

Label, refine, repeat: extending nnInteractive with dataset traversal and nnU-Net proposals

Nico Schmutzenhofer, Lukas Förner, Sina Wendrich, Kartikay Tehlan, Thomas Wendler

Angaben zur Veröffentlichung / Publication details:

Schmutzenhofer, Nico, Lukas Förner, Sina Wendrich, Kartikay Tehlan, and Thomas Wendler. 2026. "Label, refine, repeat: extending nnInteractive with dataset traversal and nnU-Net proposals." In *Bildverarbeitung für die Medizin 2026: proceedings, German Conference on Medical Image Computing, Lübeck, March 15-17, 2026*, edited by Heinz Handels, Katharina Breininger, Thomas Deserno, Andreas Maier, Klaus Maier-Hein, Christoph Palm, and Thomas Tolxdorff, 307–14. Wiesbaden: Springer Fachmedien. https://doi.org/10.1007/978-3-658-51100-5_61.

Nutzungsbedingungen / Terms of use:

CC BY 4.0



Label, Refine, Repeat

Extending nnInteractive with Dataset Traversal and nnU-Net Proposals

Nico Schmutzenhofer ^{1,2†}, Lukas Förner ^{1,2,3†}, Sina Wendrich ^{1,2},
Kartikay Tehlan ^{1,2,3}, Thomas Wendler ^{1,2,3,4,5}

¹Department of Diagnostic and Interventional Radiology and Neuroradiology, Augsburg, Germany

²Digital Medicine, University Hospital Augsburg, Augsburg, Germany

³Computer-Aided Medical Procedures and Augmented Reality, Technical University of Munich, Garching bei München, Germany

⁴Center of Advanced Analytics and Predictive Sciences, University of Augsburg, Germany

⁵Bavarian Cancer Research Center (BZKF) Augsburg, Germany

lukas.foerner@uk-augsburg.de

Abstract. Interactive segmentation tools accelerate annotation but often operate on a single image at a time and lack robust initialization. We extend nnInteractive with (i) dataset-aware traversal that supports sequential/on-the-fly loading of images and volumes, similar to MONAI Label’s iteration approach, and (ii) model-assisted pre-segmentation via nnU-Net to warm-start the interaction. Our traversal module adds persistent progress tracking and resume-from-last functionality, enabling efficient, dataset-scale labeling sessions. The nnU-Net integration generates class-wise candidate masks (2D/3D) that are overlaid in the UI for immediate acceptance, editing, or rejection. Users can toggle structures, adjust thresholds, and refine them with standard nnInteractive tools. The system supports batch precomputation or on-the-fly inference, and logs edits to facilitate auditability and reproducibility. We illustrate end-to-end workflows on the pediatric BraTS dataset of 2024 and outline evaluation protocols (clicks, correction time, Dice-Sørensen score after refinement) to quantify annotation efficiency. Together, these additions turn nnInteractive into a scalable, dataset-centric labeling environment that leverages state-of-the-art segmentation priors while preserving expert control. The code is available at: <https://github.com/Clinical-Computational-Medical-Imaging/label-refiner>

1 Introduction

Accurate and efficient tumor segmentation plays a central role in medical image analysis, forming the foundation for reliable diagnosis, personalized treatment plan-

[†]These authors contributed equally to this work.

ning, and effective disease monitoring. Despite its importance, this process is often challenging, as manual delineation of organ and anomaly boundaries is both time-consuming and highly dependent on the operator's expertise, especially when structures exhibit complex shapes or heterogeneous tissue characteristics. Among the available software solutions, *3D Slicer* [1, 2] stands out as a widely used open-source platform that offers a flexible and extensible environment for medical image visualization, processing, and quantitative analysis. To accelerate the segmentation process for clinical staff, new tools have been developed that leverage pretrained models to automatically identify and segment entire connected structures, whether benign or malignant. Approaches such as *nnInteractive* [3] exemplify this trend, offering flexible frameworks that adapt to various imaging modalities and anatomical regions while reducing the need for extensive manual input. However, most of these methods assume single-image workflows and start from scratch per case without leveraging strong priors [4, 5]. We address these challenges by combining dataset-aware traversal, model-assisted segmentation, and scalable logging with standardized evaluation protocols, resulting in an interactive segmentation tool that preserves expert control over the output. The proposed tool is implemented as an open source extension within *3D Slicer*, enabling seamless integration into existing clinical and research workflows. By integrating manual segmentation with predictions from a pretrained *nnU-Net* [6], and offering interactive refinement through *nnInteractive* prompts, we aim to reduce the time required for accurate expert segmentation. Users can substitute any task-specific *nnU-Net* model by specifying the model path in the configuration; *nnU-Net* requires training a new model for each segmentation task. Our primary objective is to reduce annotation time and interaction effort while maintaining segmentation quality comparable to purely manual annotation. We hypothesize that, in cases where models are available, *nnU-Net* proposals provide a useful initialization that reduces corrective interactions, particularly for challenging cases.

2 Materials and methods

The advancements to the *nnInteractive* plugin [7] comprise three components: volume loading, automatic segmentation, and assisted refinement.

2.1 Volume Loading

To enable rapid loading of subsequent imaging volumes, we implemented a *Next Volume* action that automatically loads the next volume from a specified input directory, supporting both NIfTI and DICOM formats.

For NIfTI, the user selects a root directory and all *.nii* and *.nii.gz* files are indexed recursively and grouped by patient ID, with each group shown as a separate case. For DICOM, we traverse the standard Patient-Study-Series levels and group the series of one patient into a selectable case.

This design ensures efficient and reproducible loading of imaging data for both single- and multi-volume analyses.

2.2 Automatic Segmentation

We integrate nnU-Net to provide an initial proposal that warm-starts the interactive session (Fig. 1, left). Two execution modes are supported: *on-the-fly* inference at load time, and *batch precomputation* prior to labeling sessions.

2.2.1 Pre-processing. Each input modality is resampled to the model’s target spacing and intensity-normalized according to nnU-Net conventions. Missing modalities can be set to *None*; dummy channels are inserted to match the expected channel count without aborting the workflow.

2.2.2 Inference. Given an image volume $I \in \mathbb{R}^{X \times Y \times Z \times C}$, the model predicts per-voxel class probabilities

$$\mathbf{p}(\mathbf{x}) = (p_c(\mathbf{x}))_{c=1}^K, \quad \sum_{c=1}^K p_c(\mathbf{x}) = 1$$

and an initial hard labelmap

$$\hat{y}(\mathbf{x}) = \arg \max_{c \in \{1, \dots, K\}} p_c(\mathbf{x})$$

where $\mathbf{x} \in \mathbb{R}^3$ denotes voxel coordinates, $p_c(\mathbf{x})$ is the predicted probability for class c , and K is the total number of classes (including background).

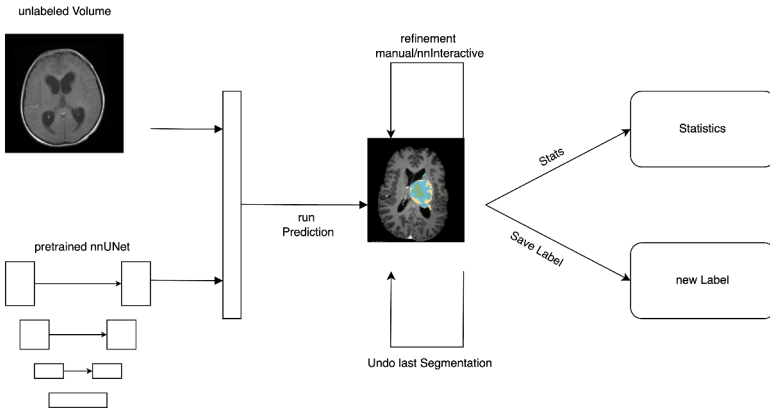


Fig. 1. Overview of the interactive segmentation pipeline. The workflow consists of two main components: (1) a computer-aided segmentation stage, where a pretrained model automatically generates an initial structure segmentation (here: tumor), and (2) a user refinement stage, where experts can iteratively refine the segmentation using either manual editing tools or nnInteractive functionalities. The refined segmentation can then either be saved for downstream use or used to compute quantitative statistics, such as the volume of the segmented structures in mm^3 .

We display task-specific label classes as defined in the model’s `dataset.json` file. For BraTS, these correspond to enhancing tumor (ET), tumor core (TC), and whole tumor (WT); for other tasks, class names are read directly from the model configuration. Softmax probabilities can optionally be saved to disk for uncertainty-aware inspection.

2.2.3 Post-processing and import. The predicted labelmap is mapped to a Slicer label entity with class-specific color/opacity tables. Optional small-component removal and morphological smoothing are available as a toggle to suppress spurious fragments. If precomputed results exist in a parallel `pred/` directory, they are loaded instantly; otherwise, on-the-fly inference is triggered on user request.

2.3 Assisted Correction

Experts refine $\hat{y}$ using nnInteractive prompts and, when desired, Slicer’s native editing tools (Fig. 1, right). We also add an undo function for reverting the most recent label change without requiring a full restart, as well as a save function for persisting the result. We model the refinement loop as a sequence of prompt-conditioned updates

$$S_t = \mathcal{F}_\theta(I, S_{t-1}, e_t), \quad t = 1, 2, \dots$$

where $S_0 = \hat{y}$ is the nnU-Net proposal, e_t is the user prompt at step t , and $\mathcal{F}_\theta$ is the nnInteractive update operator.

Prompt types. Positive/negative points (+1/−1), 2D bounding boxes, scribbles, and lasso selections. Each prompt is encoded with tool type, polarity, and 3D coordinates in scanner space.

Update cycle. On each event e_t , the client sends a compact message to the nnInteractive backend and receives an updated mask S_t which is immediately overlaid.

Class handling. Users may toggle class visibility (e.g. for BraTS, ET/TC/WT) or collapse to a single foreground for coarse editing; class-specific corrections are propagated consistently across slices.

Undo. Each accepted state S_t is pushed onto a bounded history stack $\mathcal{H}$. *Undo* pops and restores S_{t-1} . To limit memory consumption, the history stack retains at most ten states, and the oldest entry is discarded once this limit is exceeded.

Traversal integration. When a case is accepted, *Export* writes the final labelmap. *Next Volume* then advances to the next case, with resume-from-last-session supported via a lightweight manifest.

Stopping rule. The process ends when the expert accepts the segmentation.

3 Results

We evaluated the proposed workflow on 6 brain MRI cases from the BraTS-PED dataset (T1, T1ce, T2, FLAIR, per case). The unblinded segmentation approach began with the generated segmentation from the pretrained nnU-Net (BraTS-PED 2024 training dataset) and subsequently underwent iterative refinements with nnInteractive prompts. Each case was loaded using the *Next Volume* action and processed using two segmentation methods:

1. a blinded segmentation without a generated segmentation by nnU-Net. The segmentation was done using only nnInteractive prompts (positive/negative points, 2D bounding boxes, scribbles, lasso).
2. an unblinded segmentation that started from the generated segmentation by the pretrained nnU-Net and later iterative refinements with nnInteractive prompts.

We recorded per-case time (s), number of lesions, prompt count (P + N) (clicks) and difficulty (Tab. 1).

In the blinded runs, the annotator needed about 3–4 min per case (median ≈ 3.7 min) to obtain an acceptable tumor mask, with around 7 prompts in the typical case. This confirms that the baseline nnInteractive setup is already fast for small, well-contrasted lesions. When starting from the nnU-Net proposal, the effort shifted: for the easier cases (C03, C04, C09), annotation time dropped to about 1.5–2.5 min, and the number of corrective prompts was reduced to a few clicks, because the user mainly had to tighten boundaries or recover small missed foci. However, not every case benefited equally: for C05, C06, and C08, the unblinded runs were longer and required more corrections, indicating that a suboptimal proposal can also introduce extra work.

Segmentation accuracy remained nearly unchanged between the two modes. Blinded annotation achieved a median DSC of 0.850 (IQR 0.784–0.891), while unblinded annotation reached 0.888 (IQR 0.822–0.911), i.e., the median difference Δ DSC was close to zero. This is consistent with human-in-the-loop segmentation: once a trained rater is allowed to correct the mask, DSC saturates, and the main benefit of a proposal is reduced interaction, not higher final overlap. The per-case results nevertheless show that interactive refinement rescued difficult cases where the initial nnU-Net proposal was poor: for C05, the initial DSC of 0.159 was corrected to 0.911 (unblinded) and 0.913 (blinded) through user interaction. In contrast, cases that were already easy to segment manually saw little to no gain, or even a reduction (e.g., C04).

Finally, the dataset-aware traversal contributed to overall efficiency by removing non-annotation overhead: loading the next volume and, if desired, its proposal could be done from within 3D Slicer without reconfiguring paths or models. For labeling sessions that process multiple consecutive studies, this reduces friction and helps keep the annotator in a continuous workflow.

Tab. 1. Per-case results for blinded and unblinded segmented studies (6 cases: C03, C04, C05, C06, C08, C09). Initial: raw nnU-Net prediction without refinement, where Time, Prompts, and Difficulty do not apply. Difficulty: subjective rating by the annotator when using the plugin based on tumor boundary clarity and contrast of the images (–: hard, 0: moderate, +: easy). Prompts (P+N): positive clicks + negative clicks. Blinded: median time 220 s [215–280], mean time 242.5 s; median prompts 4 [4–6] + 3 [1–3], mean prompts 4.33 + 2.17. Unblinded: median time 215 s [150–285], mean time 228 s; median prompts 2 [1–3] + 5 [0–8], mean prompts 3.33 + 5.67. Dice: blinded median DSC 0.850 [0.784–0.891], mean 0.776; unblinded median DSC 0.888 [0.822–0.911], mean 0.787; Δ DSC (unblinded – blinded) median –0.002 [–0.002–0.035], mean 0.011. Volume of ground truth, V_{GT} , and segmentation, V_{Seg} , is given in cm^3 . Difficulty is a qualitative measure of how easy/hard it was to differentiate between the tumor and the background.

Case	Setting	Time (s)	Prompts (P+N)	Difficulty	V_{GT}	V_{Seg}	DSC
C03	Initial	–	–	–	16.60	19.45	0.825
	Blinded	215	6 + 1	+	16.60	15.53	0.876
	Unblinded	150	1 + 4	+	16.60	16.24	0.874
C04	Initial	–	–	–	138.66	133.02	0.930
	Blinded	225	7 + 3	+	138.66	115.91	0.784
	Unblinded	110	3 + 0	+	138.66	135.48	0.902
C05	Initial	–	–	–	17.24	180.51	0.159
	Blinded	160	1 + 0	0	17.24	16.58	0.913
	Unblinded	285	0 + 8	+	17.24	16.20	0.911
C06	Initial	–	–	–	4.95	14.82	0.170
	Blinded	280	4 + 3	–	4.95	21.13	0.367
	Unblinded	393	13 + 6	–	4.95	29.33	0.285
C08	Initial	–	–	–	84.23	78.92	0.817
	Blinded	360	4 + 3	–	84.23	87.23	0.824
	Unblinded	270	1 + 16	0	84.23	96.52	0.822
C09	Initial	–	–	–	101.67	102.10	0.892
	Blinded	215	4 + 3	0	101.67	106.40	0.891
	Unblinded	160	2 + 0	0	101.67	110.46	0.926

4 Discussion

Our goal was not to train a better segmentation network, but rather to utilize already trained networks (here, nnU-Net) in combination with nnInteractive to create a toolchain for faster annotation by experts. The experiments demonstrate that this is feasible through the integration of dataset-aware traversal, nnU-Net proposals, and nnInteractive.

Initializing the workflow with an nnU-Net proposal slightly improved the median DSC (0.850 vs. 0.888), but it achieved the intended effect of reducing annotation effort, showing a trend toward time reduction ($p=0.011$ in Wilcoxon signed-rank test with $n=6$) and corrective interactions. Given that human-in-the-loop refinement tends to saturate segmentation overlap rapidly, this shift from “better accuracy” to “less work” is expected. The benefit was most pronounced in the more challenging cases (C04, C09), where the proposal provided a spatial prior that could be efficiently

adjusted; in straightforward cases, purely interactive segmentation was already near-optimal. The dataset-aware traversal further decreased non-annotation overhead by enabling case-to-case progression within 3D Slicer, thereby maintaining workflow continuity for annotators.

Our evaluation has several constraints: (1) a single annotator limits assessment of inter-rater variability; (2) six cases are insufficient for robust statistical conclusions; (3) we evaluated only brain tumor segmentation, and generalization to other anatomies remains untested; (4) cognitive load and usability were not formally measured. Future work should address these gaps through studies with diverse annotators and segmentation tasks.

The natural next step is to close the loop by fine-tuning the base model using accepted expert corrections. This involves developing an intelligent acquisition layer that prioritizes which cases experts should label next. The system would combine multiple complementary strategies: identifying regions where the model is uncertain or ambiguous, selecting diverse batches of cases that provide non-redundant information when experts label multiple examples at once, and maintaining broad coverage across different types of cases in the dataset. Additionally, practical signals such as the effort required to correct previous cases (editing time and number of corrections needed) and the model's confidence scores can be leveraged to identify high-value cases that would most improve the model, focusing expert effort where it matters most.

Acknowledgement. This research was partially funded by the German Childhood Cancer Foundation under grant number A 2024/05 / DKS 2025.01, the Intramural Research Funding “Precision Medicine for pHGG” of the Faculty of Medicine, University of Augsburg, the Bavarian Center for Cancer Research as part of the Lighthouse “Local Therapies”, as well as by the Bavarian Ministry of Economic Affairs, Regional Development and Energy (StMWi) under grant number DIK-2310-0004// DIK0556/02.

References

1. Fedorov A, Beichel R, Kalpathy-Cramer J, Finet J, Fillion-Robin JC, Pujol S et al. 3D Slicer as an image computing platform for the Quantitative Imaging Network. *Magn Reson Imaging*. 2012;30(9):1323–41.
2. 3D Slicer. 3D slicer image computing platform. <https://www.slicer.org>. Accessed: 2025-10-27. 2025.
3. Isensee F, Rokuss M, Krämer L, Dinkelacker S, Ravindran A, Stritzke F et al. nnInteractive: redefining 3D promptable segmentation. *arXiv: 2503.08373*. 2025.
4. Diaz-Pinto A, Alle S, Ihsani A, Asad M, Nath V, Pérez-García F et al. MONAI label: a framework for AI-assisted interactive labeling of 3D medical images. *Med Image Anal*. 2022;95:103207.
5. Diaz-Pinto A, Mehta P, Alle S, Asad M, Brown R, Nath V et al. DeepEdit: deep editable learning for interactive segmentation of 3D medical images. *Proc MICCAI*. 2022:11–21.

6. Isensee F, Jaeger PF, Kohl SA, Petersen J, Maier-Hein KH. nnU-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nat Methods*. 2021;18(2):203–11.
7. de Vente C, Venkadesh KV, van Ginneken B, Sánchez CI. SlicerNNInteractive: a 3D slicer extension for nnInteractive. *arXiv: 2504.07991*. 2025.

