Stroke Lesion Segmentation with 2D Novel CNN Pipeline and Novel Loss Function

Pengbo Liu(✉)

Beijing University of Technology, Beijing, China
pengbo18555@bjut.edu.cn

Abstract. Recently, CT perfusion (CTP) has been used to triage ischemic stroke patients in the early stage, because of its speed, availability, and lack of contraindications. But CTP data alone, even with the generated perfusion maps is not enough to describe the precise location of infarct core or penumbra. Considering the good performance demonstrated on Diffusion Weighted Imaging (DWI), We propose a CTP data analysis technique using Generative Adversarial Networks (GAN) [2] to generate DWI, and segment the regions of ischemic stroke lesion on top of the generated DWI based on convolutional neutral network (CNN) and a novel loss function. Specifically, our CNN structure consists of a generator, a discriminator and a segmentator. The generator synthesizes DWI from CT, which generates a high quality representation for subsequent segmentator. Meanwhile the discriminator competes with the generator to identify whether its input DWI is real or generated. And we propose a novel segmentation loss function that contains a weighted cross-entropy loss and generalized dice loss [1] to balance the positive and negative loss in the training phase. And the weighted cross entropy loss can highlight the area of stroke lesion to enhance the structural contrast. We also introduce other techniques in our network, like GN [13], Maxout [14] in the proposed network. Data augmentation is also used in the training phase. In our experiments, an average dice coefficient of 60.65% is achieved with four-fold cross-validation. From the results of our experiments, this novel network combined with the proposed loss function achieved a better performance in CTP data analysis than traditional methods using CTP perfusion parameters.

Keywords: Generation · Segmentation · Cross-entropy loss · Generalized dice loss

1 Introduction

Stroke is the second leading cause of death worldwide, accounting for 6.24 million deaths globally in 2015 [4]. And more than 80% of stroke cases are ischemic. However, only very few of stroke patients receive recombinant tissue plasminogen

A. Crimi et al. (Eds.): BrainLes 2018, LNCS 11383, pp. 253–262, 2019.
https://doi.org/10.1007/978-3-030-11723-8_25

activator (rtPA) therapy despite proven effectiveness in reducing stroke disability. Defining location and extent of irreversibly damaged brain tissue is a critical part of the decision-making process in acute stroke. Magnetic resonance images (MRI) using diffusion and perfusion imaging can be used to distinguish between infarcted core and the penumbra. Because of the advantages in speed, availability, and contraindications, CT perfusion have been used to triage with acute stroke instead of MRI, which may shorten the scanning time for the stroke patients [3]. Automatic methods, including many commercial software have been developed to measure the perfusion maps of stroke patients, but such methods may not be good enough to solve the heterogeneities of the stroke patients. Therefore, there is a great need for advanced data analysis techniques that could help to diagnosis stroke accurately and precisely with repetitiveness, and support decision-making for treatments. In the literature, more often a optimal threshold of the CTP parameter is used, like $rCBF < 30\%$ within $delaytime > 3\,\mathrm{s}$ is the threshold of the region of core [3]. But this pre-defined threshold suffers from several drawbacks. First, progression of stroke is patient specific and a population threshold may not work in some cases. Second, thresholding means a fixed value for all patients across all scanners in all hospitals, without considerations of site differences.

In the past decade, deep learning technology, especially Convolutional Neural Network (CNN) has achieved huge successes in various computer vision tasks, like classification [7], detection [8] and segmentation [9]. And the power of CNN is demonstrated in medical imaging more and more. Ciresan et al. [10] firstly introduced CNNs to medical image segmentation by predicting a pixel's label based on the raw pixel values in a square window centered on it. But this method is quite slow because the network must run separately for every pixel within every single image and there is a lot of redundancy due to overlapping windows actually. Later on, Ronneberger et al. proposed U-Net [11], which consists of a contracting path to capture context and a symmetric expanding path that enables precise localization and can be trained end-to-end from very few images built upon the famous Fully Convolutional Network (FCN) framework [9]. Specifically, Nielsen [6] introduce the deep CNN in Acute Ischemic Stroke segmentation task with a simple encoder-decoder structure to predict the final infarct.

In this paper, different from Anne Nielsen, we proposed a novel structure of network combined with the GAN approach to solve this task. Our proposed network structure contains a generator, a discriminator and a segmentator. The structure of our work flow is shown in Fig. 1. Generator is used to synthesize the DWI images from the CT data. Discrimintator is used to perform the classification of the generated DWI image and the true DWI image. Segmentator is finally used to segment the lesion of brain on the generated DWI image.

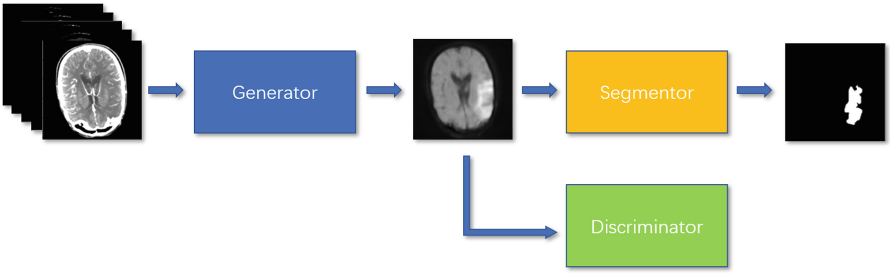

Fig. 1. Structure of our ischemic stroke lesion segmentation work flow. Generator is used to synthesize the DWI images from the CT data. Discrimintator is used to perform the classification of the generated DWI image and the true DWI image. Segmentator is finally used to segment the lesion of brain on the generated DWI image.

2 Method

2.1 Dataset and Data Preprocessing

Our framework is trained and tested using ISLES 2018 Segmentation Challenge dataset, which include imaging data from acute stroke patients in two centers who presented within 8 h of stroke onset and underwent an MRI DWI within 3 h after CTP. The training data set consists of 63 patients, and the testing set includes 40 patients. Because some patients have two slabs to cover the stroke lesion (These are non-, or partially-overlapping brain regions). In the end, we have 94 and 62 cases in training phase and testing phase separately. Each case in training have eight modalities or parametric maps, including CT, CT_4DPWI, CT_MTT, CT_Tmax, CT_CBF, CT_CBV, MR_DWI, and ground truth, OT. All modalities are shown in Fig. 2. ISLES 2018 dataset have the same size, $(256 * 256)$, and the same spacing, $(1 * 1\,\mathrm{mm})$, in x and y dimension, but with a very different slice number in z dimension. Most cases of the dataset have only two slices, which is hard to exploit the information from the 3D data via 3D-CNN method. Thus we separate the 3D images into 2D slices along z axis, which equivalently augment the data size, as more training data are available in a 2D manner with a 2D CNN network. We concatenate the channels of CT_MTT, CT_Tmax, CT_CBV, CT_CBF, and the maximal value of CT_4DPWI along time dimension, which we call PWI_max, and normalize them in the training process.

Additionally, We also perform on-the-fly data augmentation when feeding training samples. Augmentation operations include scaling, flipping, rotation, and translation are applied during the training.

2.2 Construction of Network

The whole workflow of our approach is shown in Fig. 1. The generator and segmentator are both designed based on U-Net [11] as illustrated in Fig. 4. It is a fully convolutional network, which consists of a encoder structure (left side) and

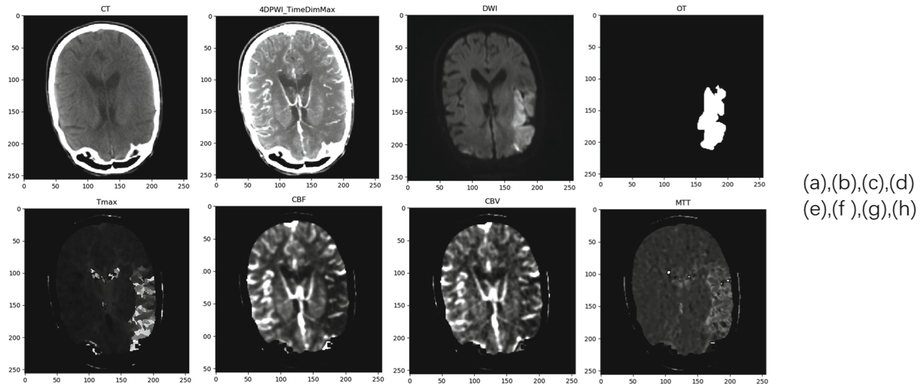

Fig. 2. An example of ISLES 2018 training data: (a) CT, (b) max value of 4DPWI, (c) DWI, (d) OT, (e) Tmax, (f) CBF, (g) CBV, (h) MTT.

a decoder structure (right side). The encoder part and decoder part consist of repeated layers of two 3 × 3 convolutions, with either MaxPooling in encoder part or UpConv layer in decoder part. With the limited GPU memory, a degradation problem is encounted in batch normalization due to the small batch size. Group normalization [13] is thus introduced in our network to solve this problem. GN's performance is relatively independent of the batch sizes used, and its accuracy is stable in a wide range of batch sizes. Inspired by the idea of maxout from Goodfellow et al. [14], maxout operation is performed in the final prediction. Normally, there are two output channels predicted for binary classification tasks, followed by softmax operation. Instead, We predict four layers, including three layers as background prediction for false positive reduction and one layer for foreground prediction. Then the max of three background channels are combined along channel dimension, to construct the final background channel for each pixel. We show a schematic drawing in Fig. 3. Through this maxout operation, many false positives could be reduced.

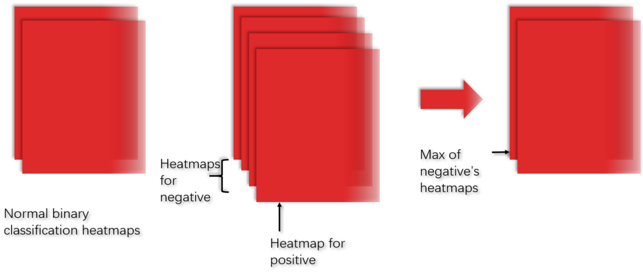

Fig. 3. Maxout for segmentation.

The discriminator in our work flow is in its simplest form, because we want to save the GPU memory used as much as we can. It only consists of five convolutional layers followed by group normalization [13], with Relu and a average pooling layer in the end. We use kernels with size 5 * 5 and stride 2 * 2, with no padding.

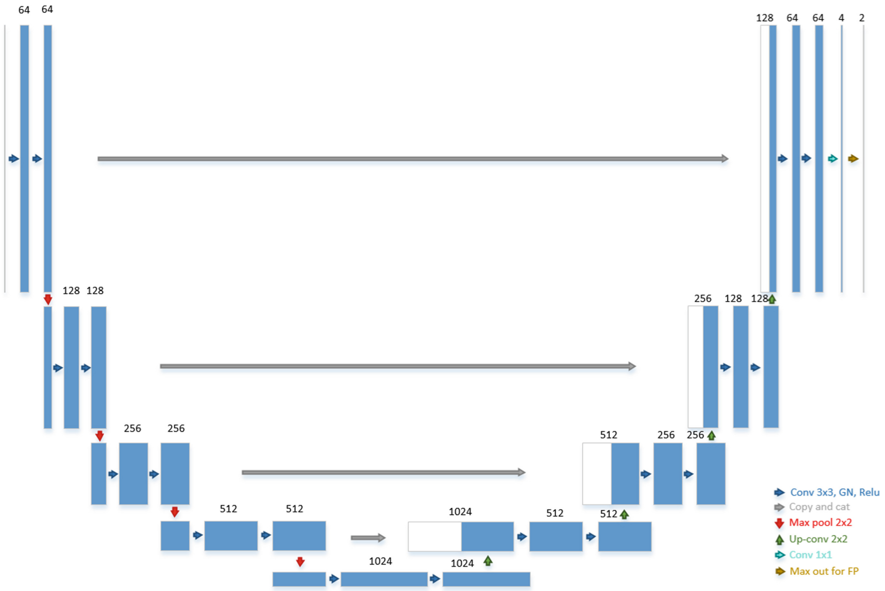

Fig. 4. U-Net with GN and Maxout, the backbone Structure of our generator and segmentator (Maxout is only used in segmentator).

2.3 Loss Function

Our loss function have three parts, which come from the generator, discriminator and segmentor, respectively. For the generator network, we measure how effectiveness the network transfer CT modalities to DWI using traditional mean square error (MSE) loss compared with the ground truth DWI. For the discrimination network, borrowed from LSGAN [15], we calculate the gap between prediction and ground truth via MSE Loss as well. During the training of the discriminator, DLoss is calculated with Eq. (2). In training of the generator and segmentator, DLoss is in the form of Eq. (3).

$$GLoss = MSELoss(Pred, Target) \tag{1}$$

$$DLoss = MSELoss(FakeDWI, 0) + MSELoss(TrueDWI, 1) \tag{2}$$

$$DLoss = MSELoss(FakeDWI, 1) \tag{3}$$

In the segmentation task, people usually use either cross entropy (CE) loss or dice loss alone. In Kaggle Carvana Image Masking Challenge, one challenger come up with a novel loss function result in a good performance with the form:

$$SegLoss = BCELoss - log(diceloss) \tag{4}$$

But in the training process, this loss is not stable when used directly. Then CE Loss combined with generalized dice loss [1] is introduced instead in the above equation:

$$SegLoss = CELoss - log(generalizeddiceloss) \tag{5}$$

We evaluate the gradient in training phase and found there would be a more reasonable ratio between positive and negative regions' gradients, bringing a more stable result. Intuitively, considering both foreground and background simultaneously is good for performance improvements.

Proper weights need to be set for each loss while training generator and segmentator. In the end, the loss function used is shown below.

$$Loss = \omega_g * GLoss + \omega_d * DLoss + \omega_s * SegLoss \tag{6}$$

3 Experiment

3.1 Implementation Details

We split the training data into four folds for cross-validation in the training phase. The input samples' size is (B, 5, 256, 256) where B is batch-size. First, we train the discriminator using generated DWI, which is synthesized via generator from the input CT data and the perfusion parameters, and the true DWI with pre-prepared labels. The discriminator's parameters were optimized using RMS optimizer with *DLoss* (Eq. (2)). Similar as in the discriminator, we also use RMS optimizer to optimize the generator and segmetator's parameters simultaneously, with *Loss* (Eq. (6)). The initial learning rate and γ is 0.0001 and 0.9, respectively. We train these two branches alternately 700 epoch in total. Learning rate will be halved in epoch 100, 300, 500. The ratio of weights, *GLoss*:*DLoss*:*SLoss*, is *0.002:0.5:1*. In GN layers, we fix the number of channels per group as 16. Because the GN layers can have different channel numbers, the group number can change across layers in this setting. The batch size is 6 in our exerments.

3.2 Results

We implemented our framework using PyTorch with cuDNN, and ran all experiments on a GPU server with 256 GB of memory, and a Nvidia GTX 1080Ti 11G GPUs. We analysed the segmentation loss function in the early stage. So the experiments result below are all based on the loss functions mentioned in Sect. 2.3.

Table 1. Result in different experiment stage.

Stage	First	Second	Third	Fourth
Dice	0.8473	0.5463	0.5829	0.6065

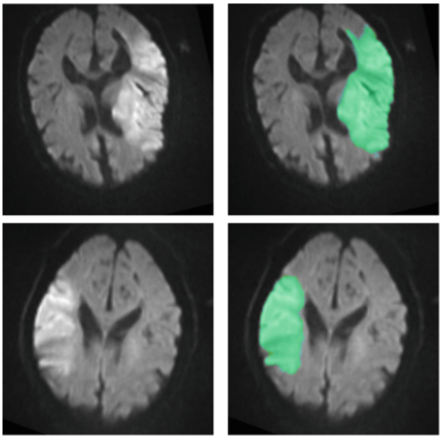

Fig. 5. Visualization of validation results of model trained on MR_DWI directly. Image in the left is the raw DWI image. DWI image overlayed by ground truth and prediction of model is in the right (green is ground truth, cyan is prediction). (Color figure online)

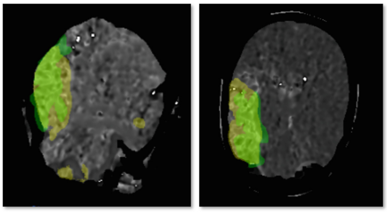

Fig. 6. Visualization of validation results of model trained on MR_MTT, MR_Tmax, MR_CBV, MR_CBF directly. MTT image overlayed by ground truth and prediction of model is used to show. (green is ground truth, yellow is prediction). (Color figure online)

The inspiration of our novel work flow is as the following. At the very begining, first stage (Table 1), we don't know there is no MR_DWI modality in testing phase. So MR_DWI is used as input to train the algorithm with a raw U-Net only. And the result is pretty good (Fig. 5) with dice coefficient equal to 0.8473. Then at the second stage (Table 1), we tried training a U-Net with MTT, Tmax, CBV and CBF, getting a much worse result of dice coefficient of 0.5463. The big gap in performance of segmentation between CTP image and DWI image drives me to transfer CTP images to a DWI image with a GAN approach. LSGAN [15] is a variation of traditional GAN [2] with a more stable performance and

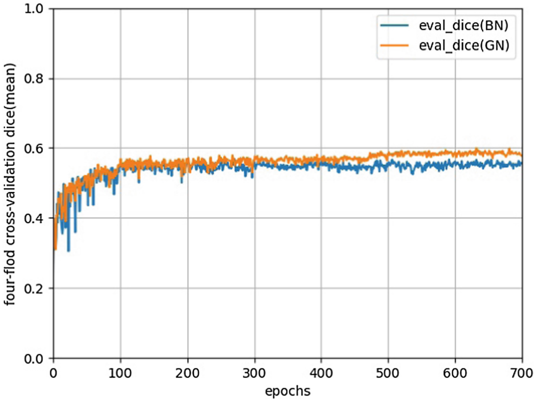

Fig. 7. Four-flod cross-validation dice (mean). Comparation between BN and GN.

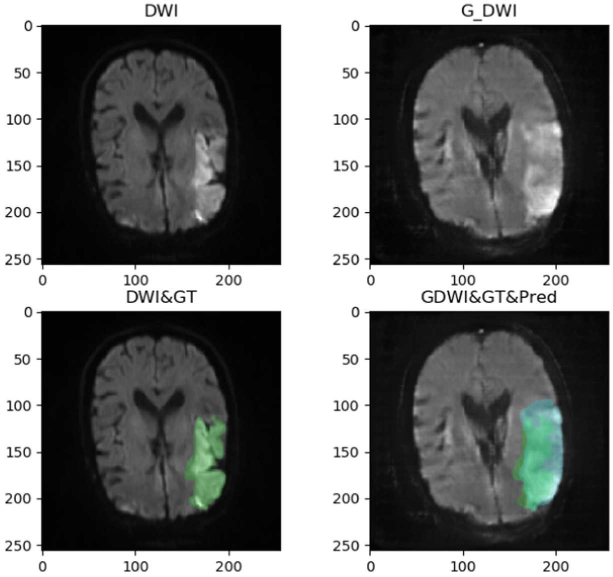

Fig. 8. Visualization of final result of model trained on our whole pipeline. (Top): Image in the left is the raw DWI image. Image in the right is our generalized DWI modality. (Bottom): Image overlayed by ground truth and prediction of model (green is ground truth, cyan is prediction). (Color figure online)

quicker convergence by changing log loss to L2 loss. In the third stage, after adding LSGAN in our work flow, the final dice coefficient score had a great improvement, from 0.5456 to 0.5811. Meanwhile, we concatenate a new channel, max of CT_4DPWI along time dimension, into the input. So, this work flow is determined to be our pipeline. In the fourth stage (Table 1), we mainly did some network structure modifications, like self attention, Maxout, GN, etc, to stable the training process and reduce the FP regions. The average dice score of four-fold cross-validation is 0.6065 in the end. The visualization of our final result is shown in Fig. 8. And the average dice score of the four fold validation sets is shown in Fig. 7. Batch normalization using BN or GN is also compared. The curve of network with GN is more stable than the curve of network with BN (Fig. 6).

4 Conclusion

In this paper, we propose a deep learning based method combined with adversarial networks to locate the region of ischemic stroke lesion automatically. And we proposed a novel pipeline for stroke lesion segmentation through DWI modality generation from CT perfusion data, which have a potentially promising prospective. This novel pipeline have a improvement over direct segmentation with a large margin. And a novel loss function is proposed as well, with maxout operation used in segmentation to achieve the state of art in ISLES challenge 2018.

References

1. Sudre, C.H., Li, W., Vercauteren, T., et al.: Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations (2017)
2. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems (2014)
3. Yu, Y., Han, Q., Ding, X., et al.: Defining core and penumbra in ischemic stroke: a voxel- and volume-based analysis of whole brain CT perfusion. Sci. Rep. **6**, 20932 (2015)
4. The top 10 causes of death (2017). http://www.who.int/mediacentre/factsheets/fs310/en/. Accessed 30 June 2017
5. Mouridsen, K., Hansen, M.B., Østergaard, L., Jespersen, S.N.: Reliable estimation of capillary transit time distributions using DSC-MRI. J. Cereb. Blood Flow Metab. **34**, 1511–1521 (2014). https://doi.org/10.1038/jcbfm.2014.111
6. Nielsen, A., Hansen, M.B., Tietze, A., et al.: Prediction of tissue outcome and assessment of treatment effect in acute ischemic stroke using deep learning. Stroke **49**, 1349–1401 (2018)
7. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: International Conference on Neural Information Processing Systems, pp. 1097–1105. Curran Associates Inc. (2012)
8. Ren, S., He, K., Girshick, R., et al.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2017)
9. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **PP**(99), 1 (2014)
10. Ciresan, D., Giusti, A., Gambardella, L.M., Schmidhuber, J.: Deep neural networks segment neuronal membranes in electron microscopy images. In: Advances in Neural Information Processing Systems, pp. 2843–2851 (2012)

11. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
12. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: International Conference on Artificial Intelligence and Statistics, pp. 315–323 (2011)
13. Wu, Y., He, K.: Group normalization (2018)
14. Goodfellow, I.J., Warde-Farley, D., Mirza, M., et al.: Maxout networks. arXiv preprint arXiv:1302.4389 (2013)
15. Mao, X., Li, Q., Xie, H., et al.: Least squares generative adversarial networks (2016)