SlicerNNInteractive: A 3D Slicer extension for nnInteractive

Coen de Vente[1,2], Kiran Vaidhya Venkadesh[3], Andras Lasso[4], Bram van Ginneken[3], and Clara I. Sánchez[1,2]

[1] *Quantitative Healthcare Analysis (qurAI) Group, Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands*
[2] *Amsterdam UMC location University of Amsterdam, Biomedical Engineering and Physics, Amsterdam, The Netherlands*
[3] *Diagnostic Image Analysis Group (DIAG), Department of Radiology and Nuclear Medicine, Radboud UMC, Nijmegen, The Netherlands*
[4] *Laboratory for Percutaneous Surgery, School of Computing, Queen's University, Kingston, Canada*

17 May 2025

Summary

`SlicerNNInteractive` integrates `nnInteractive` (Isensee et al., 2025), a state-of-the-art promptable deep learning-based framework for 3D image segmentation, into the widely used `3D Slicer` (Kikinis et al., 2014) (https://slicer.org) platform. Our extension implements a client-server architecture that decouples the computationally intensive model inference from the client-side interface. Therefore, `SlicerNNInteractive` eliminates heavy hardware constraints on the client-side and enables better operating system compatibility than existing plugins for `nnInteractive`. Running both the client and server on a single machine is also possible, offering flexibility across different deployment scenarios. The extension provides an intuitive user interface with all interaction types available in the original framework (point, bounding box, scribble, and lasso prompts), while including a comprehensive set of keyboard shortcuts for efficient workflow.

Statement of Need

Segmentation is a cornerstone of medical image analysis. Manually acquiring segmentation labels, however, is time-consuming and expensive. Interactive segmentation tools can potentially accelerate this process and reduce costs. Recently, `nnInteractive` (Isensee et al., 2025), a deep learning-based framework allowing for fast, promptable segmentation of 3D medical images was released and was shown to substantially outperform existing approaches, such as SAM2 (Kirillov et al., 2023), SegVol (Du et al., 2024), and SAM-Med-3D (Wang et al., 2023). Alongside the `nnInteractive` model, plugins in the medical image viewers MITK (MITK Team, 2024) and Napari (Sofroniew et al., 2025) were published. However, the original authors did not make an extension available for `3D Slicer`, a widely used viewer and processing environment in medical imaging research. Furthermore, these existing plugins require substantial computational resources on the machine of the image viewer itself (an NVIDIA GPU with at least 10 GB of VRAM is recommended), as these plugins do not facilitate the deployment of the backend on a separate server. Moreover, `nnInteractive` only runs on Windows and Linux, so the image viewer cannot be run on MacOS machines.

`SlicerNNInteractive` decouples the computationally intensive `nnInteractive` inference by allowing users to configure a remote server (e.g., a node of a GPU cluster), while running the client on a machine with lower computational capabilities. This approach not only broadens platform compatibility, but also addresses the resource constraints of existing plugins, making `nnInteractive` more widely available and potentially accelerating research related to promptable segmentation.

Overview of `SlicerNNInteractive`

nnInteractive

While foundation models such as SAM (Ravi et al., 2024) and SAM2 (Kirillov et al., 2023) have shown promising interactive segmentation performance in 2D natural images, their lack of volumetric awareness and the domain shift from natural to medical data resulted in limited utility in 3D medical imaging contexts. `nnInteractive` addresses these issues through an nnUNet-based architecture (Isensee et al., 2021) with residual encoders (Isensee et al., 2024) that supports diverse interation types: point, bounding box, scribble, and lasso prompts. Trained on over 120 diverse volumetric datasets across multiple modalities (CT, MRI, PET, 3D microscopy, etc.), the framework demonstrated high accuracy and versatility. Our implementation extends this capability to `3D Slicer`.

Availability and Installation

SlicerNNInteractive is available through multiple channels. The server-side is available through Docker Hub (`docker pull coendevente/nninteractive-slicer-server:latest`), Pip (`pip install nninteractive-slicer-server`), and GitHub (https://github.com/coendevente/SlicerNNInteractive). The client-side is also available in the official 3D Slicer Extensions Manager.

Client-server Setup

SlicerNNInteractive uses a client-server setup, which decouples the computationally intensive model inference from the 3D Slicer client. The server-side and client-side communicate through FastAPI endpoints. An overview of the API is shown in Figure 1. The client maintains synchronization between the image and input segmentation in 3D Slicer. To ensure a smooth user experience, the client does not transfer this data each time before processing a prompt, but only when this data have changed.

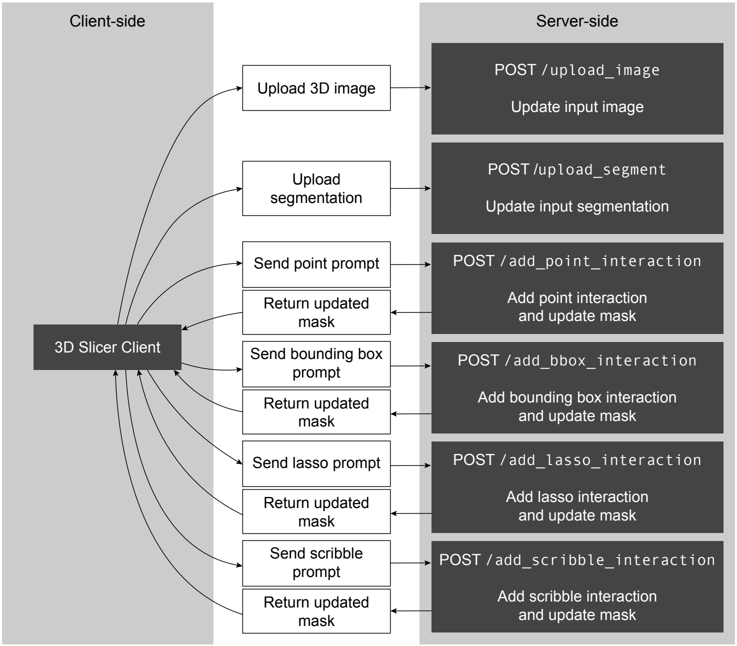
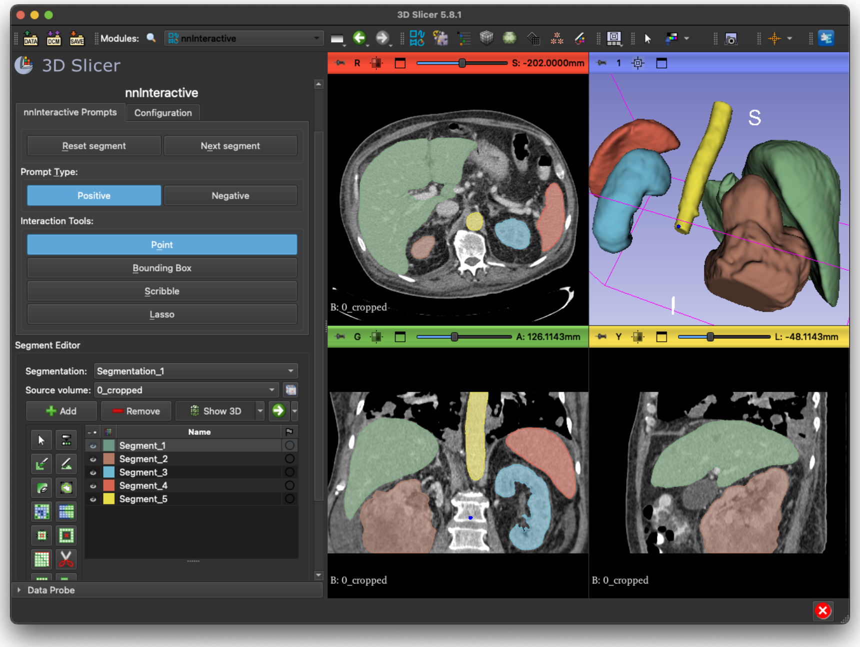

Client-side
Server-side
3D Slicer Client
Upload 3D image
Upload segmentation
Send point prompt
Return updated mask
Send bounding box prompt
Return updated mask
Send lasso prompt
Return updated mask
Send scribble prompt
Return updated mask
POST /upload_image
Update input image
POST /upload_segment
Update input segmentation
POST /add_point_interaction
Add point interaction and update mask
POST /add_bbox_interaction
Add bounding box interaction and update mask
POST /add_lasso_interaction
Add lasso interaction and update mask
POST /add_scribble_interaction
Add scribble interaction and update mask

Figure 1: API overview.

User Interface

The user interface of SlicerNNInteractive largely follows the nnInteractive Napari and MITK plugins. A screenshot of the user interface, including segmentation results, is shown in Figure 2. A video showcasing the functionalities of the extension is available here.



Segmentation: Segmentation_1
Source volume: 0_cropped

Figure 2: Screenshot of the `SlicerNNInteractive` extension.

The sidebar of the user interface consists of a menu with the tabs *nnInteractive Prompts* and *Configuration*, and the *Segment Editor*. The *Configuration* tab allows the user to change the Server URL. This URL is saved in `3D Slicer`'s settings, which will be remembered in future sessions. The *nnInteractive Prompts* menu consists of the following sections:

- **Segment buttons:** The *Reset segment* button removes all prompts from the current segment, and deletes the current segmentation on the server and client-side. The *Next segment* button creates a new empty segment in the *Segment Editor*.
- **Prompt Type:** These *Positive* and *Negative* buttons manage whether the provided prompt will be interpreted as a positive or negative prompt, respectively.
- **Interaction Tools:** The four buttons in this section activate or deactivate the interaction tools. When a prompt type is activated, the user can place the prompt in the image. When a prompt has been placed, the client synchronizes the image and the segment to the server if needed, and sends the prompt to the server. The server subsequently processes the prompt

and sends the updated segmentation back. When a prompt has been placed and processed, a new prompt of the same type can be placed immediately.

Each button in the *nnInteractive Prompts* menu has an associated keyboard shortcut, which is indicated using the underlined letters within the button text.

If a segment is selected in the *Segment Editor*, prompts will always be applied to that segment. Every time a user has switched segments, the associated segmentation is uploaded to server and used as input mask to the `nnInteractive` model. When no segment is selected, a new segment is created automatically.

Speed Measurements

We measured the interaction time of `SlicerNNInteractive` for all interaction types, in settings with lower and higher computational resources. We automatically generated user interactions in `3D Slicer`, ensuring identical prompts in all computational settings. Each measurement was repeated 10 times, for which we report the mean and standard deviation. The code for these automated tests is available on https://github.com/coendevente/SlicerNNInteractive/blob/add_timing_test/slicer_plugin/SlicerNNInteractive/SlicerNNInteractive.py.

For each interaction type, we generated automated prompts to segment the same brain tumor in the same MRI-scan. This image was `MRBrainTumor2` from `3D Slicer`'s `SampleData` extension. We also tested interaction speed for three different image sizes: `S`, `M`, and `L`. `L` was the original image, with a size of 256 × 256 × 130 voxels. `M` was a 2× in each direction downsampled version of `L`, with a size of 128 × 128 × 65 voxels. `S` was a 4× in each direction downsampled version of `L`, with a size of 64 × 64 × 32 voxels.

The client was a 14" MacBook Pro (2021, M1 Pro, Sonoma 14.5, 16 GB Memory, 8 CPU cores). In the lower computational resource experiments (`L`), the server was a Linux machine with an NVIDIA GeForce GTX 1080 Ti GPU with 11 GB VRAM (Ubuntu 22.04.1, Intel® Core™ i7-6700 CPU, 48 GB RAM). In the higher computational resource experiments (`H`), the server was a Linux machine with an NVIDIA GeForce RTX 4090 GPU with 24 GB VRAM (Ubuntu 22.04.1, 13th Gen Intel® Core™ i7-13700K CPU, 128 GB RAM).

The results of these experiments are presented in Figure 3.

Future Work

Despite the already high interaction speed, as quantitatively measured in this paper and noted by early users, our timing experiments show that a large portion of the total interaction time originates from client-side overhead – especially for larger image sizes and better computational resources. This overhead is due to processes such as updating the visualized segmentation and checking for input

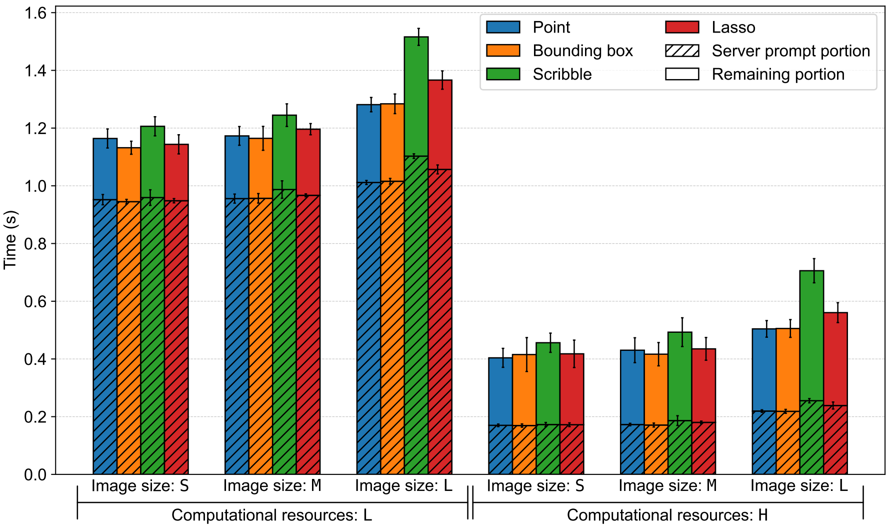


Figure 3: Interaction speed measurements of `SlicerNNInteractive`. The images with sizes `S`, `M`, and `L` were images of small, medium, and large size, respectively. Computational resources `L` and `H` are experiments using machines with lower and higher computational resources, respectively. The height of each bar and the size of each error bar represent the mean and std. dev. of 10 repeated measurements, respectively. The total height of each bar represents the total time for an interaction to be processed by our extension. The dashed portion of each bar represents the prompt server request time (including model inference, prompt upload, and mask download).

changes. Future work may focus on reducing this overhead to further improve responsiveness.

Currently, the server does not support multiple clients at the same time. `nnInteractive` releases most used VRAM instantly after having processed a prompt, so simultaneously running multiple servers – one for each individual user – on one GPU is currently a viable solution in most settings. However, in future work, we would like to implement multi-user support through one server instance to further improve resource efficiency.

Furthermore, for users with a sufficiently powerful local GPU, future versions of the plugin may support local inference directly within the Python environment of `3D Slicer`. With the upcoming release of Python 3.12 in `3D Slicer`, this functionality will become technically feasible. This would reduce the number of required installation steps even further for this group of users.

References

Du, Y., Bai, F., Huang, T., & Zhao, B. (2024). SegVol: Universal and interactive volumetric medical image segmentation. *Advances in Neural Information Processing Systems*, *37*, 110746–110783.

Isensee, F., Jaeger, P. F., Kohl, S. A., Petersen, J., & Maier-Hein, K. H. (2021). nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, *18*(2), 203–211. https://doi.org/10.1038/s41592-020-01008-z

Isensee, F., Rokuss, M., Krämer, L., Dinkelacker, S., Ravindran, A., Stritzke, F., Hamm, B., Wald, T., Langenberg, M., Ulrich, C., Deissler, J., Floca, R., & Maier-Hein, K. (2025). nnInteractive: Redefining 3D promptable segmentation. *arXiv Preprint arXiv:2503.08373*. https://doi.org/10.48550/arXiv.2503.08373

Isensee, F., Wald, T., Ulrich, C., Baumgartner, M., Roy, S., Maier-Hein, K., & Jaeger, P. F. (2024). nnU-Net revisited: A call for rigorous validation in 3D medical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 488–498. https://doi.org/10.48550/arXiv.2404.09556

Kikinis, R., Pieper, S. D., & Vosburgh, K. G. (2014). 3D slicer: A platform for subject-specific image analysis, visualization, and clinical support. In F. A. Jolesz (Ed.), *Intraoperative imaging and image-guided therapy* (pp. 277–289). Springer New York. https://doi.org/10.1007/978-1-4614-7657-3_19

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollar, P., & Girshick, R. (2023). Segment anything. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4015–4026.

MITK Team. (2024). *MITK* (Version v2024.12). https://github.com/MITK/MITK

Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K. V., Carion, N., Wu, C.-Y., Girshick, R., Dollár, P., & Feichtenhofer, C. (2024). SAM 2: Segment anything in images and videos. *arXiv Preprint arXiv:2408.00714*. https://doi.org/10.48550/arXiv.2408.00714

Sofroniew, N., Lambert, T., Bokota, G., Nunez-Iglesias, J., Sobolewski, P., Sweet, A., Gaifas, L., Evans, K., Burt, A., Doncila Pop, D., Yamauchi, K., Weber Mendonça, M., Buckley, G., Vierdag, W.-M., Royer, L., Can Solak, A., Harrington, K. I. S., Ahlers, J., Althviz Moré, D., … Zhao, R. (2025). *Napari: A multi-dimensional image viewer for python*. Zenodo. https://doi.org/10.5281/zenodo.8115575

Wang, H., Guo, S., Ye, J., Deng, Z., Cheng, J., Li, T., Chen, J., Su, Y., Huang, Z., Shen, Y., & others. (2023). SAM-Med3D: Towards general-purpose segmentation models for volumetric medical images. *arXiv Preprint arXiv:2310.15161*. https://doi.org/arXiv.2310.15161