

SUPPLEMENTARY MATERIALS

Deep Learning-Based Segmentation Method

Deep learning-based Brain MRI segmentation method

We used Split-Attention U-Net (SAU-Net), a convolutional neural network with skip pathways and a split-attention module that segments brain MRI scans. The entire image processing pipeline was implemented using Python 3.7 with TensorFlow version 1.5. Previously, we demonstrated that SAU-Net achieved better segmentation accuracy with better reliability that surpasses those of state-of-the-art methods.¹ We believe that SAU-Net has excellent potential due to its robustness to neuroanatomical variability that would enable almost instantaneous access to accurate neuroimaging biomarkers and its swift processing runtime compared to other methods investigated.

Preprocessing

We applied an identical pipeline from the pre-processing pipeline for all the data utilized in this study. Our preprocessing includes resampling, zero-padding, and intensity normalization using histogram matching. We first resampled the image in an isotropic voxel (1 mm³), then padded it with zero using a filter size of 16×16×16 and 24×24×24 for training and testing, respectively. Finally, we normalized the MRI intensity by applying a histogram matching algorithm as described elsewhere.² The entire preprocessing pipeline was implemented using the NiftiNet library.³

Deep learning segmentation

Our in-house segmentation tool was developed from the existing UNet++ deep learning architecture with a three-dimensional methodology to train 97 labels. Our deep learning design has a convolutional layer in the skip path, which bridges the semantic gap between the encoder and decoder characteristic maps. A dense skip connection in the skip path, which improves the gradient flow, has deep supervision, which enables model pruning, improves performance, or, at worst, compares using only one lossy layer (Supplementary Figure 1). The cross-entropy loss function was used for the voxel-by-voxel segmentation learning and the learning rate for the Adam optimizer was 0.0001.

For the training dataset, we performed Desikan–Killiany atlas-based FreeSurfer segmentation on 388 patients from public datasets including HCP, ADNI, PPMI, AIBL, and IXI, and two experts performed manual correction to produce a fine-tuned gold standard. Among the entire data set, we first randomly shuffled and set aside 49 datasets for testing. The remaining data were then categorized for training and validation (9.5:0.5). The training data were constructed by extracting the three-dimensional patch image using uniform sampling (96×96×96) for the individual ground truth data (Supplementary Figure 2). With the aforementioned training parameters, the model was iteratively trained 500,000 times. The batch size was set to 1, which was the limit that could be handled by the 11 GB RAM of one RTX 2080Ti GPU. For comparison, FreeSurfer software (version 7.0.0, <https://surfer.nmr.mgh.harvard.edu>) was used to identify sub-regional brain volumes.

Postprocessing

To increase the validity of the structures, we further applied connected-component labeling. No other post-processing was applied and volumes with regard to the original MRI were computed for the analysis. Segmentation results are obtained by merging inference data using a three-dimension patch sliding aggregator. Supplementary Figure 3 shows the segmentation result of brain sub-volumes.

Trial registration

We used MR images from the Carnus (cardiovascular risk of Korean nun study) sub-dataset. Clinical trial was registered with the Clinical Trials Registry at <https://cris.nih.go.kr> (KCT0006054).

REFERENCES

1. Lee M, Kim J, Ey Kim R, Kim HG, Oh SW, Lee MK, et al. Split-attention U-Net: a fully convolutional network for robust multi-label segmentation from brain MRI. *Brain Sci* 2020;10:974.
2. Nyúl LG, Udupa JK, Zhang X. New variants of a method of MRI scale standardization. *IEEE Trans Med Imaging* 2000;19:143-150.
3. Gibson E, Li W, Sudre C, Fidon L, Shakeri DI, Wang G, et al. NiftyNet: a deep-learning platform for medical imaging. *Comput Methods Programs Biomed* 2018;158:113-122.