MIST: A Simple and Scalable End-To-End 3D Medical Imaging Segmentation Framework

Adrian Celaya[1,2], Evan Lim[1], Rachel Glenn[1], Brayden Mi[1], Alex Balsells[1,2], Dawid Schellingerhout[1], Tucker Netherton[1], Caroline Chung[1], Beatrice Riviere[2], and David Fuentes[1]

[1] The University of Texas MD Anderson Cancer Center, Houston TX 77030
[2] Rice University, Houston TX 77005

Abstract. Medical imaging segmentation is a highly active area of research, with deep learning-based methods achieving state-of-the-art results in several benchmarks. However, the lack of standardized tools for training, testing, and evaluating new methods makes the comparison of methods difficult. To address this, we introduce the Medical Imaging Segmentation Toolkit (MIST), a simple, modular, and end-to-end medical imaging segmentation framework designed to facilitate consistent training, testing, and evaluation of deep learning-based medical imaging segmentation methods. MIST standardizes data analysis, preprocessing, and evaluation pipelines, accommodating multiple architectures and loss functions. This standardization ensures reproducible and fair comparisons across different methods. We detail MIST's data format requirements, pipelines, and auxiliary features and demonstrate its efficacy using the BraTS Adult Glioma Post-Treatment Challenge dataset. Our results highlight MIST's ability to produce accurate segmentation masks and its scalability across multiple GPUs, showcasing its potential as a powerful tool for future medical imaging research and development.

Keywords: Deep learning · Image segmentation · Medical imaging

1 Introduction

Medical imaging segmentation is a highly active area of research. Since its introduction in 2015, the U-Net architecture has received nearly 90,000 citations on Google Scholar [8,29]. Other architectures and frameworks like nnUNet have achieved state-of-the-art accuracy in several medical imaging benchmarks like the Brain and Tumor Segmentation (BraTS) and Medical Segmentation Decathlon (MSD) [1,11,12]. Since nnUNet's introduction in 2018, several innovative approaches for medical imaging segmentation have emerged. These innovations include new architectures like vision transformers [34] and loss functions like the boundary and generalized surface losses [6,17]. Despite these recent advances, there remains a lack of standardized tools for testing and evaluating different approaches for medical imaging segmentation. For example, several works like [3,7,10,33,39] report superior performance to nnUNet while [13] refutes these results. This inconsistency makes it difficult to evaluate and assess

claims of state-of-the-art performance for new research in deep learning-based medical imaging segmentation.

To address this inconsistency, we propose the Medical Imaging Segmentation Toolkit or MIST, a simple, modular, and end-to-end deep learning-based medical imaging segmentation framework that allows researchers to seamlessly train, test, and evaluate existing and new methods on their data in a consistent and reproducible way. MIST has a rule-based data analysis and preprocessing pipeline that it makes available to multiple architectures, loss functions, and training parameters. Additionally, MIST has a standardized and flexible evaluation pipeline that computes multiple metrics like the Dice, Hausdorff distance, and surface Dice for any user-defined segmentation class. This standardization of several pipelines allows for fair comparisons between methods and their effect on segmentation accuracy for any given dataset.

2 Methods

In this section, we describe the required data format for MIST, the different pipelines in MIST, and the auxiliary features, such as evaluation, postprocessing, and test-time inference. We also provide the specific settings we use to train on the BraTS challenge data.

2.1 Data Format

MIST uses the same file structure as the BraTS challenge datasets [15,35]. That is, MIST assumes that the train and test (optional) data directories are formatted so that each sub-directory corresponds to one patient and that each imaging modality (i.e., T1, T2, or CT) is its own NIfTI file. The labels should all be in a single NIfTI file in the same sub-directory for each patient. Once the dataset is in the correct format, the next step is to prepare a small JSON file containing the details of the dataset. These include information like the imaging modality, naming conventions for the images and masks, labels, and the final classes that we use for evaluation (i.e., labels 1, 2, and 3 for whole tumor). An example of this JSON file is provided in the documentation page here. MIST also provides support for converting datasets that are in CSV or MSD format.

2.2 Pipelines

MIST contains three main pipelines: Data Analysis, Preprocessing, and Training. These pipelines can all run at once with a single command or individually. We also provide several auxiliary pipelines for evaluation, postprocessing, and test-time inference.

Data Analysis The data analysis pipeline gathers parameters about the dataset including cropping to foreground, target spacing, patch size selection, normalization parameters, and is described further below. The results of this analysis are saved in a `config.json` file.

Cropping to Foreground Often, zeros or useless background information surround the region of interest. For example, zeros surround the region outside of the brain in the BraTS datasets, and the table and air surrounding the body in CT imaging do not provide relevant information. From a computational perspective, it is desirable to crop the images to the foreground to make them smaller. Additionally, cropping to the foreground also reduces the sparsity of the dataset. MIST crops each image by first windowing it with its 33rd and 99.5 percentile values, then using an Otsu filter to produce a foreground threshold. Note that the windowing percentiles were determined experimentally with several datasets like LiTS, MSD, NFBS, BraTS, and AMOS [2,14,28]. We then crop the image according to the binary mask resulting from this threshold (i.e., one if the voxel intensity is greater or equal to the threshold). To determine if we need to use cropping, we check to see if, on average, the resulting volume reduction is at least 20%.

Target Spacing To determine the target spacing for a dataset, we first look at the distribution of the voxel spacings along each axis. Our initial target spacing is the median spacing in each direction. However, suppose the resulting target spacing is anisotropic (i.e., the maximum and minimum spacing ratio is greater than three), then we take the 10th percentile spacing along the lowest resolution axis in that case. However, if resampling with this selected target spacing results in an example taking up more than 2 GB of memory, then MIST will print a warning to users asking them to consider coarsening the image. We set this 2 GB threshold to conserve computational resources. Larger images take longer to load from disk and require more resources to manipulate (i.e., patch extraction and augmentation).

Patch Size Selection We set the patch size for training from the median resampled image size by taking the nearest power of two less than or equal to each dimension in the median image size up to a maximum patch size. The default value for this maximum patch size is $256\times256\times256$. However, users can adjust this maximum patch size based on their knowledge of their computing resources.

Normalization Parameters If the dataset consists of MR or other non-CT images, then our windowing parameters are the 0.5 and 99.5 percentile values of either the entire image or the non-zero values of the image if, on average, the ratio of the number of non-zero to zero-valued voxels is less than 0.2. We use the mean and standard deviation with the same rules regarding the non-zero values for normalization. On the other hand, if we are dealing with CT images, then we compute the 0.5 and 99.5 percentile values, mean, and standard deviation of all of the voxels over the entire dataset corresponding to regions labeled as non-zero in the ground truth segmentation masks.

Other Checks In addition to the previous analysis steps, MIST checks to see that the header information in the images and masks agree and, if there are multiple modalities, that the headers in each image agree. If there is a disagreement

between the headers, MIST prints a warning with the patient ID to the console and excludes the example from training.

Preprocessing The preprocessing pipeline takes the information from the analysis pipeline to crop the images based on their foreground mask (if called for), reorient to right-anterior-inferior (RAI), resample to the target spacing, and window and normalize the intensity values. This pipeline can also compute the distance transform maps (DTMs) for each class in the ground truth masks or apply bias correction for MR images. Users also have the option to skip this pipeline if, for example, their NIfTI files contain preprocessed images already. Figure 1 illustrates the workflow for this pipeline.

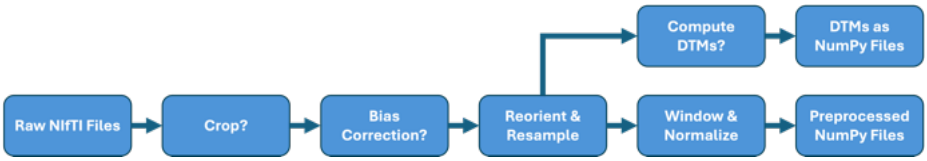

Fig. 1: Workflow for the MIST preprocessing pipeline. This preprocessing pipeline takes raw NIfTI files and outputs cropped (optional), bias corrected (optional and MR only), reoriented, resampled, windowed, and normalized NumPy files. This pipeline can also compute the DTMs for each label in the ground truth mask and output them as NumPy files.

Reorient & Resample After optionally cropping or bias correcting each image, the preprocessing pipeline will reorient each image and mask to the RAI orientation. For resampling images to the target spacing, we use third-order spline interpolation. We apply one-hot encoding and use linear interpolation to each channel for masks. If the current image spacing is anisotropic, we first resample along the low-resolution axis using nearest-neighbor interpolation and then apply the appropriate interpolation along the other two axes [11]. We implement these operations using SimpleITK to optimize the speed and computational performance of these operations [24].

Window & Normalize Before normalizing the voxel intensities in each image, the preprocessing pipeline windows each image by either the image's 0.5 and 99.5 percentile or, if we are using CT images, by the precomputed windowing parameters. After windowing, MIST applies z-score normalization with either the image's mean and standard deviation or the precomputed normalization parameters for CT images. Again, in the case that the ratio of non-zero to zero-valued voxels is less than 0.2, then the mean and standard deviation are computed from the non-zero voxels only, and we multiply the resulting images by the non-zero mask (different from foreground mask) to preserve the zero-valued voxels.

Distance Transform Maps A DTM represents the distance between each voxel and an object's closest boundary or edge. The values in a DTM are positive on the exterior, zero on the boundary, or negative in the object's interior. As a result, they can provide helpful information during training. Several loss functions and shape-aware networks use DTMs to improve the accuracy of medical imaging segmentation methods [6,16,17,21]. MIST allows users to pre-compute the DTMs for each label in the ground truth segmentation masks. Each channel in the resulting DTM represents the DTM for that label (i.e., channel one is the DTM for label one). If a label does not exist in the image, the DTM will be a 3D array with all values being the diagonal distance of the image. Additionally, MIST allows users to output normalized DTMs whose values range from -1 to 1.

Training Once preprocessing is complete, MIST uses a five-fold cross-validation to train and evaluate a model on the dataset. In other words, for each fold MIST sets aside 20% of the data as a validation dataset and uses the remaining 80% of the data for training. Users can adjust these percentages, specify a different number of folds, or even specify custom folds if, for example, they are running leave-one-institution-out cross-validation.

MIST's default behavior is to use all available GPUs. When MIST runs multi-GPU training, it uses data parallelism via PyTorch's DistributedDataParallel package [20]. Additionally, MIST uses the Nvidia Data Loading Library (DALI) to handle data loading, patch extraction, and random augmentation. This feature further optimizes its computational performance by running most of the operations on the GPUs.

Network Architectures MIST supports six different network architectures. These architectures include nnUNet (default) [11], U-Net [8,29], Swin UNETR [3], PocketNet [4], and MedNeXt [30]. MIST also provides users with options like adding deep supervision [37] or variational autoencoder regularization [32] to most of these architectures. Other regularization features like L2 and L1 regularization and gradient clipping are available for all architectures. For more details about the supported architectures (and their differences), please refer to our documentation page. Finally, MIST's modular design allows users to implement their own architectures and integrate them into MIST with minimal effort.

Loss Functions Like with the choice of network architecture, MIST allows users to choose from a variety of loss functions like Dice with Cross Entropy (default) [26], the clDice loss [31], and boundary-based loss functions like the Boundary, Hausdorff, and Generalized Surface losses [6, 16, 17]. Again, MIST's modular design allows researchers to implement custom loss functions and seamlessly use them within the MIST framework.

Other Training Options Finally, MIST gives users the flexibility to choose different optimizers like Adam (default) [18], stochastic gradient descent, and AdamW

[23]. MIST also supports several learning rate schedulers, with a constant learning rate of 0.0003 being the default. Additional features including transfer learning are included in our documentation page here.

Auxiliary Pipelines

Evaluation Once training is complete, the evaluation pipeline launches to assess the accuracy of the predictions with respect to the final classes defined by the user in their dataset JSON file. By default, these metrics are the Dice similarity coefficient and the 95th percentile Hausdorff distance (HD95). Other metrics like the surface Dice and average surface distance are also available. The output of the evaluation pipeline is a CSV file called `results.csv` where each row shows the accuracy for each class and metric for one patient. The last five rows give the mean, standard deviation, median, and 25th and 75th percentiles for all of the metrics and classes. The evaluation pipeline is also available as a stand-alone pipeline that can run on any set of predictions. With the exception of the surface Dice, all of our metrics are in line with Metrics Reloaded [25].

Postprocessing Postprocessing is available as an optional stand-alone pipeline. This pipeline allows users to specify which labels to apply a user-specified set of operations. These operations include the removal of small objects, only keeping the top-k components (adjustable value of k), morphological cleaning, and filling holes with a specified value. Once the postprocessing pipeline finishes, the evaluation pipeline launches and compares the change in accuracy to the baseline results. MIST computes this change in accuracy as a weighted average of the specified metrics. Note that MIST applies postprocessing to the discrete predictions, not the softmax probabilities.

Test-Time Inference Finally, MIST provides a stand-alone inference pipeline that takes as input one or more MIST models and a CSV or JSON file with the paths to test-time data. This pipeline writes test-time predictions to a user-specified output directory.

2.3 BraTS 2024 Training Protocols

Our choice of architecture is the Pocket nnUNet with deep supervision (two supervision heads) and with residual convolution blocks [4,11]. MIST automatically selects a patch size of 128×128×128. We use eight NVIDIA H100 GPUs with a batch size of 32 uniformly distributed across the GPUs. Additionally, we use L2 regularization with a penalty parameter equal to 1e-5. Our choice of loss function is the Dice with Cross Entropy loss. While the default number of epochs per fold for MIST is 1,000, we train for 15,000 epochs per fold. We use a cosine learning rate schedule with an initial learning rate of 0.001 [22]. Within each fold, we set aside 2.5% of the training data (27 patients) as a validation set to select the best model. Once training is complete, MIST runs inference

on the challenge validation data using test time augmentation (flipping along each axis and averaging each prediction) for each model. Inference uses sliding windows with an overlap of 0.5 with Gaussian blending ($\sigma = 0.125$). The predictions from all five models are averaged to produce a final prediction. All other options and hyperparameters are left at their default values. Please refer to MIST's documentation for these values.

3 Results

3.1 Accuracy

We use MIST and the parameters described in Section 2.3 to perform a five-fold cross-validation (CV) with the BraTS Adult Glioma Post-Treatment Challenge data [35]. Table 1 gives the results of this five-fold CV in terms of the Dice score and HD95. Tables 2 and 3 give the accuracy of the predictions in the validation set in terms of the same metrics and their lesion-wise versions. Table 4 gives the accuracy of the predictions in the test set for the lesion-wise metrics.

Class	Labels	Dice		HD95 (mm)	
		Mean (Std.)	Median	Mean (Std.)	Median
NETC	1	0.8068 (0.3173)	1.0000	27.492 (87.446)	0.0000
SNFH	2	0.8978 (0.1027)	0.9262	5.0248 (15.888)	1.8660
ET	3	0.8030 (0.2823)	0.9193	26.638 (84.388)	1.0000
RC	4	0.7754 (0.2994)	0.9066	30.955 (88.628)	2.2361
TC	1, 3	0.7962 (0.2870)	0.9183	27.983 (85.971)	1.7321
WT	1, 2, 3	0.9063 (0.0100)	0.9328	5.0210 (15.846)	2.0000

Table 1: Dice and 95th percentile Hausdorff distances for each segmentation class after a five-fold cross-validation on the BraTS Adult Glioma Post-Treatment Challenge training data.

3.2 Computational Performance

In addition to the accuracy of our predictions, we also report the scalability of MIST for different numbers of A100 and H100 GPUs for multiple batch sizes. Figures 2 and 3 show these results for A100 and H100 GPUs, respectively. In these figures, we see that MIST effectively scales for multiple GPUs. This scaling is especially evident for the H100 GPUs. It is also important to note that we observe a roughly two times speed up with the H100 GPUs, which is in line with what Nvidia reports in their comparisons between the two types of graphics cards.

Class	Labels	Dice	HD95 (mm)
NETC	1	0.7010	58.165
SNFH	2	0.9208	6.6536
ET	3	0.7427	27.950
RC	4	0.6841	49.321
TC	1, 3	0.7379	28.514
WT	1, 2, 3	0.9257	6.6367

Table 2: Validation accuracy in terms of the Dice and 95th percentile Hausdorff distances for each segmentation class in the BraTS Adult Glioma Post-Treatment Challenge validation data. Note that we only report the mean for each metric here since the competition evaluation pipeline does not provide the other statistics in this case.

Class	Labels	Lesion-Wise Dice		Lesion-Wise HD95 (mm)	
		Mean (Std.)	Median	Mean (Std.)	Median
NETC	1	0.7561 (0.3711)	1.0000	53.029 (115.98)	0.0000
SNFH	2	0.8504 (0.2063)	0.9449	30.391 (67.397)	1.4142
ET	3	0.7253 (0.3195)	0.8624	46.885 (100.89)	2.0000
RC	4	0.7024 (0.3595)	0.8834	54.809 (116.18)	3.5821
TC	1, 3	0.7066 (0.3147)	0.8361	49.134 (98.166)	3.5056
WT	1, 2, 3	0.8602 (0.1982)	0.9451	29.309 (66.096)	1.9126

Table 3: Lesion-wise validation accuracy in terms of the Dice and 95th percentile Hausdorff distances for each segmentation class in the BraTS Adult Glioma Post-Treatment Challenge validation data.

Class	Labels	Lesion-Wise Dice		Lesion-Wise HD95 (mm)	
		Mean (Std.)	Median	Mean (Std.)	Median
NETC	1	0.8125 (0.3146)	1.0000	30.461 (90.135)	0.0000
SNFH	2	0.8774 (0.1718)	0.9528	18.213 (49.103)	1.0000
ET	3	0.7767 (0.3028)	0.9281	42.190 (98.363)	1.0000
RC	4	0.7442 (0.3372)	0.9183	47.834 (108.14)	2.0000
TC	1, 3	0.7764 (0.2978)	0.9178	42.697 (97.131)	2.0000
WT	1, 2, 3	0.8761 (0.1769)	0.9509	20.261 (52.819)	1.7321

Table 4: Lesion-wise validation accuracy in terms of the Dice and 95th percentile Hausdorff distances for each segmentation class in the BraTS Adult Glioma Post-Treatment Challenge test data.

4 Discussion

Our results indicate that models trained using MIST can produce accurate adult post-operative gliomas, achieving median Dice scores of at least 0.9 for all of the segmentation classes in our five-fold CV. We choose to submit the raw output of our MIST pipeline to the BraTS 2024 competition, omitting the use of

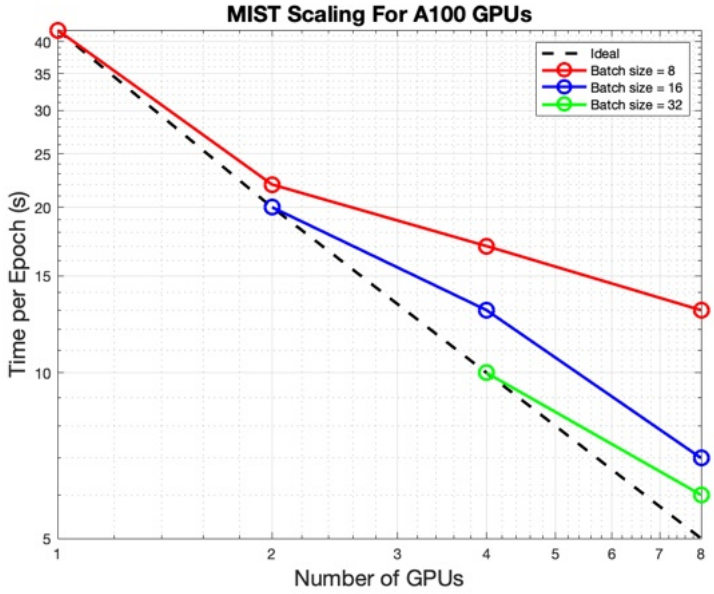

Fig. 2: Time per epoch for different batch sizes and numbers of A100 GPUs. Here, we see that MIST achieves better scaling with larger batch sizes for A100 GPUs.

post-processing. However, future submissions to the continuous evaluation and competitions will explore different post-processing strategies with MIST by using its built-in post-processing pipeline. Indeed, previous BraTS competitions cite replacing small objects belonging to the ET class with the SNFH label as beneficial [12]. Additionally, we may see improvements in accuracy by using information from DTMs using boundary-based losses like the Generalized Surface Loss. Finally, different architectures like the multi-grid inspired FMG-Net or vision transformer-based architectures like Swin UNETR are also available on MIST [3,5]. These architectures may also improve our results.

Future work will include an ablation study of the parameters that we selected for the BraTS dataset, like the use of deep supervision, residual convolution blocks, L2 regularization, and pocket architectures. Additionally, an in-depth comparison between other frameworks like nnUNet, MedNeXt, and Swin UNETR for this dataset will be an objective of future work. For an in-depth comparison of these frameworks for a liver segmentation dataset, please refer to [27].

MIST is able to utilize multiple GPUs to speed up training efficiently. This efficient use of multiple GPUs is especially true for H100s, where we see a nearly six times speed up when going from one to eight GPUs. The ability to take advantage of multiple GPUs and compute nodes effectively will be essential to

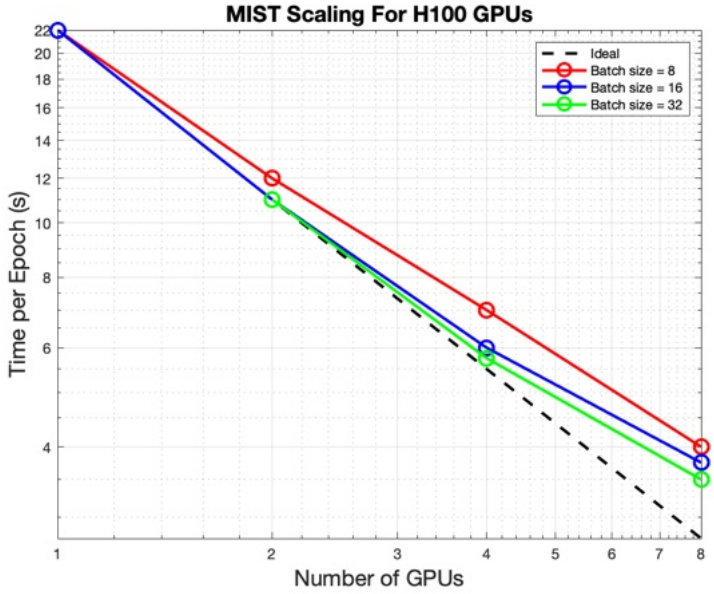

Fig. 3: Time per epoch for different batch sizes and numbers of H100 GPUs. Here, we see that MIST achieves near optimal scaling for all batch sizes.

keep training times manageable as larger datasets like TotalSegmentor become available [9,38]. As the size and quality of publicly available 3D medical imaging datasets increase, the creation of foundation models for medical imaging segmentation will be beneficial to speed up training through transfer learning [19,36]. MIST already supports transfer learning with other MIST models. Given this capability and its scalability, MIST can be instrumental in the creation of future foundation models.

In conclusion, our results show that MIST is capable of achieving accurate segmentation results while also scaling well to multiple GPUs by effectively taking advantage of technologies like PyTorch's DDP and Nvidia's DALI loader. MIST's simplicity, accuracy, and scalability have the potential to make MIST a valuable tool in the development of foundation models for medical imaging tasks like segmentation and radiation treatment planning. MIST's modular design makes it easy for researchers to add new loss functions or architectures to its pipelines. This modularity and easy integration into MIST's existing pipelines allow for new research to be evaluated and compared to existing methods fairly and consistently. MIST is still under active development. We encourage researchers and the broader medical imaging community to contribute to its development by using it, providing feedback, or contributing new features.

5 Availability & Attribution

MIST is an open-source package (Apache 2.0 license) and is available on GitHub or PyPI. Please cite this manuscript, [4], and [5] if you use MIST for your own publications.

6 Acknowledgments

The Department of Defense supports Adrian Celaya through the National Defense Science & Engineering Graduate Fellowship Program. This research was partially supported by the Tumor Measurement Initiative through the MD Anderson Strategic Research Initiative Development (STRIDE), NSF-2111147, NSF-2111459, and NIH R01CA195524

References

1. Antonelli, M., Reinke, A., Bakas, S., Farahani, K., Kopp-Schneider, A., Landman, B.A., Litjens, G., Menze, B., Ronneberger, O., Summers, R.M., et al.: The medical segmentation decathlon. Nature communications **13**(1), 4128 (2022)
2. Bilic, P., Christ, P.F., Vorontsov, E., Chlebus, G., Chen, H., Dou, Q., Fu, C.W., Han, X., Heng, P.A., Hesser, J., et al.: The liver tumor segmentation benchmark (LiTS). arXiv preprint arXiv:1901.04056 (2019)
3. Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., Wang, M.: Swin-unet: Unet-like pure transformer for medical image segmentation. In: European conference on computer vision. pp. 205–218. Springer (2022)
4. Celaya, A., Actor, J.A., Muthusivarajan, R., Gates, E., Chung, C., Schellingerhout, D., Riviere, B., Fuentes, D.: Pocketnet: A smaller neural network for medical image analysis. IEEE transactions on medical imaging **42**(4), 1172–1184 (2022)
5. Celaya, A., Riviere, B., Fuentes, D.: FMG-net and W-net: multigrid inspired deep learning architectures for medical imaging segmentation. arXiv preprint arXiv:2304.02725 (2023)
6. Celaya, A., Riviere, B., Fuentes, D.: A generalized surface loss for reducing the Hausdorff distance in medical imaging segmentation. arXiv preprint arXiv:2302.03868 (2023)
7. Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A.L., Zhou, Y.: Transunet: Transformers make strong encoders for medical image segmentation. arXiv preprint arXiv:2102.04306 (2021)
8. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3d u-net: learning dense volumetric segmentation from sparse annotation. In: International conference on medical image computing and computer-assisted intervention. pp. 424–432. Springer (2016)
9. D'Antonoli, T.A., Berger, L.K., Indrakanti, A.K., Vishwanathan, N., Weiß, J., Jung, M., Berkarda, Z., Rau, A., Reisert, M., Küstner, T., et al.: TotalSegmentator MRI: Sequence-independent segmentation of 59 anatomical structures in MR images. arXiv preprint arXiv:2405.19492 (2024)
10. Hatamizadeh, A., Nath, V., Tang, Y., Yang, D., Roth, H.R., Xu, D.: Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images. In: International MICCAI brainlesion workshop. pp. 272–284. Springer (2021)

11. Isensee, F., Jaeger, P.F., Kohl, S.A., Petersen, J., Maier-Hein, K.H.: nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. Nature methods **18**(2), 203–211 (2021)
12. Isensee, F., Jäger, P.F., Full, P.M., Vollmuth, P., Maier-Hein, K.H.: nnU-Net for brain tumor segmentation. In: Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 6th International Workshop, BrainLes 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4, 2020, Revised Selected Papers, Part II 6. pp. 118–132. Springer (2021)
13. Isensee, F., Wald, T., Ulrich, C., Baumgartner, M., Roy, S., Maier-Hein, K., Jaeger, P.F.: nnu-net revisited: A call for rigorous validation in 3d medical image segmentation. arXiv preprint arXiv:2404.09556 (2024)
14. Ji, Y., Bai, H., Ge, C., Yang, J., Zhu, Y., Zhang, R., Li, Z., Zhanng, L., Ma, W., Wan, X., et al.: Amos: A large-scale abdominal multi-organ benchmark for versatile medical image segmentation. Advances in neural information processing systems **35**, 36722–36732 (2022)
15. Karargyris, A., Umeton, R., Sheller, M.J., Aristizabal, A., George, J., Wuest, A., Pati, S., Kassem, H., Zenk, M., Baid, U., et al.: Federated benchmarking of medical artificial intelligence with MedPerf. Nature machine intelligence **5**(7), 799–810 (2023)
16. Karimi, D., Salcudean, S.E.: Reducing the Hausdorff distance in medical image segmentation with convolutional neural networks. IEEE Transactions on medical imaging **39**(2), 499–513 (2019)
17. Kervadec, H., Bouchtiba, J., Desrosiers, C., Granger, E., Dolz, J., Ayed, I.B.: Boundary loss for highly unbalanced segmentation. Medical image analysis **67**, 101851 (2021)
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
19. Lee, H.H., Gu, Y., Zhao, T., Xu, Y., Yang, J., Usuyama, N., Wong, C., Wei, M., Landman, B.A., Huo, Y., et al.: Foundation models for biomedical image segmentation: A survey. arXiv preprint arXiv:2401.07654 (2024)
20. Li, S., Zhao, Y., Varma, R., Salpekar, O., Noordhuis, P., Li, T., Paszke, A., Smith, J., Vaughan, B., Damania, P., et al.: Pytorch distributed: Experiences on accelerating data parallel training. arXiv preprint arXiv:2006.15704 (2020)
21. Li, S., Zhang, C., He, X.: Shape-aware semi-supervised 3D semantic segmentation for medical images. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part I 23. pp. 552–561. Springer (2020)
22. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016)
23. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
24. Lowekamp, B.C., Chen, D.T., Ibáñez, L., Blezek, D.: The design of SimpleITK. Frontiers in neuroinformatics **7**, 45 (2013)
25. Maier-Hein, L., Reinke, A., Godau, P., Tizabi, M.D., Buettner, F., Christodoulou, E., Glocker, B., Isensee, F., Kleesiek, J., Kozubek, M., et al.: Metrics reloaded: recommendations for image analysis validation. Nature methods **21**(2), 195–212 (2024)
26. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 fourth international conference on 3D vision (3DV). pp. 565–571. Ieee (2016)

27. Patel, N., Celaya, A., Eltaher, M., Glenn, R., Savannah, K.B., Brock, K.K., Sanchez, J.I., Calderone, T.L., Cleere, D., Elsaiey, A., et al.: Training robust t1-weighted magnetic resonance imaging liver segmentation models using ensembles of datasets with different contrast protocols and liver disease etiologies. Scientific Reports **14**(1), 20988 (2024)
28. Puccio, B., Pooley, J.P., Pellman, J.S., Taverna, E.C., Craddock, R.C.: The preprocessed connectomes project repository of manually corrected skull-stripped T1-weighted anatomical MRI data. GigaScience **5**(1) (10 2016). https://doi.org/10.1186/s13742-016-0150-5, https://doi.org/10.1186/s13742-016-0150-5, data available for download at: http://preprocessed-connectomes-project.org/NFB_skullstripped/
29. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
30. Roy, S., Koehler, G., Ulrich, C., Baumgartner, M., Petersen, J., Isensee, F., Jaeger, P.F., Maier-Hein, K.H.: Mednext: transformer-driven scaling of convnets for medical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 405–415. Springer (2023)
31. Shit, S., Paetzold, J.C., Sekuboyina, A., Ezhov, I., Unger, A., Zhylka, A., Pluim, J.P., Bauer, U., Menze, B.H.: cldice-a novel topology-preserving loss function for tubular structure segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16560–16569 (2021)
32. Tang, J., Li, T., Shu, H., Zhu, H.: Variational-autoencoder regularized 3D MultiResUNet for the BraTS 2020 brain tumor segmentation. In: Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 6th International Workshop, BrainLes 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4, 2020, Revised Selected Papers, Part II 6. pp. 431–440. Springer (2021)
33. Tang, Y., Yang, D., Li, W., Roth, H.R., Landman, B., Xu, D., Nath, V., Hatamizadeh, A.: Self-supervised pre-training of swin transformers for 3d medical image analysis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20730–20740 (2022)
34. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
35. de Verdier, M.C., Saluja, R., Gagnon, L., LaBella, D., Baid, U., Tahon, N.H., Foltyn-Dumitru, M., Zhang, J., Alafif, M., Baig, S., et al.: The 2024 brain tumor segmentation (BraTS) challenge: Glioma segmentation on post-treatment MRI. arXiv preprint arXiv:2405.18368 (2024)
36. Wang, G., Wu, J., Luo, X., Liu, X., Li, K., Zhang, S.: Mis-fm: 3d medical image segmentation using foundation models pretrained on a large-scale unannotated dataset. arXiv preprint arXiv:2306.16925 (2023)
37. Wang, L., Lee, C.Y., Tu, Z., Lazebnik, S.: Training deeper convolutional networks with deep supervision. arXiv preprint arXiv:1505.02496 (2015)
38. Wasserthal, J., Breit, H.C., Meyer, M.T., Pradella, M., Hinck, D., Sauter, A.W., Heye, T., Boll, D.T., Cyriac, J., Yang, S., et al.: TotalSegmentator: robust segmentation of 104 anatomic structures in CT images. Radiology: Artificial Intelligence **5**(5) (2023)
39. Zhou, H.Y., Guo, J., Zhang, Y., Yu, L., Wang, L., Yu, Y.: nnformer: Interleaved transformer for volumetric segmentation. arXiv preprint arXiv:2109.03201 (2021)