NTNU
Kunnskap for en bedre verden

Department of Computer Science

TDT4265 - Computer Vision and Deep Learning

Project Report (VÅR 2024)

Group 187:
Alexander Huhn
Felix Zijie Rong

28th April 2024

Contents

Chapter 1

Technical Details of the Models

1.1 Hyperparameters

The training of the models is initialized using a *.yaml* config file, which can be found in the *model/configs* folder, e.g. *TDT4265_StarterCode_2024/final_final/dints/configs/hyper_parameters.yaml*. All relevant hyperparameters (learning rate, optimizer, early stopping, batch size, epochs, continue training, ...) can be found there. The config files also include the preprocessing transforms with respective parameters.

1.2 Model architecture

The implementation of the model and preprocessing transforms has been taken from the MONAI framework Cardoso et al., 2022; Consortium, 2020a. Details about the model A (segresnet Myronenko, 2018) and model B (dints He et al., 2021) can be found in the MONAI documentation Consortium, 2020b.

An overview of the model architecture is given for both models A and B in Fig. 1.1 and Fig. 1.2, respectively.

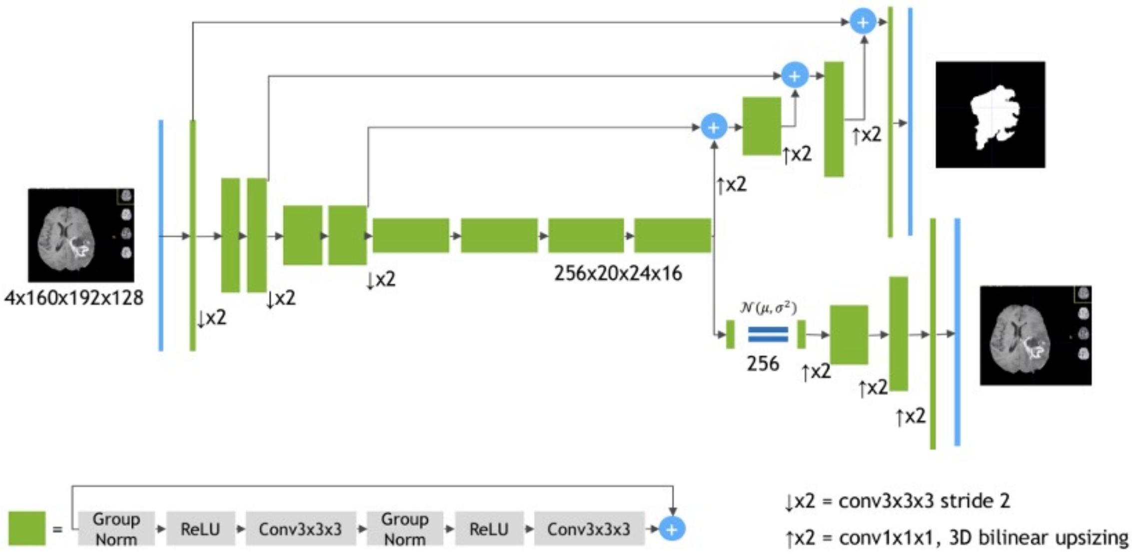

Figure 1.1: Model A - Segresnet Myronenko, 2018.

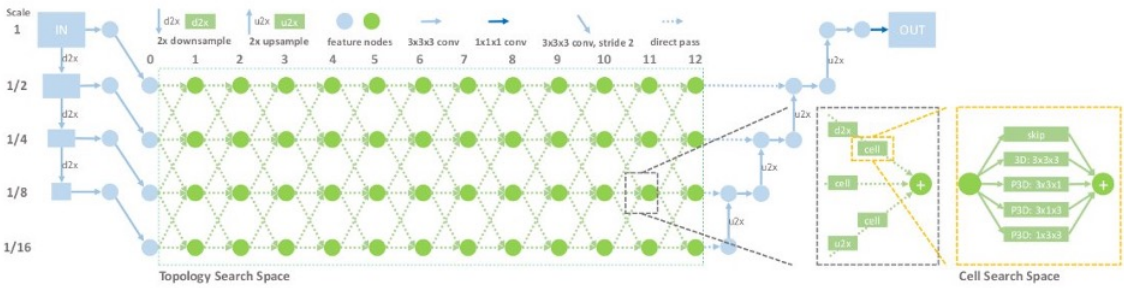

Figure 1.2: Model B - Dints He et al., 2021.

Chapter 2

Model usage (how to train the model)

2.1 Conda environment

To train the model, a conda environment needs to be installed. All required python libraries are listet in the conda environment file *environment.yaml* and it includes the detailed instructions (shell commands) to install, update or remove the environment.

2.2 Model training

Each model is trained by running the python command for training, written at *docs/README.md* in the respective model folder. The training has been conducted on the IDUN cluster using the following srun command (interactive slurm session) or an equivalent slurm submit script (*run.slurm*), for example in case of Model A:

```
$ srun --partition=GPUQ --account=ie-idi --time=12:00:00 --nodes=1
--ntasks-per-node=1 --gres=gpu:1 --cpus-per-task=2 --mem=64G --pty bash

$ cd final_final/segresnet
$ timeout 43200 CUDA_VISIBLE_DEVICES=0 python scripts/train.py run
--config_file=configs/hyper_parameters.yaml
```

2.2.1 Pre-Training

For pre-training the model on the imageCAS dataset, the models are trained first on imageCAS for 12h and then continue training will be activated (*finetuning=true* in [model_name]/configs/hyper_parameter.yaml) and the model will be trained on the ASOCA dataset. The detailed settings and scripts for this can be found in the *with_pretrain* folder. The subfolders *with_pretrain/ImageCAS* and *with_pretrain/ASOCA* means that the model is first pretrained on imageCAS and afterwards the pretrained model is further trained on ASOCA respectively.

2.3 Inference, postprocessing and visualization

In order to run the inference, following script is used:

```
$ time python scripts/infer.py run --config_file=configs/hyper_parameters.yaml
```

The following notebooks are used for further postprocessing, evaluation and visualization of the results: *evaluate_mean_dice_and_hd95.ipynb* and *3d_visualize.ipynb* and *train_loss_plot.ipynb*.
NB! The plots in *train_loss_plot.ipynb* require a latex environment and *3d_visualize.ipynb* works most likely only local since pyvista 3d plots will most likely an error on the idun cluster.

2.4 Data format

Regarding the data format, the data paths are given in the respective *model1_auto3dseg/data.json* or *data_ImageCAS.json*. All data samples should be in a data_flatten (or data_flatten_ImageCAS) folder with unique naming. For the imageCAS dataset, the script *preproc_imCAS_data.py* can be used to automatically create those files.

2.5 Data exploration

Can be found in the notebooks *data_exploration.ipynb* and *explore_transforms.ipynb*.

Bibliography

Cardoso, M. Jorge et al. (2022). *MONAI: An open-source framework for deep learning in healthcare*. arXiv: 2211.02701 [cs.LG].

Consortium, The MONAI (2020a). *Project MONAI*. URL: http://doi.org/10.5281/zenodo.4323059.

— (2020b). *Project MONAI*. URL: https://github.com/Project-MONAI/tutorials/blob/main/auto3dseg/docs/algorithm_generation.md.

He, Yufan et al. (June 2021). 'DiNTS: Differentiable Neural Network Topology Search for 3D Medical Image Segmentation'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5841–5850.

Myronenko, Andriy (2018). *3D MRI brain tumor segmentation using autoencoder regularization*. arXiv: 1810.11654 [cs.CV].

List of Figures