

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

Andres Diaz-Pinto^{a,b,*}, Sachidanand Alle^b, Vishwesh Nath^b, Yucheng Tang^b, Alvin Ihsanib, MuhammadAsada, Fernando Pérez-García^{a,c}, Pritesh Mehta^{a,c}, Wenqi Lib, Mona Flores^b, Holger R. Roth^b, Tom Vercauteren, Daguang Xub, Prerna Dograb, Sebastien Ourselina, Andrew Feng^b, M. Jorge Cardoso^a



Problem

- The high cost and time required for manual annotations in medical imaging

- 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

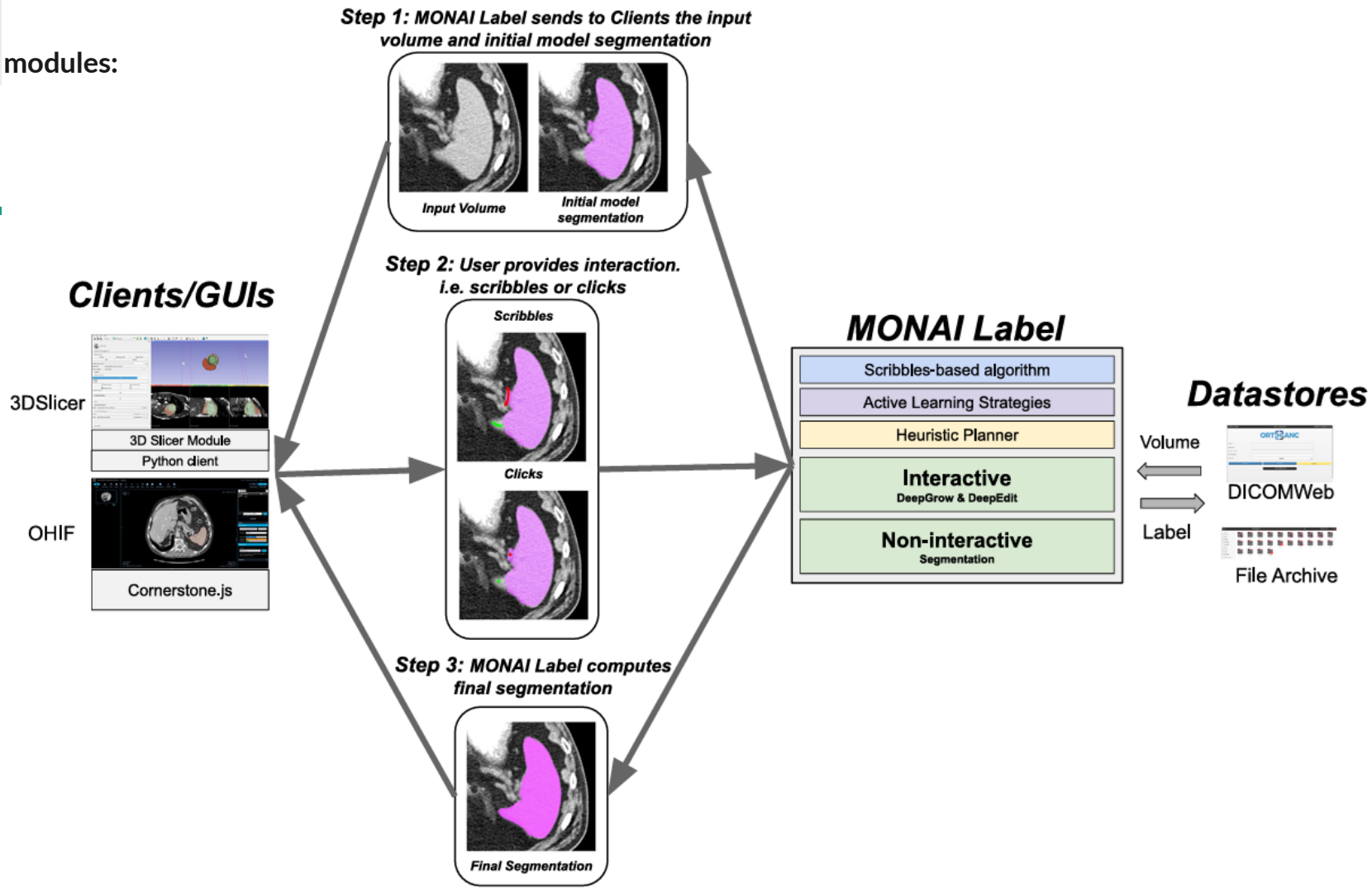
- semi automatic platform:

 - BioMedisa an online platform developed for the semi-automatic segmentation of large volumetric images

Monai label:

MONAI Label as an open-source framework designed for faster, AI-assisted annotation of medical images

Monai label modules:





Annotation approaches

-Interactive :

DeepGrow (Sakinis et al., 2019)

DeepEdit (Diaz-Pinto et al., 2022)

Scribbles-based method

-Non-interactive(Automatic Segmentation):

allows the researcher_ to create a segmentation pipeline

Use network in Monai/(UNet,Highresnet, ResNet,etc) to automatically segment images

Interactive approaches: DeepGrow

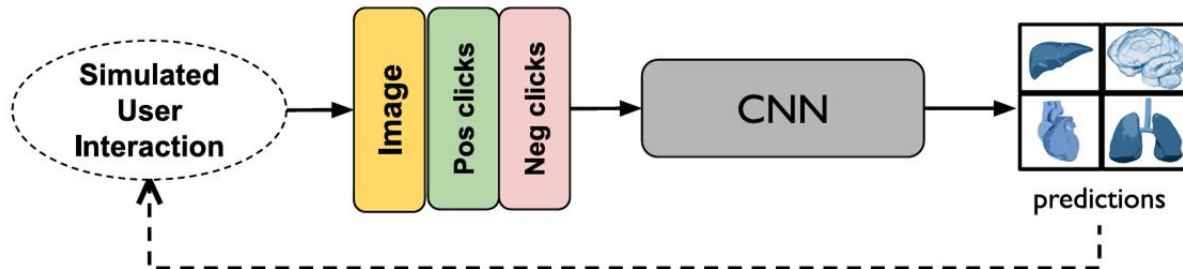
- user guides the segmentation with positive and negative clicks (Sakinis et al., 2019)

The positive clicks → expand the segmentation to include that location

negative clicks → contracting the segmentation to exclude

Training:

Using the simulation of positive and negative guidance derived from FP and FN





Interactive approaches: DeepEdit

Different from DeepGrow:

- click-free segmentation inference and click-based segmentation editing

- DeepEdit facilitates the user experience and development of **new active learning**

Training phase:

- a combination of simulated clicks and standard non-interactive training

- two training stages: For half of the iterations→tensors representing the foreground and background points are zeros

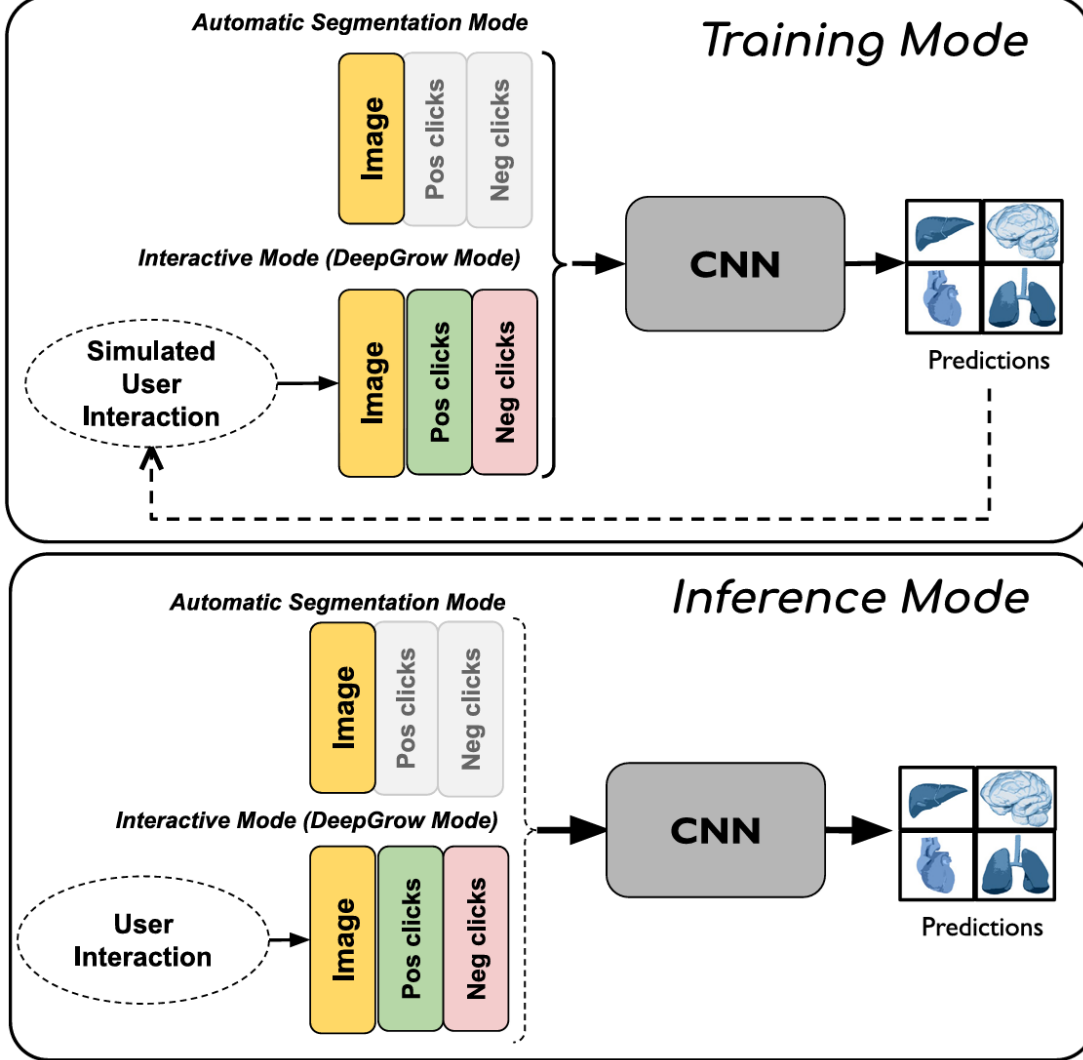
Other half→ positive and negative clicks are simulated(Sakinis et al. (2019)

Automatic inference phase:

- the tensors representing positive and negative clicks → zeros.



DeepEdit





Interactive approaches: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

-Advantage:

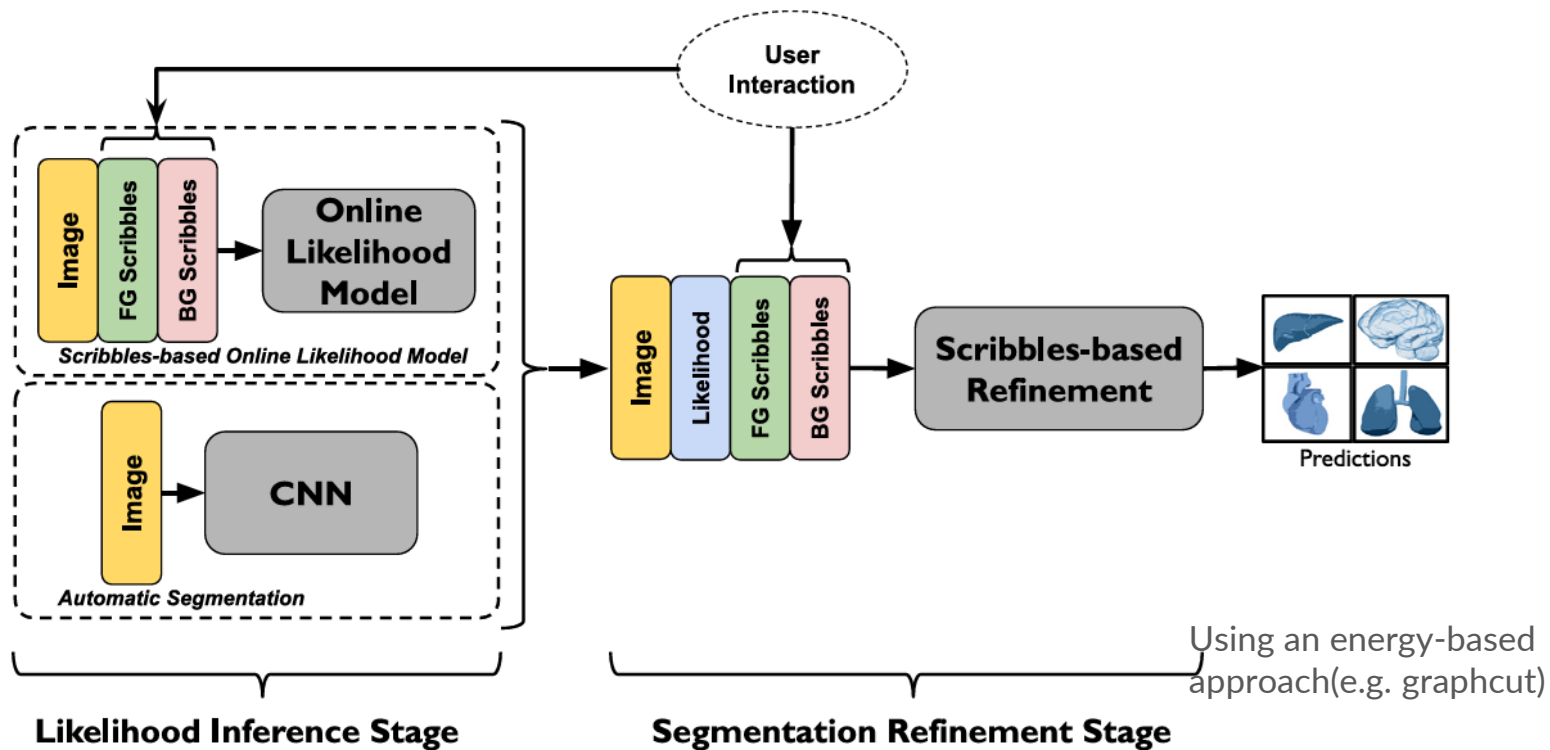
flexibility;providing both minimal interactions for simpler delineation tasks and detailed interactions for more difficult segmentations

- MONAI Label provides **APIs** for implementing scribbles-based interactive segmentation workflows:

Scribble-based seg can be used in 2 modes:

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

Scribbles-based interactive segmentation





Non-Interactive approaches: Automated segmentation model

The MONAI Label supports automated segmentation:

- non-interactive algorithm based on a (CNN) (i.e. UNet).
- Researchers can use any of the available networks created in MONAI Core for their purposes

Active learning

-Derive uncertainty(epistemic) from a ML model:

1-Query-by-committee

2-Utilizing dropout during inference with bootstrapping → monai uses as single model is cheaper

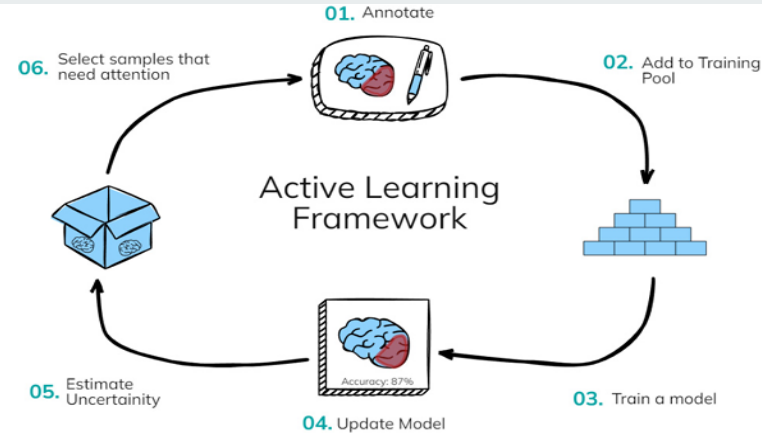
3- Bayesian models

-* user is able to use ensemble or bayesian as well

-acquisition function:

choose subset of samples with highest uncertainty → Monai uses this one

-* user is able to use other acquisition functions as well





Active learning

-Network Architecture:

Dropout-based techniques are model agnostic

-Hyperparameters of AL:

-number of simulations when using dropout

-the number of data points to be annotated

-parameters particular to acquisition functions

-Why DeepEdit allows the usage of uncertainty-based AL techniques?



Graphical user interfaces

- **Locally-installed:** **3D Slicer** is an open-source multi-platform software package used for medical imaging research .

- handles calls/events created by the user interaction

- supports click interaction and allows the user to upload images and labels

- Researchers can modify this module according to type of interaction, to customize their MONAI Label segmentation applications

-**Web-based:** **Open Health Imaging Foundation (OHIF)** The OHIF Viewer is an open-source and web-based viewer 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

- allows the user to create their own label mask and interact with the MONAI Label server



Developing and deploying MONAI Label apps

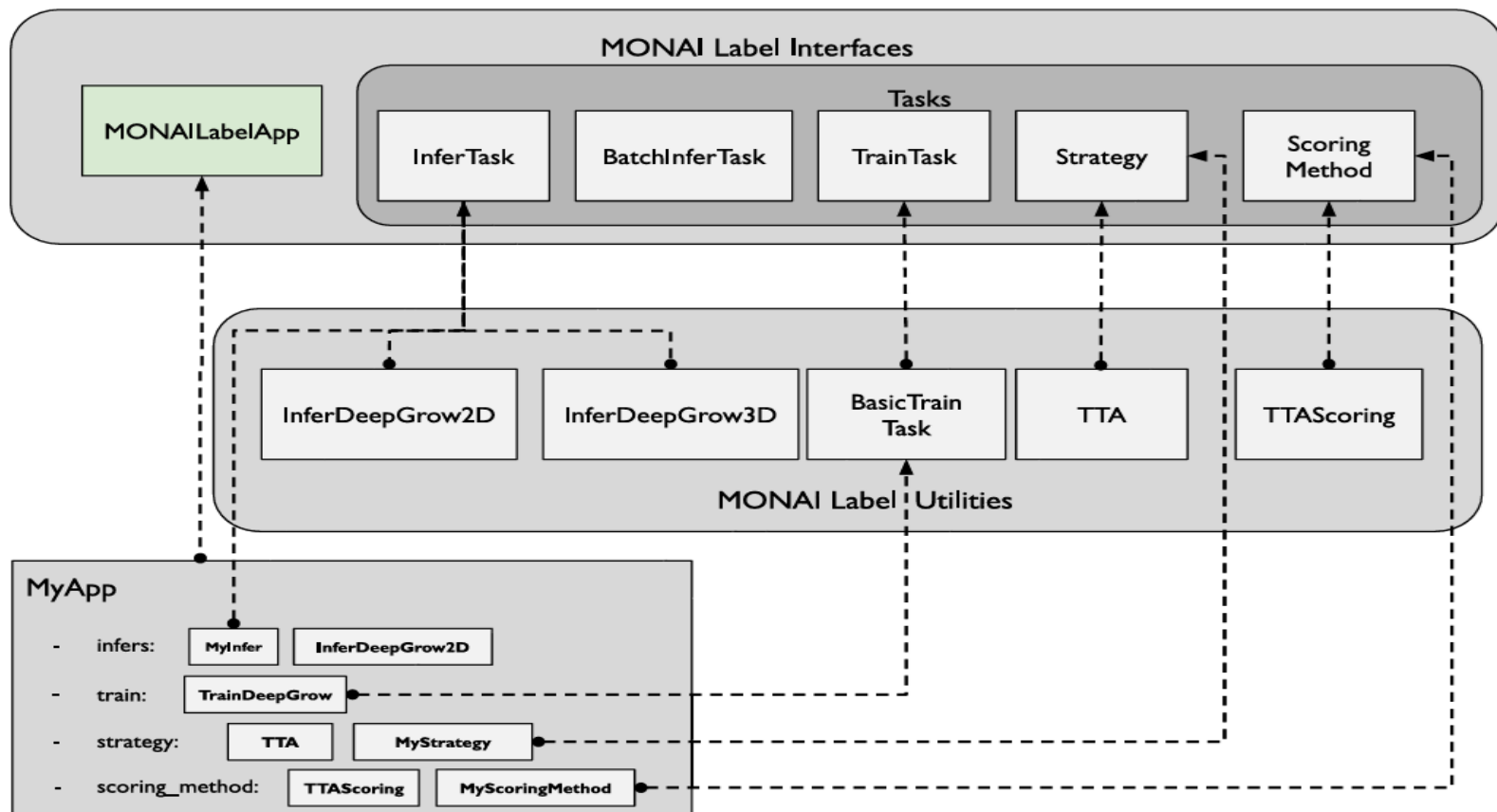
-Developing a MONAI Label App :

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

-To develop a new MONAI labeling app, developers must inherit the MONAILabelApp interface .

Labeling applications typically consist of:

- **Inferencing tasks** to allow end-users to use selected pre-trained or actively trained models.
- **Training tasks** used to train a set of models in the background
- **Image selection strategies** that choose the unannotated image that is least represented in the already labeled images





Experiments and results: DeepGrow

- The spleen segmentation dataset from the medical segmentation challenge (MSD)
- 41 3D CT volumes. 6 for validation . 5 for initial training and the others assumed as unlabeled volumes.
- conducted in a 4 stage : user annotates a certain number of 3D volumes

Baselines :paintbrush and advanced contour-based techniques

Results: time taken by the user to annotate a single 3D volume grows less as more training data is added per stage. At the 4th stage, it can be observed that by utilizing the combined pipeline of Deepgrow 2D & 3D, the user can annotate 3D volumes in approximately **1-2.5 min**, which is 10x faster than the advanced traditional technique.

-scribbles-based online likelihood vs. traditional paintbrush

on average 2 min were required to annotate a sample using the scribbles-based method which is 12.5× and 6.25× faster than using the paintbrush and contour-based method



Experiments and results: DeepEdit

- cardiac magnetic resonance images (CMR) to segment the left atrium.

- 20 CMR images. 16 for training and 4 for validation. A learning rate of $1e-4$, batch size=1, Adam optimizer, 50 epochs, and data augmentation: random affine transformation

- time an expert annotator took to manually annotate the left atrium (10 min) using the manual/basic available tools in 3D Slicer → around 160 min to fully segment the train split (16 CMR images) before they can start training a deep learning model

- With DeepEdit:

- start the training process after segmenting the first one or two CMR images.

- allows clinician to use the obtained model to continue the annotation of the other images.

- might not perform well first, but 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



Table 1

Time spent on annotating the Spleen MSD dataset: This table presents the total time spent on each stage annotating volumes from the Spleen dataset using several manual tools (paintbrush, contour-based and scribbles-based methods) and the interactive DeepGrow model available on MONAI Label. As the DeepGrow model needs initial training, a manual tool has to be used to get the first annotated samples (User's Choice).

	Annotated volumes	Paint Brush (PB) method	Contour-Based (CB) method	Scribbles-Based (SB) method	DeepGrow method
Stage 1	11	~275 min	~137.5 min	~22 min	User's Choice (PB/CB/SB)
Stage 2	11 + (5) = 16	~400 min	~200 min	~32 min	~ 31 min
Stage 3	11 + 5 + (10) = 26	~650 min	325 min	~52 min	~ 36 min
Stage 4	11 + 5 + 10 + (10) = 36	~900 min	~450 min	~72 min	~ 38.5 min

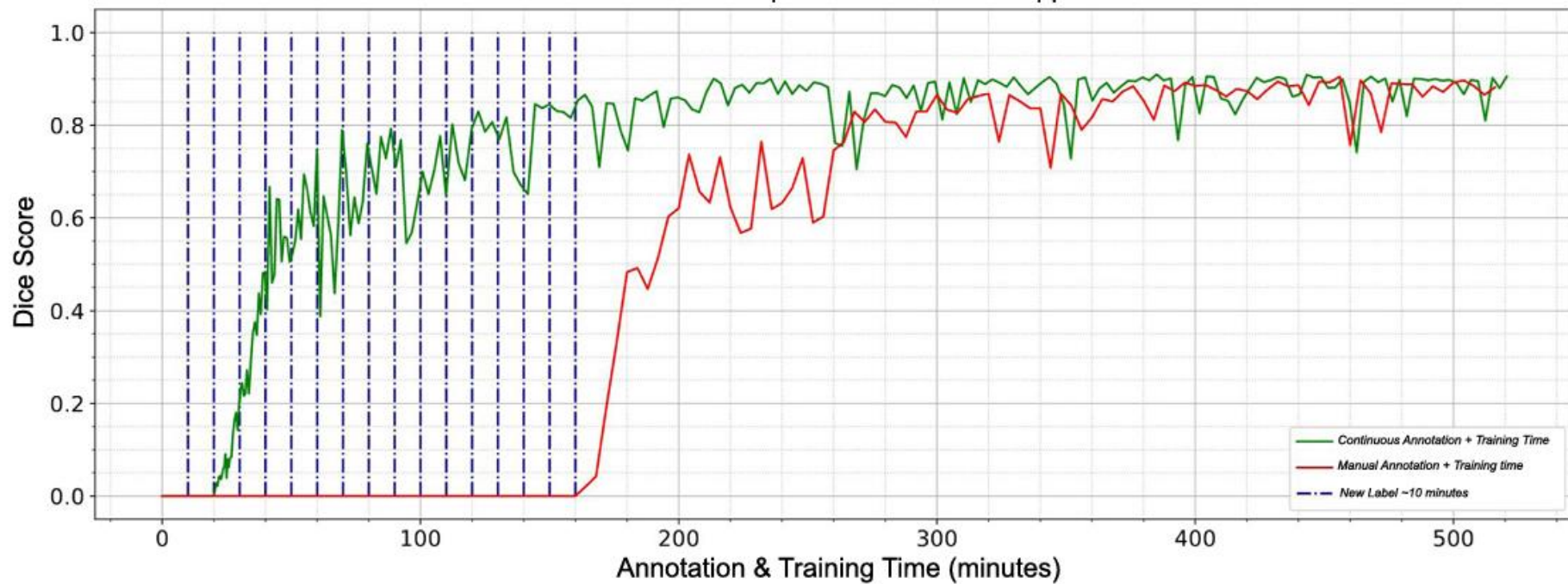

Table 2

Obtained results from the interactive DeepGrow model: Total annotation time per stage, training time on each stage, and the validation Dice Scores on the Spleen MSD dataset. The validation set is composed of nine 3D volumes (20%) that were randomly selected from the Spleen MSD dataset.

	Total annotation time using DeepGrow	Training time DeepGrow 2D & 3D	Validation dice DeepGrow 2D	Validation dice DeepGrow 3D
Stage 1	275 min	90 min	0.891	0.730
Stage 2	30 min	135 min	0.924	0.873
Stage 3	45 min	250 min	0.948	0.945
Stage 4	15 min	360 min	0.967	0.959

Validation of the DeepEdit approach

Validation comparison - Left Atrium App



At first no results when not using deepedit