Automated segmentation of lesions from 3D PET/CT in MICCAI AutoPET 2023 Challenge

Andriy Myronenko[0000−0001−8713−7031], Dong Yang[0000−0002−5031−4337], Yufan He[0000−0003−4095−9104], and Daguang Xu[0000−0002−4621−881X]

NVIDIA
amyronenko@nvidia.com

Abstract. Automated lesion segmentation from 3D PET/CT allows for quantitative disease analysis and monitoring. In this work, we describe our solution to the AutoPET 2023[1] challenge. We use an automated segmentation method Auto3DSeg[2] available in MONAI[3]. Our method achieves a 78% average Dice score, based on our 5-fold cross-validation random data split.

Keywords: Auto3DSeg · MONAI · Segmentation · PET/CT.

1 Introduction

Computer tomography (CT) provides valuable insights into human body anatomy, with detailed information of organs and bones (see Fig. 3). Even though CT modality can be effectively used for tumor analysis, it has a low soft tissue contrast, where some lesion intensity patterns are indistinct. Complimentary, 3D Positron Emission Tomorgraphy (PET) has low spatial resolution, but allows to effectively highlight areas of hyperactivity, which are often associated with lesions (see Fig. 2). Fluorodeoxyglucose (FDG) is the most widely used PET tracer in an oncological setting reflecting glucose consumption of tissues. Combining 3D PET and CT scans allows for more accurate analysis and diagnosis.

Automated Lesion Segmentation in PET/CT (AutoPET23) challenge provides a platform for researchers to develop and evaluate their solutions for lesion segmentation from 3D PET and 3D CT paired images [3,2]. AutoPET23 provides 1014 cases for training (which originates from 900 patients). The hidden test set consists of 200 cases. The evaluating metrics include a combination of DICE, false positive and false negative volume scores.

Our solution is based on Auto3DSeg from MONAI [1], using an ensemble of 20 SegResNet models [8][4], which we describe in Sec. 2.

[1] https://autopet-ii.grand-challenge.org/
[2] https://monai.io/apps/auto3dseg
[3] https://github.com/Project-MONAI/MONAI
[4] https://docs.monai.io/en/latest/networks.html#segresnetds

2 Method

We implemented our approach with MONAI [1] using Auto3DSeg open-source project. Auto3DSeg is an automated solution for 3D medical image segmentation, utilizing open source components in MONAI, offering both beginner and advanced researchers the means to effectively develop and deploy high-performing segmentation algorithms.

A baseline segmentation training with Auto3DSeg can be achieved with the following command:

```
#!/bin/bash
python -m monai.apps.auto3dseg AutoRunner run \
    --input=./input.yaml --algos=segresnet
```

where a user provided input configuration (input.yaml) include:

```
# This is the YAML file "input.yaml"
modality: CT
datalist: "./dataset.json"
dataroot: "/data/autopet23"
extra_modalities: {pet : pet}
```

When running this command, Auto3DSeg will analyze the dataset, generate hyperparameter configurations for several supported algorithms, train them, and produce inference and ensemble. The system will automatically scale to all available GPUs and also supports multi-node training. The 3 minimum user options (in input.yaml) are data modality, location of the dataset (dataroot), and the list of input filenames with an associated fold number (datalist). In addition to a 3D CT, AutoPET23 challenge provides a 3D PET image, which should be indicated in the config in the "extra_modalities". We generate the 5-fold equal split assignments randomly.

Currently, the default Auto3DSeg setting trains three 3D segmentation algorithms: SegResNet [8], DiNTS [5] and SwinUNETR [4] with their unique training recipes. SegResNet and DiNTS are convolutional neural network (CNN) based architectures, whereas SwinUNETR is based on transformers [9]. Here we used only SegResNet for simplicity and describe its training procedure in this paper to be self-inclusive. At inference, we ensemble 20 checkpoints of SegResNet (5-folds trained 4 times).

2.1 Data

AutoPET23 provides 1014 PET/CT cases for training (originating from 900 patients). The hidden test set includes 200 PET/CT cases. We split the training set into 5-folds equally, and use the following on-the-fly data normalization during training:

- 3D CT image is normalized to $[0, 1]$ intensity interval from a $[-250, 250]$ input CT interval.

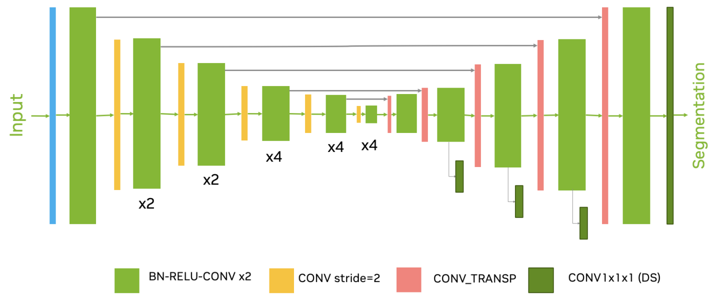

Fig. 1. SegResNet network configuration. The network uses repeated ResNet blocks with batch normalization and deep supervision

- 3D PET is clipped to $[0, 10]$, to reduce the potentially high values. We use the SUV (standardized uptake values) PET representation, provided by the organizers.
- We use the provided 3D CT and PET images that are already resampled to 2x2x3 mm common resolution, without any additional resampling.

Auto3DSeg can cache all data in RAM during the first training epoch to speed up training. To reduce the cache size required, the data is automatically cropped to an approximate foreground CT region (a bounding box around positive values).

No external datasets nor pretrained model weights were used, even though it was permitted in the AutoPET23 challenge.

2.2 Proposed Method

The underlying network architecture is SegResNet based on [8] from MONAI[5]. It is an asymmetric encode-decoder based semantic segmentation network, a U-net alike convolutional neural network with deep supervision (see Figure 1).

The encoder part uses residual network blocks, and includes 5 stages of 1, 2, 2, 4, 4 blocks respectively. It follows a common CNN approach to downsize image dimensions by 2 progressively and simultaneously increase feature size by 2. All convolutions are $3 \times 3 \times 3$ with an initial number of filters equal to 32. The decoder structure is similar to the encoder one, but with a single block per each spatial level. Each decoder level begins with upsizing with transposed convolution: reducing the number of features by a factor of 2 and doubling the spatial dimension, followed by the addition of encoder output of the equivalent spatial level.

We use a combined Dice-CE loss[7,6][6], and sum it over all deep-supervision sublevels[7]:

[5] https://docs.monai.io/en/latest/networks.html#segresnetds
[6] https://docs.monai.io/en/stable/losses.html#diceceloss
[7] https://github.com/Project-MONAI/MONAI/blob/dev/monai/losses/ds_loss.py

$$Loss = \sum_{i=0}^{4} \frac{1}{2^i} Loss(pred, target^{\downarrow}) \tag{1}$$

where the weight $\frac{1}{2^i}$ is smaller for each sublevel (smaller image size) i. The target labels are downsized (if necessary) to match the corresponding output size using nearest neighbor interpolation.

We use the AdamW optimizer with an initial learning rate of $2e^{-4}$ and decrease it to zero at the end of the final epoch using the Cosine annealing scheduler[8] with 3 warmup epochs. We use batch size of 1 (per each of 8 GPUs), random crop of 192×192×384, weight decay of $1e^{-5}$, and optimize for 1000 epochs. We use several augmentations including random rotation and scale, random flips (all axes), random smoothing, noise, and intensity shift/scale. The input has 2 channels (concatenated CT and PET images), and the output has 1 channel (sigmoid based binary segmentation).

2.3 Inference

Our inference is an ensemble of 20 SegResNet model checkpoints (5-folds trained 4 times). Each inference uses a sliding window strategy (same ROI size of 192×192×384 as used during training) with an overlap of 0.625. Auto3DSeg uses SlidingWindowInfererAdapt() class from MONAI [9], which automatically manages sliding inference stitching, as well adaptively manages GPU memory (e.g. to offload results fully or partially to RAM). This helps to prevent GPU OOM for large images, especially sincere AutoPET23 has a hard limit requirement of max 16GB GPU memory.

We use post-processing of the final segmentation result in attempt to reduce both false positives and false negatives. To reduce false positives, we used a binary morphology to merge connected components that are very close to each other. And to reduce false negatives, we search the predicted probability map to find potential candidate lesions to append, that are below the 0.5 threshold, but have high SUV (PET) values.

The development environments used for training is presented in Table 1, and was done inside of a docker "nvidia/pytorch:23.06-py3", including PyTorch 2.1 and MONAI 1.2. The training protocol is shown in Table 2.

3 Results

Based on our random 5-fold split, the average Dice scores per fold are shown in Table 3 for 1 representative round of training (out of 4 total). When computing the Dice score, only the positive cases (cases with at least some lesions) were used during averaging.

A visualization of the ground truth labels and the predicted results of one of the cases is shown in Fig. 2 (PET) and 3 (CT).

[8] https://docs.monai.io/en/latest/optimizers.html#warmupcosineschedule
[9] https://docs.monai.io/en/latest/_modules/monai/inferers/inferer.html

Table 1. Development environments.

Docker	nvcr.io/nvidia/pytorch:23.06-py3
System	Ubuntu 22.04.2 LTS
RAM	400G
GPU (number and type)	8x NVIDIA V100 32G
CUDA version	12.1
Programming language	Python 3.10
Deep learning framework	MONAI 1.2, PyTorch 2.1

Table 2. Training protocols.

Network initialization	Random
Batch size	8
Patch size	192×192×384
Total epochs	1000
Optimizer	AdamW
Initial learning rate (lr)	2e-4
Lr decay schedule	Cosine
Loss function	Dice-CE loss
Number of model parameters	87M

Table 3. Dice score using our 5-fold data random split. Each fold corresponds to the best checkpoint model trained during 5-fold cross-validation .

fold 0	fold 1	fold 2	fold 3	fold 4	Average
77.85	78.61	77.43	77.35	79.62	78.17±0.85

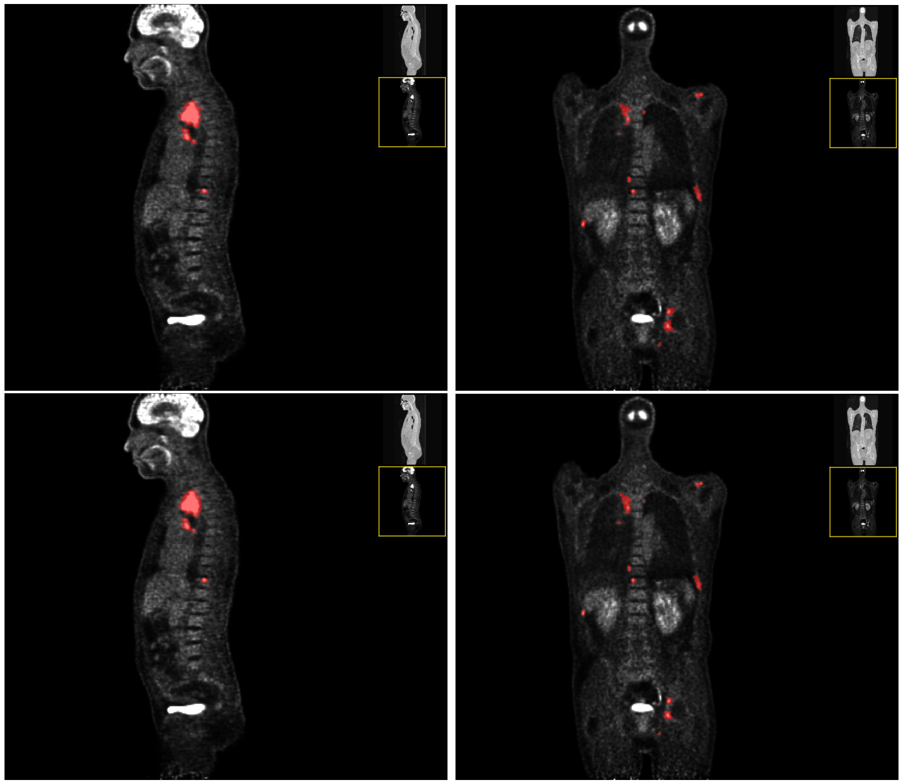

Fig. 2. A 3D PET visualization of an example case from the AutoPET23 training set: top row - ground truth, bottom row - predicted result.

4 Conclusion

In this work, we describe our submission to the AutoPET23 challenge. Our solution utilizes Auto3DSeg from MONAI, and on the 5-fold cross-validation split, it achieves a Dice score of 78% on average.

References

1. Cardoso, M.J., Li, W., Brown, R., Ma, N., Kerfoot, E., Wang, Y., Murrey, B., Myronenko, A., Zhao, C., Yang, D., Nath, V., He, Y., Xu, Z., Hatamizadeh, A., Myronenko, A., Zhu, W., Liu, Y., Zheng, M., Tang, Y., Yang, I., Zephyr, M., Hashemian, B., Alle, S., Darestani, M.Z., Budd, C., Modat, M., Vercauteren, T., Wang, G., Li, Y., Hu, Y., Fu, Y., Gorman, B., Johnson, H., Genereaux, B., Erdal, B.S., Gupta, V., Diaz-Pinto, A., Dourson, A., Maier-Hein, L., Jaeger, P.F., Baumgartner, M., Kalpathy-Cramer, J., Flores, M., Kirby, J., Cooper, L.A.D., Roth, H.R., Xu, D., Bericat, D., Floca, R., Zhou, S.K., Shuaib, H., Farahani, K., Maier-Hein, K.H., Aylward, S., Dogra, P., Ourselin, S., Feng, A.: MONAI: An open-source framework for deep learning in healthcare (2022) 1, 2

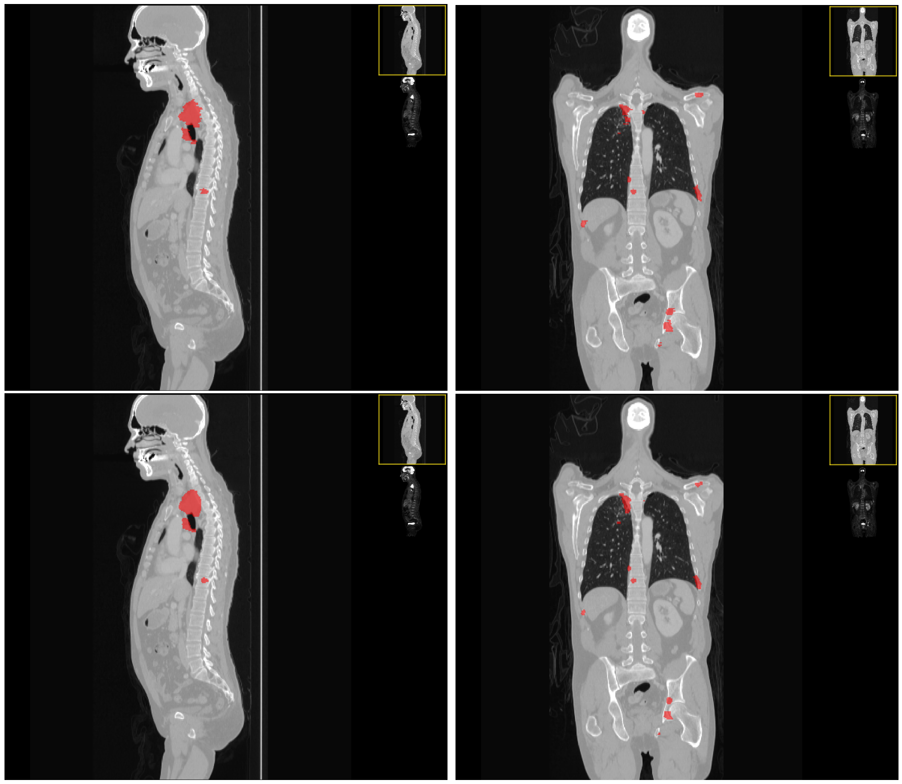

Fig. 3. A 3D CT visualization of an example case from the AutoPET23 training set: top row - ground truth, bottom row - predicted result.

2. Gatidis, S., Früh, M., et al.: The autoPET challenge: Towards fully automated lesion segmentation in oncologic PET/CT imaging. https://doi.org/10.21203/rs.3.rs-2572595/v1 1
3. Gatidis, S., Hepp, T., Früh, M., La Fougère, C., Nikolaou, K., Pfannenberg, C., Schölkopf, B., Küstner, T., Cyran, C., Rubin, D.: A whole-body FDG-PET/CT Dataset with manually annotated Tumor Lesions. Scientific Data **9** (2022) 1
4. 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) 2
5. He, Y., Yang, D., Roth, H., Zhao, C., Xu, D.: Dints: Differentiable neural network topology search for 3d medical image segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5841–5850 (2021) 2
6. 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) 3
7. Milletari, F., Navab, N., Ahmadi, S.: V-Net: Fully convolutional neural networks for volumetric medical image segmentation. CoRR (2016), http://arxiv.org/abs/

1606.04797 3
8. Myronenko, A.: 3D MRI brain tumor segmentation using autoencoder regularization. In: International MICCAI Brainlesion Workshop. pp. 311–320. Springer (2018) 1, 2, 3
9. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. CoRR **abs/1706.03762** (2017), http://arxiv.org/abs/1706.03762 2