A Multi-stage Framework for Cerebral Microbleeds Detection and Segmentation

Zihao Chen[1] and Qi Dou[2]

[1] School of Biomedical Engineering, Shanghai Jiao Tong University, Shanghai, China
harrychenzh@sjtu.edu.cn
[2] Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, China

Abstract. We propose a multi-stage cerebral microbleeds detection and segmentation framework for MICCAI 2021 challge VAscular Lesions DetectiOn(Where is VALDO) task2. This technical report describes the framework we proposed and the details including preprocessing applied on images, network architectures, training strategies, inference procedure and initial performance we obtained on training set by 5-fold cross validation.

Keywords: Cerebral microbleeds · Detection · Segmentation · VALDO.

1 Methodology

Inspired by the automatic Cerebral Microbleeds(CMB) detection method[1], we proposed a multi-stage framework for CMB detection and segmentation, which consists of three stages: screening, discrimination and segmentation. In screening stage, a 3D Fully Convolutional Network(FCN) takes the whole volumetric data and outputs a score volume which represents the probability of CMB at corresponding voxels. After postprocessing, the CMB candidates are obtained by mapping the score volume back to the input volume space. Then a 3D CNN model is used in discrimination stage to identify the true CMBs from the candidates and reduce the false positives. Finally, a U-Net is used for segmentation.

1.1 Preprocessing

Referring to the preprocessing steps of nnU-Net[2], intensity normalization, resampling and cropping are applied in images. Firstly, the gray value of background are set to 0 and the foreground are normalized by z-score normalization. Next, all images are resampled to the median voxel spacing of the dataset, which is [0.49, 0.49, 0.80] in this dataset. Third-order bspline and nearest-neighbour interpolation are applied in MRI scan, linear interpolation is used in segmentation mask. Finally, the images are cropped to a region of non-zero values. Cropping is optional for validation data because the proposed method could deal with arbitrary-size input.

1.2 Network architectures

Screening. Target of screening stage is to detect CMBs fastly and achieve high sensitivity. Compared to traditional CNN, FCN converts the final fully connected layer to convolution layer. As a result, FCN can take volume of arbitrary size and output the score volume with a single forward propagation, which greatly accelerates the inference speed but still retains good performance. When training the FCN, we set the block size to [20, 20, 16] because 98.4% of CMBs can be covered by a bounding box of this size. The network architecture is dipicted as Table. 1. In which C represents a convolution layer, C(BN) is the combination of a convolution layer and a batch normalization layer, and M represents a maxpooling layer. Leaky ReLU with 0.01 negative slope is used for non-linear activation function.

Table 1. Architecture of screening network

Layer	Kernel size	Stride	Output size	Feature maps
Input	-	-	20×20×16	1
C1(BN)	5×5×3	1	16×16×14	64
C2(BN)	5×5×3	1	12×12×12	64
M1	2×2×2	2	6×6×6	64
C3(BN)	3×3×3	1	4×4×4	128
C4(BN)	3×3×3	1	2×2×2	128
C5	2×2×2	1	1×1×1	256
C6	1×1×1	1	1×1×1	2

Discrimination. The discrimination stage aims to identify true CMB from the candidates. By enlarging the block size, the network could learn richer contextual information and better distinguish true CMBs from other mimics. But considering the CMBs are small, too large neighbourhood may introduce redundant information and weak the performance. So we set the block size of discrimination stage [24, 24, 20]. The detailed architecture of discrimination network is depicted as Table. 2. C, BN, M and activation function are the same as in screening part.

Segmentation. In segmentation stage, a U-Net with input size of [32, 32, 24] is used to obtain voxel-wise probability of CMBs. The network has two blocks per resolution stage both in downsampling and upsampling, while each block consists of a convolution layer, batch normalization layer and a leaky ReLU layer. Downsampling is implemented by strided convolution and upsampling is implemented by transposed convolution. The number of feature maps at beginning is 32, which doubles in each downsampling and halves in each upsampling.

Table 2. Architecture of discrimination network

Layer	Kernel size	Stride	Output size	Feature maps
Input	-	-	24×24×20	1
C1(BN)	5×5×3	1	20×20×18	64
C2(BN)	5×5×3	1	16×16×16	64
M1	2×2×2	2	8×8×8	64
C3(BN)	3×3×3	1	6×6×6	128
C4(BN)	3×3×3	1	4×4×4	128
FC1	-	-	1×1×1	512
FC2	-	-	1×1×1	2

1.3 Training procedure

Obtaining training samples. The training samples are made up of positives, negatives and additional false positives. Positives are extracted from connected components of segmentation mask and augmented by translation(<4 voxels), rotation(<10°) and flip(lateral). Negatives are randomly selected from the training MRI scans. And the false positives are acquired by applying the initial screening model on training scans. Finally, about 0.4 million training samples are generated, and the proportion of positives, negatives and supplemental false positives is around 1:2:1.

Training protocol. For screening and discrimination stage, the loss function is weighted cross entropy loss, and the weights for background and CMB are 0.25 and 0.75. Stochastic gradient descent with Nesterov momentum (μ=0.99) and an initial learning rate of 1e-2 is used for learning network weights. Dice loss and Adam optimizer are used for segmentation stage. The learning rate decays according to $(1 - \frac{epoch}{epoch_{max}})^{0.9}$. In each iteration, training samples are resampled to ensure the number and the proportion of samples are fixed under different conditions. In discrimination and segmentation stage, only the positives and false positives are used for training. The number of iterations in screening, discrimination and segmentation are 150, 80 and 100, and the numbers of samples per iteration are 0.2 million, 0.1 million and 8k respectively.

1.4 Inference

Firstly, the screening model inputs a whole volume and outputs a 3D score volume. After thresholding and non-maximum suppression, the score volume is mapped back to the original input space and then the CMB candidates are obtained. In order to achieve high sensitivity, the threshold is set to 0.1. Next, for each candidate, discrimination model takes its neighbourhood and predicts whether it is true CMB. During discrimination stage, the center of candidates is slightly translated in order to prevent missing the true CMB. The threshold is set to 0.5. Finally, the segmentation model outputs voxel-wise probability map

of all candidates extracted by above two stages. Then the prediction are mapped back to input space and the results of overlap regions are averaging, the final probability map is obtained.

2 Experiment and results

The training dataset includes 72 MRI scans acquired from three sources, each scan has three modality, T1, T2 and T2*. It was observed that CMBs are not obvious in T1 and T2 images, so only T2* images were used in our framework. To evaluate the performance of proposed framework, 5-fold cross validation was adopted:

Table 3. Quantitative results of 5-fold cross validation

Fold	Dice	Volume difference	F1	Count difference
0	0.65	47.6	0.73	1.0
1	0.54	19.2	0.79	1.6
2	0.47	36.9	0.59	2.6
3	0.47	209.5	0.61	6.4
4	0.52	47.5	0.66	1.3
Average	0.53	72.1	0.68	2.6

As shown in Table. 3, our proposed framework achieved 0.53 average Dice and 0.68 average F1. It is found that in some cases the CMB is only a few voxels in size, which means the quantitative performance could be extremely bad if the predictions of model were slightly off. Therefore, the detection algorithm plays a important role in CMB segmentation task.

References

1. Dou, Q., Chen, H., Yu, L., Zhao, L., Qin, J., Wang, D., Mok, V.C., Shi, L., Heng, P.A.: Automatic detection of cerebral microbleeds from mr images via 3d convolutional neural networks. IEEE transactions on medical imaging **35**(5), 1182–1195 (2016)
2. Isensee, F., Jaeger, P.F., Kohl, S.A., Petersen, J., Maier-Hein, K.H.: nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. Nature methods **18**(2), 203–211 (2021)