Domain Adaptation and Semantic Segmentation using UVCGAN and nnUNet

Shaikh Muhammad Uzair Noman[1], AmirhosseinVahidi[2], Mina Rezaei[3]

[1,2,3] Ludwig-Maximilians-Universität München, Germany

Abstract. This project is for the crossMoDA challenge segmentation task. The challenge is about the unsupervised domain adaption segmentation tasks. To address the problem, we divide the project into two phases. First, we used one of the cycleGAN variants to translate T1 images to synthetic T2. Then, in the second phase, we used different variants of nnunet for the segmentation of the synthetic T2 images. Furthermore, we used the trained model for self-training to produce pseudo-labels for unlabeled T2 images. In the end, we trained a new model with synthetic images and T2 images with pseudo-labels.

Keywords: Cross-modality, Segmentation, CycleGAN.

1 Method

1.1 Domain adaptation

Model

CycleGAN is one of the prominent models for image translation tasks. The vision transformer model (Vit) is famous for having more generalization power than the convolutional neural network. UVCGAN [1] takes advantage of these two models by using Vit-UNet as a generator for CycleGAN [2].

1.2 Image segmentation

Model

We used different variants of nnUNet [3]. The details of different variants you can find in the following:

1.3 Self-training

Self-training idea is motivated by Entropy minimization over different classes. But the minimum entropy is not the best choice around the decision boundaries. To overcome the hard separation of entropy minimization near boundaries and have a smoother transition, we need to add more regularization (e.g., data augmentation) [4].

2 Implementation

2.1 Hyper-parameters

Table 1 UVCGAN hyper-parameters

UVCGAN	Learning-rate	1e-4
	Batch size	32
	Number of steps per epoch	1000
	Resize	256
	Epochs	200
	Center crop	224
	Numbers of GPUs	8
	Training time	23hours and 53 minutes
	Generator	Vit-unet-12
	Model	CycleGAN

We tried different training routines to see which performs better. Since Vestibular schwannoma (VS) and cochlea were present in atmost 15 slices of the 3D images and also we have labels for only 210 images. The huge imbalance between the non-presence and presence of VS and cochlea. We heuristically tried to see which of the nnUNet model to see which is better than among the variants.

We used the default hyper parameters for nnUNet. We tried four different nnUNet variants. First one is the vanilla nnUNet. In second one, the encoder is Resnet. And last two are two variants with different augmentations.

Table 2 nnUNet different variants details

Vanilla nnUNet
ResencUNet_DA3_BN (ResNet and Data Augmentation)
nnUNet_DA3 (Data Augmentation)
nnUNet_DA5 (Data Augmentation

We found that during creating pseudo-labels, the models showed that most of the labels were empty (black background). The models learned the black background more easily then VS and Cochlea. This gave us the indication that models are overfitting and we should go with variants which are using a lot of Data Augmentation. In our case, nnUNet with Data Augmentation like Rotation, Blur, Contrast, Mirroring, Brightness, etc. We found that nnUNet_DA5 generalized better than the other variants and we selected DA5 as our chosen model.

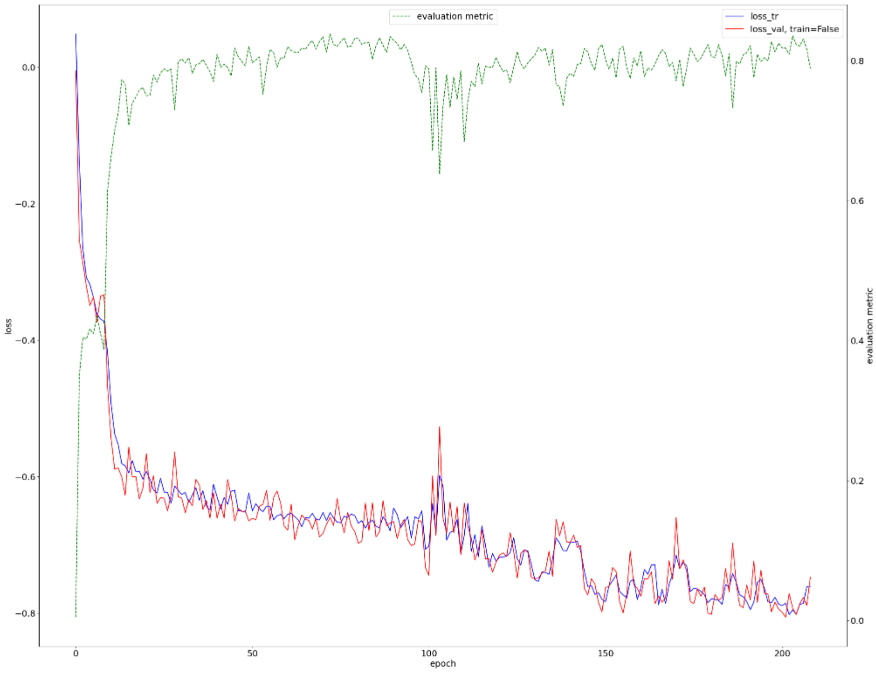

Figure 1 Vanilla nnUNet

Vanilla nnUNet has a good foreground dice score of around 0.8, while the training and validation losses reaches nearly to -0.7 after 200 epochs. ResencUNet_DA3_BN on the other hand, performed similar to Vanilla nnUNet. nnUNet_DA3 and DA5 has squigglier graph but generalize quite better. Finally, we took DA5 as our selected model and trained it on pseudo labels from training the model on hrT2 images.

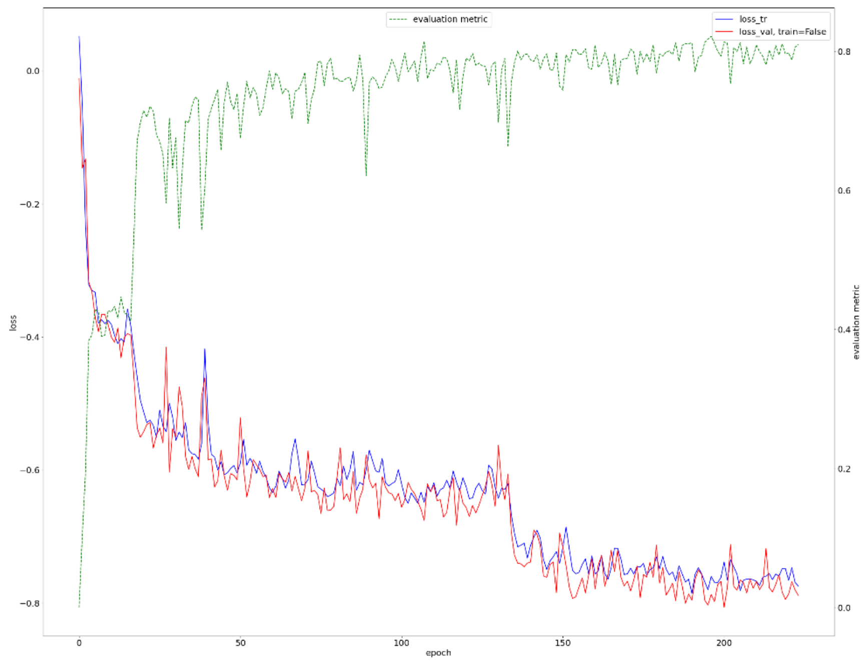

Figure 2 ResencUNet_DA3_BN

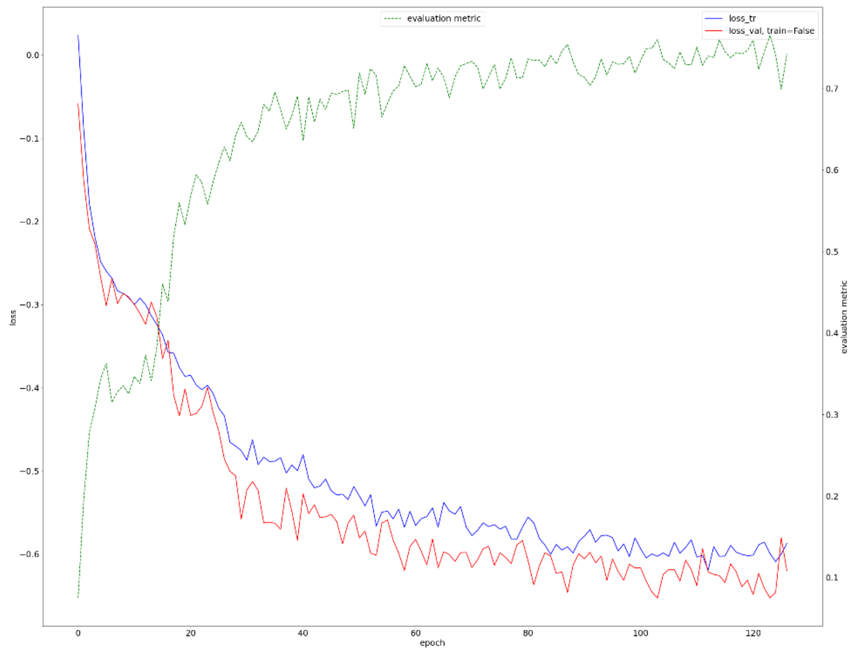

Figure 3 nnUNet_DA3

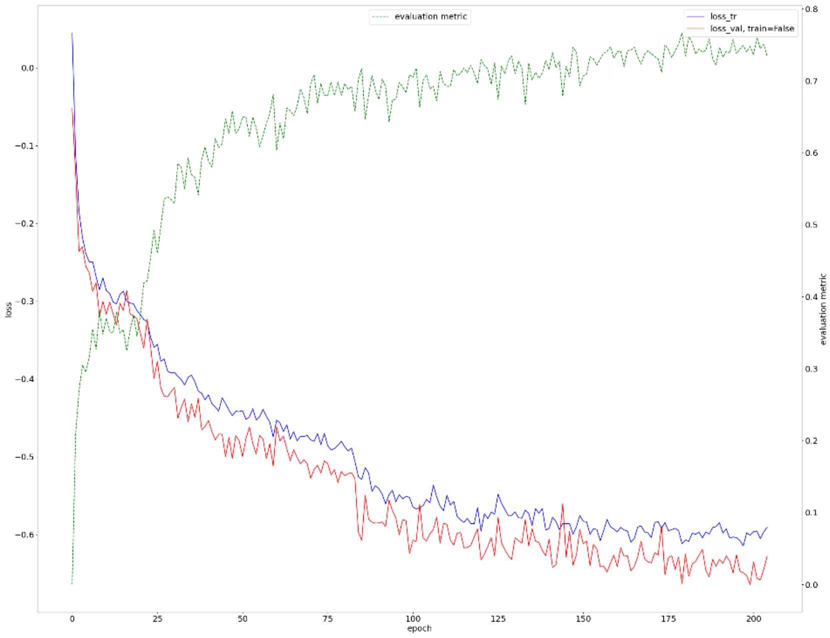

Figure 4 nnUNet_DA5

2.2 Deployment

We used Docker to containerize the code. The shipped code is just an inference part, to keep the docker image as light as possible.

docker run --rm --gpus all -v [input directory]:/input/:ro -v [output directory]:/output -it [image name]

The code requires GPU which can be utilized into the container through --gpus all parameter. Image name is uzairnoman/crossmoda:latest. Complete is available on github on the following link.

References

1. UVCGAN repository : https://github.com/LS4GAN/uvcgan. [Accessed 12 07 2022]
2. D. Torbunov, Y. Huang, H. Yu, J. Huang, S. Yoo, M. Lin, B. Viren and Y. Ren, "UVCGAN: UNet Vision Transformer cycle-consistent GAN for unpaired image-to-image translation," *eprint arXiv:2203.02557*, 2022.
3. nnUNet repository : https://github.com/MIC-DKFZ/nnUNet. [Accessed 03 08 2022].
4. Y. a. B. Y. Grandvalet, Semi-supervised Learning by Entropy Minimization, MIT Press, 2004.