VALDO challenge submission - task 2: dawai

***blob loss* for microbleeds segmentation benchmark**

Florian Kofler[1,2,3], Suprosanna Shit[1,3], Ivan Ezhov[1,3], Izabela Horvath[4], Rami Al-Maskari[1,3,4], Jan Kirschke[2], Claus Zimmer[2], Benedikt Wiestler[2⋆], and Bjoern H. Menze[1,5⋆]

[1] Department of Informatics, Technical University Munich, Germany
[2] Department of Diagnostic and Interventional Neuroradiology, School of Medicine, Klinikum rechts der Isar, Technical University of Munich, Germany
[3] TranslaTUM - Central Institute for Translational Cancer Research, Technical University of Munich, Germany
[4] HMGU - Helmholtz Institute Munich, Germany
[5] Department of Quantitative Biomedicine, University of Zurich, Switzerland
Corresponding author: `florian.kofler@tum.de`

Abstract. We present our contribution to the VALDO challenge Task 2: Segmentation of cerebral microbleeds. We employ a standard U-Net training with a *dice+bce* extended to a *blob loss*.

Keywords: segmentation, MR, cerebral microbleeds, VALDO

1 Network training

1.1 CNN

We use a basic 3D U-Net implemented in MONAI [6] inspired by [1]. We train with all three input modalities (T1, T2, T2S) with a dropout of *0.1* and employ *mish* as activation function [3], otherwise we stick to the default parameters of the U-Net implementation.

1.2 Training data

We use exclusively the labeled microbleed data provided by challenge organizers for training.

1.3 Data preprocessing and augmentation

First, we assert that there are no *infinite*, *-infinite* or *nan* values in the input data or labels. Afterwards we apply a quantile based normalization and augment the training data with flips, a tiny bit of Gaussian noise and random affine transformations. Due to hardware problems, we stick to this rudimental augmentation protocol to maintain training times on acceptable levels.

⋆ contributed equally as senior authors
[6] https://docs.monai.io/en/latest/networks.html#monai.networks.nets.BasicUNet

1.4 Training procedure and model selection

We train our U-Net with *batch size 4* on two crops of size *192x192x32* per batch element, which leads to *44/48GB VRAM* consumption on a *NVDIA Quadro RTX 8000.* The crops are randomly sampled and have a 95 percent probability to be based on a foreground center voxel. We do not use a conventional train- / validation split and train with an initial learning rate of *1e-3* on all available training data for 222 epochs [7] select the model checkpoint from the last epoch. *Ranger21* serves as optimizer [5]. Further, we extend an equally weighted *dice + binary cross entropy* loss to a *blob loss* using $\alpha = 0.66$ and $\beta = 0.33$ for model training [2].

1.5 Adjustments from validation phase

Our original network was based on 400 epochs of training with Ranger [4] as the optimizer. As we observed all exams with either 0 or nan results on the validation set, we completely retrained for the test submission. Therefore, we reduced the training epochs to 222, switched the optimizer to Ranger21 [5] and switched the activation function from *leaky RelU* to *mish* [3] to attempt better generalization. In addition, we introduced the *nan/infinity* corrections and increased overlap for the sliding window inference from *0.5* to *0.7*. As we noticed difficulties when training from scratch, we warm-started the training with model weights from our submission for the lacune segmentation task.

1.6 Final inference and test time augmentations

We use a sliding window inference with a batch size of 7 and a patch size of *256x256x64* and an overlap of 0.7. Additionally, we employ flips and slight Gaussian noise as test-time augmentations. Again, we normalize input data and assert that there are no nan or infinite values. To generate the uncertainty map, we compute the absolute inverse of the model outputs.

Acknowledgement

Bjoern Menze, Benedikt Wiestler and Florian Kofler are supported through the SFB 824, subproject B12. Supported by Deutsche Forschungsgemeinschaft (DFG) through TUM International Graduate School of Science and Engineering (IGSSE), GSC 81. Suprosanna Shit and Ivan Ezhov are supported by the Translational Brain Imaging Training Network(TRABIT) under the European Union's 'Horizon 2020' research & innovation program (Grant agreement ID: 765148). With the support of the Technical University of Munich – Institute for Advanced Study, funded by the German Excellence Initiative. Suprosanna Shit are supported by the Graduate School of Bioengineering, Technical University of Munich. Research reported in this publication was partly supported by AIME GPU cloud services.

[7] Again, this is not a deliberate decision but dictated by hardware constraints.

References

1. Falk, T., Mai, D., Bensch, R., Çiçek, Ö., Abdulkadir, A., Marrakchi, Y., Böhm, A., Deubner, J., Jäckel, Z., Seiwald, K., et al.: U-net: deep learning for cell counting, detection, and morphometry. Nature methods **16**(1), 67–70 (2019)
2. Kofler, F., Shit, S., Ezhov, I., Fidon, L., Al-Maskari, R., Li, H., Bhatia, H., Loehr, T., Piraud, M., Erturk, A., et al.: blob loss: instance imbalance aware loss functions for semantic segmentation. arXiv preprint arXiv:2205.08209 (2022)
3. Misra, D.: Mish: A self regularized non-monotonic neural activation function. arXiv preprint arXiv:1908.08681 (2019)
4. Wright, L.: Ranger - a synergistic optimizer. https://github.com/lessw2020/Ranger-Deep-Learning-Optimizer (2019)
5. Wright, L., Demeure, N.: Ranger21: a synergistic deep learning optimizer. arXiv preprint arXiv:2106.13731 (2021)