Supplemental Materials

Tissue-at-risk and Ischemic Core Estimation using Deep Learning in Acute Stroke

Supplemental Methods
Supplemental Tables
Supplemental Figures

Supplemental Methods

Neural Network Details

An attention-gated U-Net architecture was implemented in this study (Figure I)[1, 2]. We combined the traditional U-Net architecture with attention gates[3] to focus on target areas in the images. This was achieved by combining contextual information from output of previous layers (coarser scales) and symmetric encoding layers. A "2.5D" model was used in this study to balance the computational efficiency and 3D image information. "2.5D" means that five consecutive slices of DWI, ADC, Tmax, MTT, CBF, CBV, and masks of Tmax and ADC were used to predict the probability of infarct on the center slice. The ground truth was a binary mask of final infarct lesion of the middle slice. Image mirroring around the midline was used to augment the training data size. The output of the model is a probability map with voxel values that ranged from 0 to 1. A value close to 1 indicates the voxel is more likely to be infarcted, while a value close to 0 indicates the voxel is likely to be spared.

The model was trained with the ADAM optimizer (learning rate 0.0005) using a batch size of 16 and 120 epochs. 50% dropout was implemented to prevent overfitting in both encoding and decoding layers[4]. We used a mixed loss function of weighted binary cross-entropy, mean absolute error (L1 loss), Dice score coefficient (DSC), and volume loss as described below:

$$\text{Weighted binary cross entropy} = -\frac{1}{N}\sum_{i=0}^{N} R_1 y_i \log(p_i) + R_0(1 - y_i)\log(1 - p_i)$$

p is the predicted probability. y is the ground truth value of that voxel (0 = not infarcted, 1 = infarcted). N is the total number of voxels. Since stroke lesions are only present in relatively a small fraction of all brain voxels, weighting was applied to balance the numbers of positive and negative voxels. The weights were calculated based on the ratio of the positive and negative voxels of each training batch:

$$R_0 = \frac{1}{\frac{N_-}{N_+}+1}$$

$$R_1 = 1 - R_0$$

where N_- and N_+ represent the number of negative and positive voxels per batch, respectively.

$$\text{L1 loss} = \sum_{i=0}^{N}|y_i - p_i|$$

$$\text{Dice score coefficient (DSC)} = \frac{2N_{TP}}{2N_{TP} + N_{FP} + N_{FN}}$$

$$\text{Volume loss}^5 = \frac{|\sum_{i=0}^{N} p_i - \sum_{i=0}^{N} y_i|}{N_+}$$

N_{TP}, N_{FP}, and N_{FN} are the number of true positive, false positive, and false negative voxels,

respectively. The volume loss was added to counter possible volumetric bias introduced by DSC loss[5].

The loss function was then expressed as:

$$\text{Loss} = \text{Weighted binary cross entropy} + \text{L1 loss} + 0.5 \times (1 - \text{DSC}) + 0.5 \times \text{Volume loss}$$

The weight of 0.5 was given to DSC and volume loss to adjust them to a similar scale as the weighted binary cross entropy and L1 loss.

The implementation was based on Keras (version 2.2.2) with Tensorflow (version 1.10.0) backend. All tests were conducted on a workstation equipped with Quadro GV100 and Tesla V100-PCIE graphical processing units (Nvidia, Santa Clara, CA, USA).

Supplemental Tables

|	Minimal reperfusion (n=33)		Major reperfusion (n=67)	
Table I. Prediction performance of pretraining, separate and thresholding approach.				
	Volume difference, %	Absolute volume difference, %	Volume difference, %	Absolute volume difference, %
Pre-training approach	16 (-23-105)	42 (21-105)	0 (-44-81)	51 (18-89)
Separate approach	27 (-32-151)	51 (30-151)	2 (-35-125)	57 (18-125)
p-value	.4	.3	.007	.02
Thresholding approach	9 (-32-92)	38 (22-92)	-33 (-68-36)	60 (33-103)
p-value	.8	.2	<.001	.3

Data are expressed as median (interquartile range). Percentage are calculated by volume difference / true volume.

Paired comparisons were made between pretraining and separate approach, and pretraining and thresholding approach.

In minimal reperfusion patients, the thresholding approach used the union of Tmax > 6 sec and ADC < 620 x 10^{-6} mm^2/s; in major reperfusion patients, the thresholding approach used the ADC < 620 x 10^{-6} mm^2/s.

Supplemental Figures and Figure Legends

**Figure I.** The block diagram of the attention-gated U-Net model and the schematic of the attention gate.
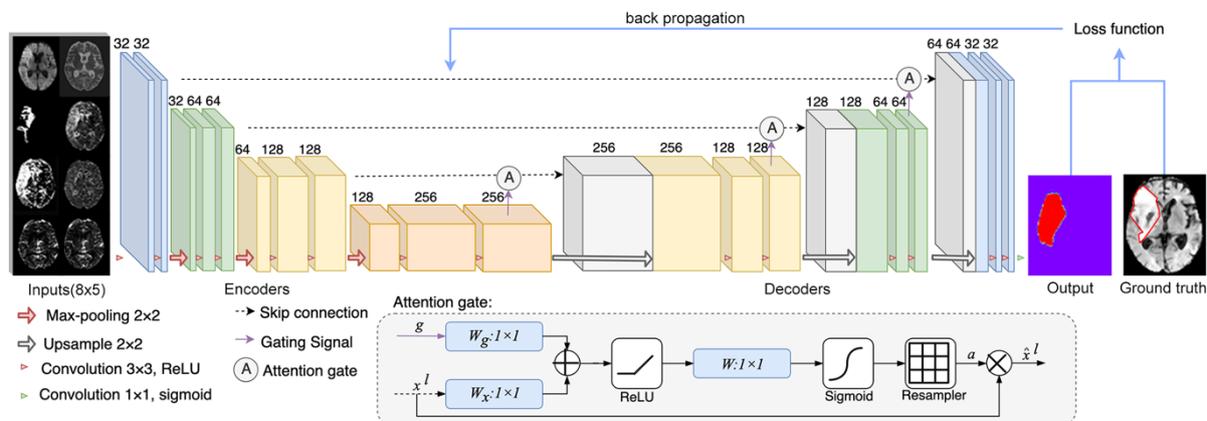
Input images included 5 consecutive slices of diffusion-weighted imaging (b=1000 s/mm^2), apparent diffusion coefficient and its mask with a threshold of less than 620×10^{-6} mm^2/s, Tmax and its mask with a threshold of more than 6 sec, mean transit time, cerebral blood flow, and cerebral blood volume maps. The number of channels is denoted on the top of the box. The skip connections allow high-resolution features to be maintained during training. In an attention gate, the output of previous layer (g) and the symmetric encoding layer (xl) undergo convolution (with 1-by-1 kernel), summation, and ReLU activation. Then another convolution with sigmoid activation is applied to the extract attention coefficient (a), which is then multiplied with the skip connection.

**Figure II.** Attention map output from the attention gate at each level of the U-net.
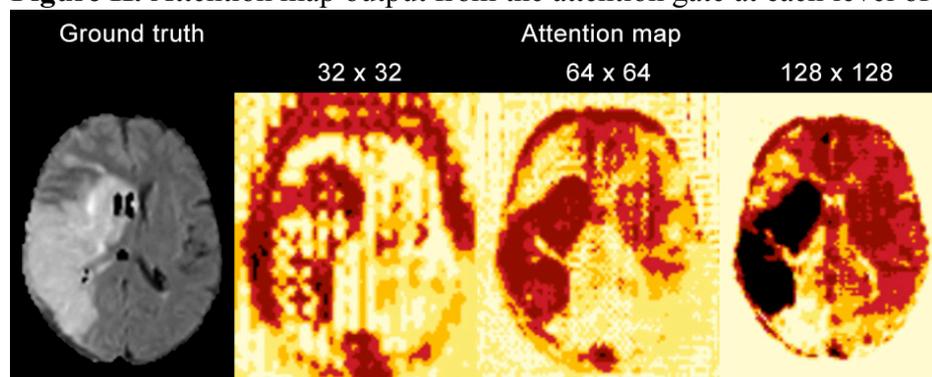
The first attention map locates right after the bottle neck with a size of 32 x 32. The third attention map locates before the model output with a size of 128 x 128.

Figure III. Volume predictions from the pre-training, separate, and thresholding approaches.
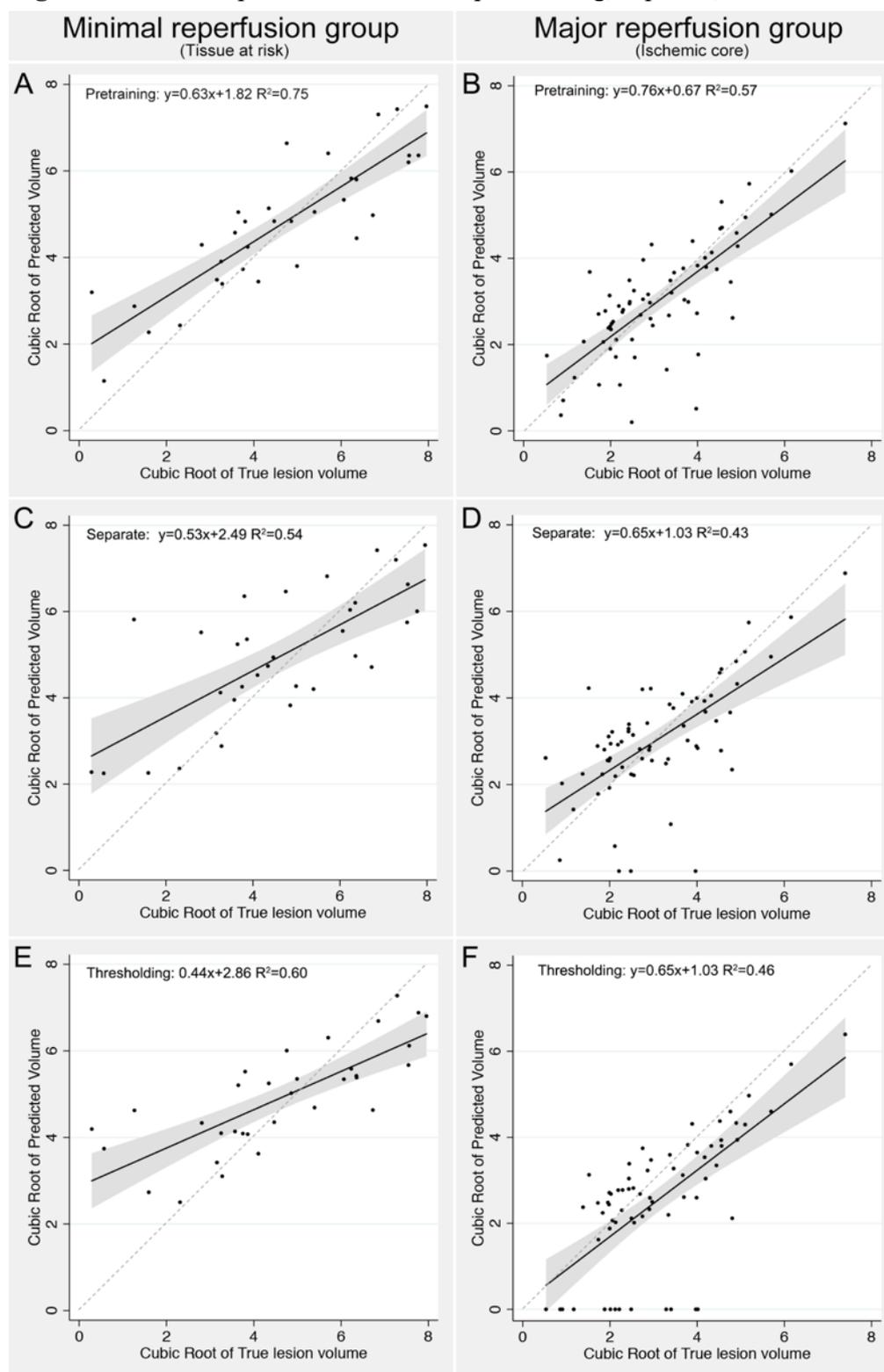

In both minimal (A,C,E) and major (B,D,F) reperfusion patients, the pre-training approach had the best fit and less variation than either the separate or thresholding approach. (F) Note that ADC thresholding erroneously predicted zero lesion size (circled with dashed grey line) more commonly than the deep learning model (13 vs. 0 patients, respectively).

Figure IV. Two representative cases of predictions from the pre-training, separate, and thresholding approaches.
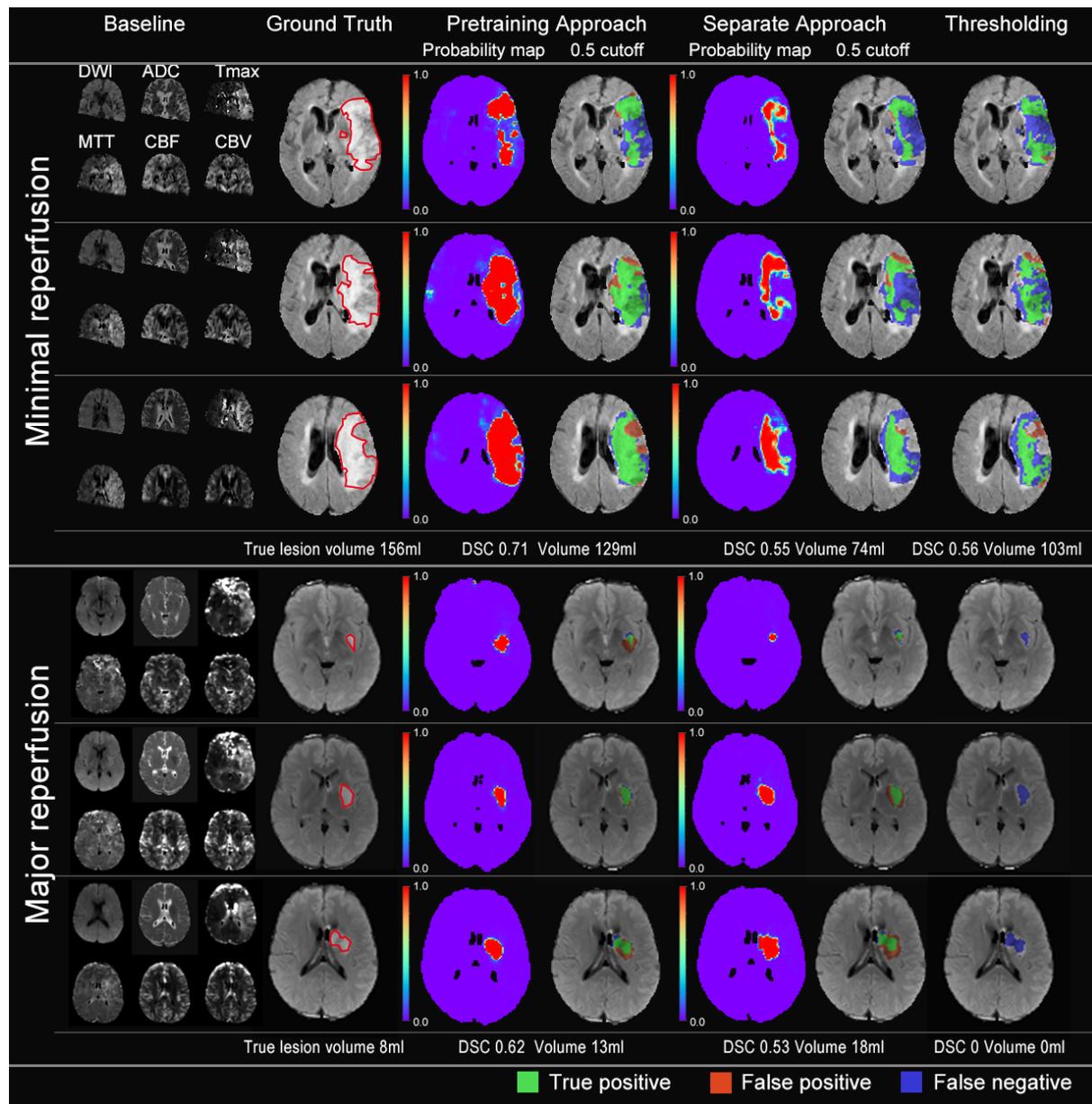
(Top) 71 year-old male treated with IV tPA only, which achieved 0% reperfusion. Of note, only voxels with both diffusion and perfusion information are included in the model. His true lesion is used to define tissue-at-risk. (Bottom) 45 year-old male treated with IV tPA and thrombectomy, which achieved 100% reperfusion. His true lesion is used to define ischemic core. The pre-training approach had more accurate prediction than either the separate approach and thresholding method both visually, with DSC analysis, and volumetrically. Green areas overlayed on the FLAIR image represents true positive, Blue represents false negative, and red represents false positive.

Figure V. Example of how the deep learning models would be used for triage compared with the thresholding method.
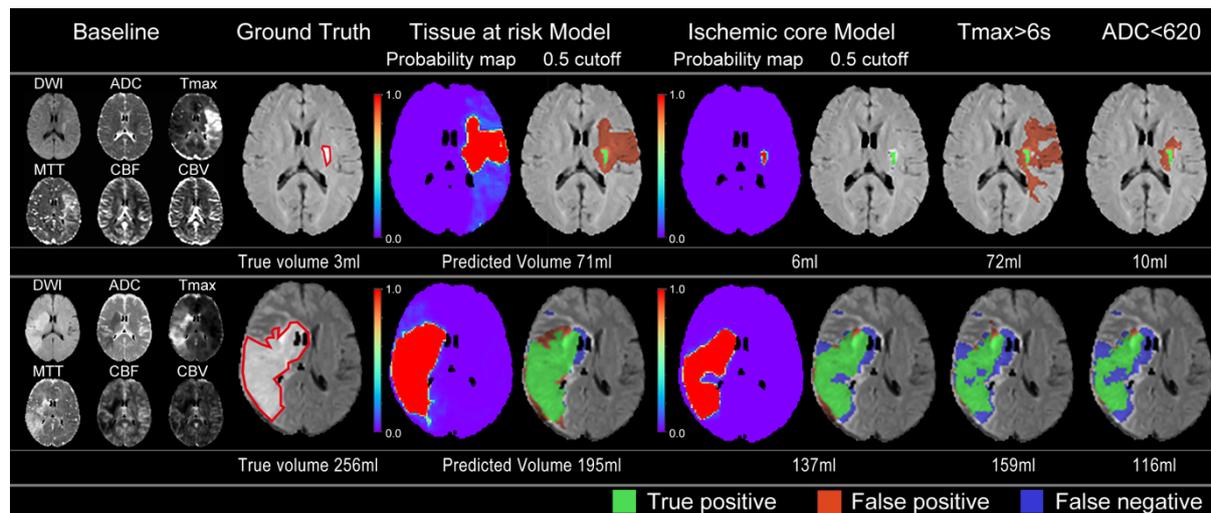
(Top) a 44 year-old female with large mismatch between tissue-at-risk and ischemic core prediction (mismatch ratio of 11.8), indicating a small stroke if the patient received successful treatment or a much larger stroke if the patient did not receive any treatment. She received IV tPA bridging with thrombectomy which achieved 24 hr reperfusion rate of 100%. At 90 days, the patient had full recover with mRS of 0. However, (Bottom) another 44 year-old female with small mismatch ratio of 1.4, indicating limited tissue salvage despite successful treatment. She received the same treatment but only achieved 24 hr reperfusion rate of 10% and had moderately severe disability with mRS of 4 at 3 months.

This illustrates how estimated tissue outcome (with and without reperfusion) can be obtained from the deep learning approach and facilitate clinical decision making on whether to treat the patient. Green areas overlayed on the FLAIR image represents true positive, Blue represents false negative, and red represents false positive.

Supplemental References

1. Pinto A, McKinley R, Alves V, Wiest R, Silva CA, Reyes M. Stroke lesion outcome prediction based on mri imaging combined with clinical information. *Front Neurol*. 2018;9:1060

2. Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. *arXiv e-prints*. 2015

3. Oktay O, Schlemper J, Le Folgoc L, Lee M, Heinrich M, Misawa K, et al. Attention u-net: Learning where to look for the pancreas. *arXiv e-prints*. 2018

4. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A simple way to prevent neural networks from overfitting. *J Mach Learn Res*. 2014;15:1929-1958

5. Bertels J, Robben D, Vandermeulen D, Suetens P. Optimization with soft dice can lead to a volumetric bias. 2019:arXiv:1911.02278