4. Experiments and Results

4.1. DeepGrow Performance

Experiment: The spleen segmentation dataset from medical segmentation challenge (MSD) was utilized for experimentation. The dataset consists of a total of 41 3D CT volumes. Six volumes were randomly selected for validation and the remaining were divided into five for initial training and the others assumed as unlabeled volumes. The experiment is conducted in a 4 stage approach where at every stage the user annotates a certain number of 3D volumes and then adds them to the training pool. At stage 1 the user annotates 5 volumes, at stage 2 the user adds another 5 and at stage 3 and stage 4 the user annotates 10 volumes (more volumes could be annotated because the user could annotate faster due to AI assistance or deepgrow based annotation). For comparison, we also evaluated the scribbles-based online likelihood segmentation that aims at minimizing human interactions

needed to manually annotate a dataset. Baselines of paint brush and advanced contour based techniques were also used for estimating how much time is needed by a human annotator who is utilizing traditional annotation tools. To perform fair comparison the training time for AI models will be observed separately and will not included in the annotation time when comparing to traditional approaches. The reason being that training is performed once and is not a part of the annotation process.

Results: Observing Table 1, it can be seen that the time taken by the user to annotate a single 3D volume grows lesser as more training data is added per stage for the AI model to learn from. At 4th stage it can be observed that utilizing the combined pipeline of Deepgrow 2D & 3D the user can annotate 3D volumes in approximately 1 - 2.5 minutes which is 10x faster as even compared to advanced traditional technique of contouring to manually annotate the 3D volume.

Comparing scribbles-based online likelihood segmentation with the traditional paint brush and contour-based techniques in Table 1, we note that scribbles-based method significantly improves time to manually annotate dataset, while minimizing

user-interactions required. In particular, on average 2 minutes were required to annotate a sample using scribbles-based method which is 12.5× and 6.25× faster than using paint brush and contour-based method, respectively.

4.2. DeepEdit Performance

In order to demonstrate how the DeepEdit annotation approach can be used to facilitate medical image segmentation, we used cardiac magnetic resonance images (CMR) from the Cardiac task available in the Medical Segmentation Decathlon (MSD) [7] to segment the left atrium. This dataset is composed of 20 CMR images. A split of 80% for training and 20% for validation was used in this example. This means, 16 images were randomly selected for training and 4 images for validation. A learning rate of 1e-4, batch size equal to 1, Adam optimizer, 50 epochs by default and random affine transformation were used for data augmentation.

For this experiment, we assume a clinician takes approximately 10 minutes to manually annotate the left atrium using the available tools in 3DSlicer (i.e. Grow from Seeds, Brush, etc). This means, a clinician should spend around 160 minutes to fully segment the train split (16 CMR images) before they can start training a deep learning model (See red line in Figure 7). However, if the clinician uses MONAI Label with the DeepEdit approach, they could start the training process after segmenting the first one or two CMR images. This allows the clinician to use the obtained model to continue the annotation of the other images.

The model trained on one or two images might not perform well in the beginning, but it helps the clinician to quickly create a label that they can modify using the interactive part of the DeepEdit (clicks), which significantly reduces the time they spend on the other images. See Figure 7 (Green line)

This way of annotating and training interactive deep learning models allows the clinicians to reduce the time and effort spent on this process. As it can be seen from Figure 7 and Table 1, MONAI Label could help clinicians to reduce the time they spend annotating images by almost 60%. This can be seen by checking the difference in time to get 0.60 Validation Dice (20% - Four 3D volumes of the heart MSD dataset) score when using the standard way (Red Line) against the interactive way using the DeepEdit in MONAI Label.

5. Discussion

We have introduced MONAI Label[7], a free and open source image labeling and learning tool that enables users to create annotated datasets and build AI annotation models for clinical evaluation. MONAI Label reduces the time and effort of annotating new datasets and enables the adaptation of AI to the task at hand by continuously learning from user interactions via two different user interfaces: 3DSlicer and OHIF.

MONAI Label employs three main different annotation algorithms: two interactive (DeepGrow and DeepEdit), and one non-interactive Automatic Segmentation algorithms. In addition to this, a scribbles-based algorithm is also available to quickly segment the images.

MONAI Label platform is compatible with other libraries beyond MONAI Core [34]. It was developed to help researchers and clinicians to facilitate 3D medical image annotations and allow them to easily implement new machine learning or deep learning algorithms. For that reason, we designed it in a way that users can seamlessly work with different deep learning frameworks and high-level libraries such as Ignite and PyTorch Lightning.

Finally, MONAI Label offers a heuristic planner that considers available GPU and intensity and spatial information of the training set to define the data transform and hyperparameters used during training and inference.

Further work includes more user interactions such as ROIs and closed curves, and DICOM SEG support. The authors welcome constructive feedback, feature requests, and contributions from the community.

Code Availability Statement

MONAI Label is an open source project and is freely available at `https://github.com/Project-MONAI/MONAILabel` under the Apache-2.0 license. It was developed and tested for Linux and Windows operating systems. Any common browser can be used to work on OHIF viewer. MONAI Label can be installed according to the installation instructions available in the GitHub project.

---

[7] `https://github.com/Project-MONAI/MONAILabel`

Table 1: **Obtained results from the DeepGrow validation using the Spleen MSD dataset:** Time spent on manual and automatic annotation, and Validation Dice score obtained from the Spleen MSD dataset using manual tools and the DeepGrow annotation app on MONAI Label. Validation Dice Scores were obtained from nine 3D volumes (20%) randomly selected from the Spleen MSD dataset.

|	Annotated volumes	Annotation Time (Paint Brush)	Annotation Time (Contour-based method)	Annotation Time (Scribbles-based method)	Manual OR AI Annotation Time per volume	Total annotation time	Training Time (DeepGrow 2D + 3D)	Validation Dice Score DeepGrow 2D	Validation Dice Score DeepGrow 3D
Stage 1	11	275 mins	137.5 mins	22 mins	~25 mins	~275 mins	90 mins	0.891	0.730
Stage 2	$11 + (5) = 16$	400 mins	200 mins	32 mins	~6 - 7.5 mins	~30 mins	135 mins	0.924	0.873
Stage 3	$11 + 5 + (10) = 26$	650 mins	325 mins	52 mins	~3.5 - 5 mins	~45 mins	250 mins	0.948	0.945
Stage 4	$11 + 5 + 10 + (10) = 36$	900 mins	450 mins	72 mins	~1 - 2.5 mins	~15 mins	360 mins	0.967	0.959
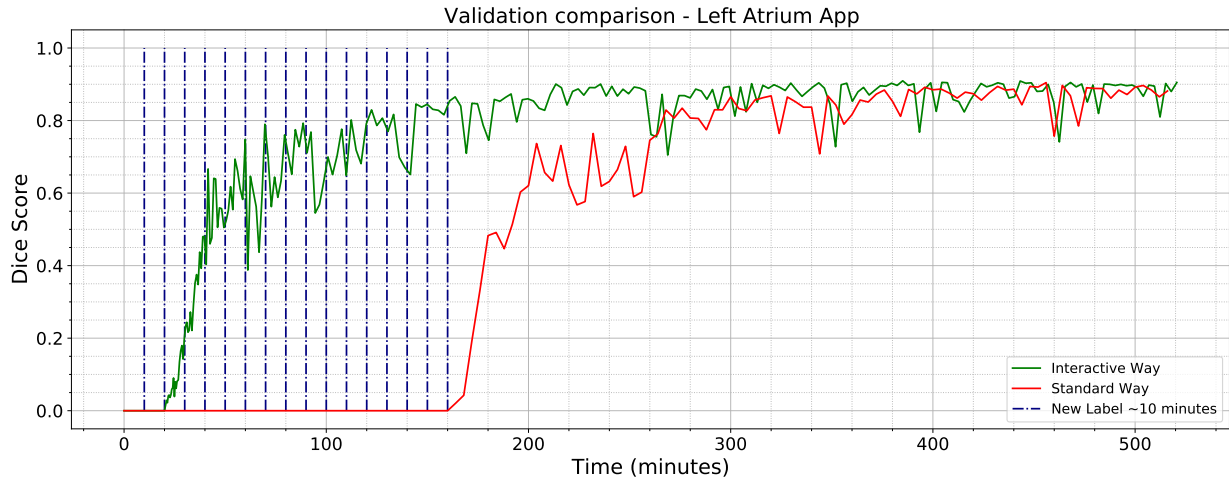
Figure 7: **Validation of the DeepEdit approach:** Obtained results on left atrium segmentation on the MSD dataset using the DeepEdit approach. Vertical blue lines represent the time spent by a clinician to annotate and submit a label to MONAI Label platform.

References

[1] F. Isensee, P. F. Jaeger, S. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation," *Nature Methods*, vol. 18, pp. 203–211, 2020.

[2] Y. He, D. Yang, H. Roth, C. Zhao, and D. Xu, "DiNTS: Differentiable Neural Network Topology Search for 3D Medical Image Segmentation," *arXiv: 2103.15954*, vol. abs/2103.15954, 2021.

[3] A. Hatamizadeh, D. Yang, H. Roth, and D. Xu, "UN-ETR: Transformers for 3D Medical Image Segmentation," *arXiv: 2103.10504*, vol. abs/2103.10504, 2021.

[4] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Springer, 2015.

[5] O. Çiçek, A. Abdulkadir, S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation," in *Medical Image Computing and Computer-Assisted Intervention*, pp. 424–432, 2016.

[6] F. Milletari, N. Navab, and S.-a. Ahmadi, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation," in *3DV*, 2016.

[7] M. Antonelli, A. Reinke, S. Bakas, K. Farahani, AnnetteKopp-Schneider, B. A. Landman, G. Litjens, B. Menze, O. Ronneberger, R. M. Summers, B. van Ginneken, M. Bilello, P. Bilic, P. F. Christ, R. K. G. Do, M. J. Gollub, S. H. Heckers, H. Huisman, W. R. Jarnagin, M. K. McHugo, S. Napel, J. S. G. Pernicka, K. Rhode, C. Tobon-Gomez, E. Vorontsov, H. Huisman, J. A. Meakin, S. Ourselin, M. Wiesenfarth, P. Arbelaez, B. Bae, S. Chen, L. Daza, J. Feng, B. He, F. Isensee, Y. Ji, F. Jia, N. Kim, I. Kim, D. Merhof, A. Pai, B. Park, M. Perslev, R. Rezaiifar, O. Rippel, I. Sarasua, W. Shen, J. Son, C. Wachinger, L. Wang, Y. Wang, Y. Xia, D. Xu, Z. Xu, Y. Zheng, A. L. Simpson, L. Maier-Hein, and M. J. Cardoso, "The medical segmentation decathlon," *arXiv preprint arXiv:2106.05735*, vol. abs/2106.05735, 2021.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, vol. 2017, pp. 6000–6010, 2017.

[9] T. Sakinis, F. Milletari, H. Roth, P. Korfiatis, P. M. Kostandy, K. Philbrick, Z. Akkus, Z. Xu, D. Xu, and B. J. Erickson, "Interactive segmentation of medical images through fully convolutional neural networks," *arXiv preprint arXiv:1903.08205*, vol. abs/1903.08205, 2019.

[10] F. Zhao and X. Xie, "An Overview of Interactive Medical Image Segmentation," *Annals of the British Machine Vision Association*, vol. 2013, no. 7, pp. 1–22,

12

2013.

[11] N. Xu, B. Price, S. Cohen, J. Yang, and T. Huang, "Deep Interactive Object Selection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 373–381, 2016.

[12] E. Agustsson, J. R. Uijlings, and V. Ferrari, "Interactive Full Image Segmentation by Considering All Regions Jointly," in *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 11614–11623, 2019.

[13] X. Luo, G. Wang, T. Song, J. Zhang, M. Aertsen, J. Deprest, S. Ourselin, T. Vercauteren, and S. Zhang, "MIDeepSeg: Minimally interactive segmentation of unseen objects from medical images using deep learning," *Medical Image Analysis*, vol. 72, p. 102102, 2021.

[14] G. Wang, W. Li, M. A. Zuluaga, R. Pratt, P. A. Patel, M. Aertsen, T. Doel, A. L. David, J. Deprest, S. Ourselin, and T. Vercauteren, "Interactive Medical Image Segmentation Using Deep Learning with Image-Specific Fine Tuning," *IEEE Transactions on Medical Imaging*, vol. 37, no. 7, pp. 1562–1573, 2018.

[15] G. Wang, M. A. Zuluaga, W. Li, R. Pratt, P. A. Patel, M. Aertsen, T. Doel, A. L. David, J. Deprest, S. Ourselin, and T. Vercauteren, "DeepIGeoS: A Deep Interactive Geodesic Framework for Medical Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1559–1572, 2019.

[16] P. A. Yushkevich, J. Piven, H. Cody Hazlett, R. Gimpel Smith, S. Ho, J. C. Gee, and G. Gerig, "User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability," *Neuroimage*, vol. 31, no. 3, pp. 1116–1128, 2006.

[17] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *International Journal of Computer Vision*, pp. 321–331, 1988.

[18] A. Fedorov, R. Beichel, J. Kalpathy-Cramer, J. Finet, J.-C. Fillion-Robin, S. Pujol, C. Bauer, D. Jennings, F. Fennessy, M. Sonka, J. Buatti, S. Aylward, J. V. Miller, S. Pieper, and R. Kikinis, "3D Slicer as an Image Computing Platform for the Quantitative Imaging Network," *Magnetic resonance imaging*, vol. 30, pp. 1323–1341, Nov. 2012.

[19] M. Nolden, S. Zelzer, A. Seitel, D. Wald, M. Müller, A. M. Franz, D. Maleike, M. Fangerau, M. Baumhauer, L. Maier-Hein, K. H. Maier-Hein, H. P. Meinzer, and I. Wolf, "The medical imaging interaction toolkit: Challenges and advances: 10 years of open-source development," *International Journal of Computer Assisted Radiology and Surgery*, vol. 8, no. 4, pp. 607–620, 2013.

[20] K. K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool, "Deep Extreme Cut: From Extreme Points to Object Segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 616–625, 2018.

[21] H. R. Roth, D. Yang, Z. Xu, X. Wang, and D. Xu, "Going to extremes: Weakly supervised medical image segmentation," *Machine Learning and Knowledge Extraction*, vol. 3, no. 2, pp. 507–524, 2021.

[22] NVIDIA, "NVIDIA AI-Assisted Annotation (AIAA) - Clara Train SDK," 2021.

[23] R. Adams and L. Bischof, "Seeded Region Growing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 641–647, 1994.

[24] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, 1988.

[25] S. Berg, D. Kutra, T. Kroeger, C. N. Straehle, B. X. Kausler, C. Haubold, M. Schiegg, J. Ales, T. Beier, M. Rudy, *et al.*, "Ilastik: interactive machine learning for (bio) image analysis," *Nature Methods*, vol. 16, no. 12, pp. 1226–1232, 2019.

[26] P. Lösel and V. Heuveline, "Enhancing a diffusion algorithm for 4D image segmentation using local information," in *SPIE Medical Imaging 2016: Image Processing*, vol. 97842L, 2016.

[27] P. D. Lösel, T. van de Kamp, A. Jayme, A. Ershov, T. Faragó, O. Pichler, N. Tan Jerome, N. Aadepu, S. Bremer, S. A. Chilingaryan, M. Heethoff, A. Kopmann, J. Odar, S. Schmelzle, M. Zuber, J. Wittbrodt, T. Baumbach, and V. Heuveline, "Introducing Biomedisa as an open-source online platform for biomedical image segmentation," *Nature Communications*, vol. 11, 2020.

[28] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient N-D image segmentation," *International Journal of Computer Vision*, vol. 70, no. 2, pp. 109–131, 2006.

[29] A. Criminisi, T. Sharp, and A. Blake, "Geos: Geodesic image segmentation," in *European Conference on Computer Vision*, pp. 99–112, Springer, 2008.

[30] D. S. Marcus, T. R. Olsen, M. Ramaratnam, and R. L. Buckner, "The extensible neuroimaging archive toolkit: An informatics platform for managing, exploring, and sharing neuroimaging data," *Neuroinformatics*, vol. 5, no. 1, pp. 11–33, 2007.

[31] A. Fedorov, W. J. Longabaugh, D. Pot, D. A. Clunie, S. Pieper, H. J. Aerts, A. Homeyer, R. Lewis, A. Akbarzadeh, D. Bontempi, W. Clifford, M. D. Herrmann, H. Höfener, I. Octaviano, C. Osborne, S. Paquette, J. Petts, D. Punzo, M. Reyes, D. P. Schacherer, M. Tian, G. White, E. Ziegler, I. Shmulevich, T. Pihl, U. Wagner, K. Farahani, and R. Kikinis, "Nci imaging data commons," *Cancer Research*, vol. 81, no. 16, pp. 4188–4193, 2021.

[32] T. Urban, E. Ziegler, R. Lewis, C. Hafey, C. Sadow, A. D. Van den Abbeele, and G. J. Harris, "Lesiontracker: extensible open-source zero-footprint web viewer for cancer imaging research and clinical trials," *Cancer research*, vol. 77, no. 21, pp. e119–e122, 2017.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[34] MONAI Consortium, "MONAI: Medical Open Network for AI," 3 2020.

[35] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.

[36] C. Rother, V. Kolmogorov, and A. Blake, ""grabcut" interactive foreground extraction using iterated graph cuts," *ACM transactions on graphics (TOG)*, vol. 23, no. 3, pp. 309–314, 2004.

[37] G. Wang, W. Li, M. Aertsen, J. Deprest, S. Ourselin, and T. Vercauteren, "Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks," *Neurocomputing*, vol. 338, pp. 34–45, 2019.

13

[38] E. Riba, D. Mishkin, D. Ponsa, E. Rublee, and G. Bradski, "Kornia: an open source differentiable computer vision library for pytorch," in *Winter Conference on Applications of Computer Vision*, 2020.