Encoder-decoder-style CNN-based brain segmentation, applied to MRBrains 2018

Mohsin Shaikh[1], Alexander Schlaefer[2], René Werner[1,3]

[1] Department of Computational Neuroscience, University Medical Center Hamburg-Eppendorf, Germany
[2] Institute of Medical Technology, Hamburg University of Technology, Germany
[3] DAISYlabs, Forschungszentrum Medizintechnik Hamburg (FMTHH), Germany

1. Architecture

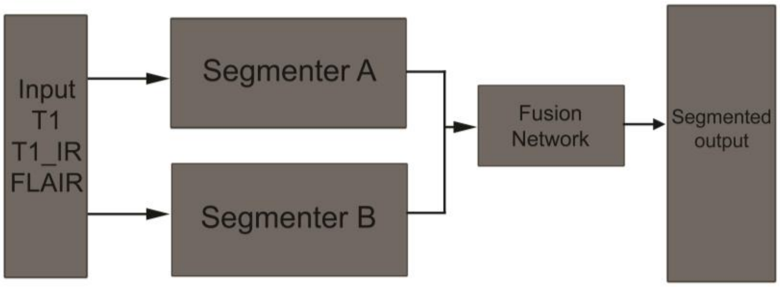

Fig 1: Overview of applied architecture

The applied architecture is built using three components: two initial pixel-wise segmentation networks, subsequently referred to as *Segmenter A* and *Segmenter B*, and a fusion network, which produces the final pixel-wise segmented output based on the inputs received from the initial segmentation networks. Segmenter A is based on the Segnet [1] architecture, whereas Segmenter B and the Fusion Network are based on the U-Net [2] architecture. Architecture details are given in respective succeeding sections. All networks were trained only on the dataset provided by the MRBrainS18 Challenge. Segmenter A and B networks utilize three modalities, namely T1, T1 Inversion Recovery and FLAIR, as input; sought pixel output labels were Grey matter, White matter and Cerebrospinal Fluid, Background and a merged label for Brain Stem and Cerebellum. No preprocessing was applied to the input data. Weight initialization of the networks was random. All networks were trained on weighted Cross Entropy Loss function. Parameters in the networks were updated based on the Adam Optimization algorithm. A constant batch size of 16 was used for all networks.

1.1 Architecture of *Segmenter A*

Segmenter A takes slices of 3 channel (T1, T1_IR and FLAIR) of 240 x 240 pixels as input and outputs a 5 channel 240 x 240 pixel-wise segmented output, of which each channel corresponds to a segmentation label. The network was trained for 120 epochs. As already suggested in [1], this type of network can be divided into two sections: encoder and decoder. The encoder part is made up of 4 convolution blocks; each block consists of 3 convolution layers, 3 batch normalization layers and a ReLU non linearity is applied after each batch normalization layer. Finally, the output of each convolution block is downsampled using Max Pooling layers. Pooling indices are stored for upsampling in the decoder part. All convolution layers use 3 x 3 convolution kernels with stride and padding of 1. The decoder is also made up of 4 convolution blocks, with each block containing the same number of convolution and batch normalization layers to the encoder. The main difference between the encoder and decoder network is its Max Unpooling layers, which takes output from convolution block and pooling indices from the corresponding encoder section as input, producing an upsampled version of the input. The network was trained with a learning rate of 0.01

1.2 Architecture of *Segmenter B*

Similar to Segmenter A, Segmenter B also takes slices of 3 channels (T1, T1_IR and FLAIR) and 240 x 240 pixel as input and outputs a 5 channel 240 x 240 pixel-wise segmented output; each channel corresponds to a segmentation label. This network was also trained for 120 epochs. Segmenter B is based on [2]; hence, it has two arms, a downsampling encoder and an upsampling decoder. Each convolution block in the network has 3 convolution layers with kernel size of 3 x 3 and stride and padding of 1, and 3 batch normalization layers, followed by ReLU activation. Encoder section has 3 such blocks, each of which is followed by a Max Pooling layer, whereas the decoder has 4 of it, followed by a transposed convolution layer. A channel-wise concatenation takes place between the corresponding upsampled output of convolution blocks of the encoder and decoder section. The applied learning rate was 0.01.

1.3 Fusion Network

The Fusion Network is also based on [2]. It can be regarded as a smaller version of Segmenter B with lesser number of parameters to learn. This network is only concerned with proper fusion of the respective inputs from the parent networks, preventing overfitting. This network converges after being trained for 80 epochs. The network has 2 convolution blocks in downsampling encoder and upsampling decoder section, each block containing 2 convolution layers, kernel size 3 x 3 with stride and padding of 1, and 2 batch normalization layers, followed by ReLU activation. Each convolution block in the decoder section is preceded by a bilinear upsampling layer. The fusion network takes two 5 channel 240 x 240 pixel-wise concatenated outputs produced by the Segmenter networks as input and produces 5 channel pixel-wise segmented final output.

References

[1] Segnet : IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume: 39, Issue: 12, Dec. 1 2017)

[2] Unet: Ronneberger O., Fischer P., Brox T. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N., Hornegger J., Wells W., Frangi A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham