3D Brain Tumor Segmentation Through Integrating Multiple 2D FCNNs

Xiaomei Zhao[1,2(✉)], Yihong Wu[1(✉)], Guidong Song[3], Zhenye Li[4], Yazhuo Zhang[3,4,5,6], and Yong Fan[7]

[1] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China
yhwu@nlpr.ia.ac.cn
[2] University of Chinese Academy of Sciences, Beijing, China
zhaoxiaomei14@mails.ucas.ac.cn
[3] Beijing Neurosurgical Institute, Capital Medical University, Beijing, China
[4] Department of Neurosurgery, Beijing Tiantan Hospital, Capital Medical University, Beijing, China
[5] Beijing Institute for Brain Disorders Brain Tumor Center, Beijing, China
[6] China National Clinical Research Center for Neurological Diseases, Beijing, China
[7] Department of Radiology, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, USA

Abstract. The Magnetic Resonance Images (MRI) which can be used to segment brain tumors are 3D images. To make use of 3D information, a method that integrates the segmentation results of 3 2D Fully Convolutional Neural Networks (FCNNs), each of which is trained to segment brain tumor images from axial, coronal, and sagittal views respectively, is applied in this paper. Integrating multiple FCNN models by fusing their segmentation results rather than by fusing into one deep network makes sure that each FCNN model can still be tested by 2D slices, guaranteeing the testing efficiency. An averaging strategy is applied to do the fusing job. This method can be easily extended to integrate more FCNN models which are trained to segment brain tumor images from more views, without retraining the FCNN models that we already have. In addition, 3D Conditional Random Fields (CRFs) are applied to optimize our fused segmentation results. Experimental results show that, integrating the segmentation results of multiple 2D FCNNs obviously improves the segmentation accuracy, and 3D CRF greatly reduces false positives and improves the accuracy of tumor boundaries.

Keywords: Brain tumor segmentation
Fully Convolutional Neural Networks · 3D Conditional Random Fields
Multi-views

1 Introduction

Brain tumor segmentation results provide the volume, shape, and localization of brain tumors, which are crucial for brain tumor diagnosis and monitoring.

A. Crimi et al. (Eds.): BrainLes 2017, LNCS 10670, pp. 191–203, 2018.
https://doi.org/10.1007/978-3-319-75238-9_17

Automatic brain tumor segmentation methods can emancipate doctors from the manual segmentation work, which is tedious and time-consuming [1]. Brain tumor segmentation technologies develop fast in recent years [2], especially those methods based on deep learning.

Up to now, many types of deep learning models have been successfully used in medical image analysis areas. According to statistics, segmentation is the most common subject among the literatures that apply deep learning to analyze medical images, and Convolutional Neural Networks (CNNs) are the most successful type of deep learning models for image analysis [3]. CNNs based methods have won many medical image segmentation challenges, such as Multimodal Brain Tumor Segmentation Challenge (BRATS) [4] and International Symposium on Biomedical Imaging (ISBI) cell tracking challenge [5].

Many kinds of medical images, such as the Magnetic Resonance Images (MRI), which can be used to segment brain tumors, are 3D images. To take full use of 3D information for medical image analysis, it is better to use 3D CNNs. However, 3D CNNs have large memory and training time requirements [6]. Therefore, many researchers have tried to integrate multi-view 2D CNNs for 3D medical images analysis, such as [6,7]. These methods integrated their multi-view 2D CNNs into one deep network, and sent the 2D patches in multi-views centered at the same voxel into their deep networks at the same time, predicting the label of these 2D patches' common center voxel. Under this situation, 3D images could only be segmented patch by patch, which is a very slow testing strategy, even if we change their CNNs into FCNNs. To improve the segmentation efficiency, multiple FCNN models, each of which is trained to segment slices in different views, can be combined by fusing their segmentation results, such as [8,9], rather than by fusing into one deep network. In this way, each FCNN model can still be tested by 2D slices, guaranteeing the testing efficiency. In this paper, we integrate multiple 2D FCNNs by integrating their segmentation results.

This paper is developed from our previous work [10]. In [10], 3 integrated networks of FCNNs and CRF-RNN [11] were used to segment brain images slice by slice from axial, coronal, and sagittal views respectively, and then their segmentation results were fused by voting. While in this paper, the segmentation results of multiple FCNNs are fused by averaging and then 3D CRF [12] is used to optimize the fused results. 3D CRF costs much more time than CRF-RNN, but it has a much better performance. The details of our method are given in the following sections.

2 Materials and Methods

2.1 Materials

We use the dataset provided by Multimodal Brain Tumor Segmentation Challenge (BraTS) 2017 [13–15] to train and test our segmentation method. The multimodal Magnetic Resonance Imaging (MRI) scans for each patient include native (T1), post-contrast T1-weighted (T1Gd), T2-weighted, and T2-weighted

fluid attenuated inversion recovery (Flair) volumes. Different from the datasets provided during BraTS 2014-2016, which include both pre- and post-operative scans, the dataset provided in this year only includes pre-operative MRI scans. BraTS 2017 has separated its dataset into 3 subsets: training subset, validation subset and testing subset. The training subset contains 210 High Grade Gliomas (HGG) cases and 75 Low Grade Gliomas (LGG) cases. The validation and the testing subsets contain 46 and 146 cases respectively, with unknown grades. All the ground truths of these subsets are produced by manual segmentation. Annotations include: enhancing tumor (label = 4), edema (label = 2), necrosis and non-enhancing tumor (label = 1), and others (label = 0). There is no tissue labeled as 3.

2.2 Methods

The proposed segmentation method consists of 5 main steps: pre-processing, segmenting brain images slice by slice from 3 different views using 3 2D FCNN models respectively, fusing segmentation results obtained in 3 different views, optimizing the fused segmentation results by 3D CRF, and post-processing. In the following we will introduce each of our segmentation steps in detail.

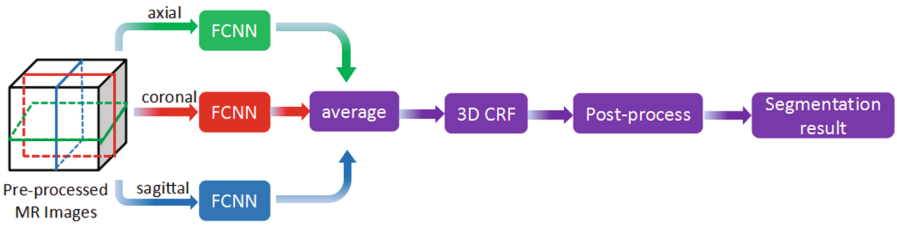

Fig. 1. Flowchart of our brain tumor segmentation method

Pre-process. To make similar intensities in MRI scans of the same modality have similar tissue meanings, pre-processing steps are utilized. Our pre-processing steps include N4ITK [16] and intensity normalization. The applied intensity normalization method normalize each image' intensity by normalizing its gray value of the highest histogram bin $\hat{I}$ and robust deviation $\tilde{\sigma} = \sqrt{\sum_{k=1}^{N}(I_k - \hat{I})^2/N}$ [10], where N denotes the total number of voxels in an image, I_k denotes the intensity value of the k^{th} voxel in the image. We also change intensity range of each image to 0-255 linearly. For more details of our pre-processing steps, please refer to [10].

Segment Brain Images by FCNNs. Our FCNNs brain tumor segmentation method is a patch-based segmentation method. A patch is a local region

extracted from an image. It can be 2D or 3D. In this paper, we use 2D square patches. A patch-based segmentation method transforms a segmentation task to a classification task. Patches are the objects to be classified. The label of a patch is same as the label of its center pixel. During testing, the traditional patch-based CNNs segmentation methods segment images by classifying patches one by one [17]. To accelerate the testing speed, Fully Convolutional Neural Networks (FCNNs), whose stride of each layer is set to 1, can be used. These FCNN networks can be trained by patches and tested by slices, improving the testing efficiency greatly [10,18].

In this paper, we use the same FCNN structure proposed in [19] as shown in Fig. 2, which has two different sizes of inputs. The large inputs pass through several convolutional and pooling layers and turn into small feature maps. These feature maps together with small inputs are used to predict their center pixel's label. Different from [19], we train 3 FCNN models in this paper, using 2D patches extracted from axial, coronal, and sagittal slices respectively. During testing, we use these 3 segmentation models to segment brain images slice by slice from 3 different views and obtain 3 segmentation probability maps.

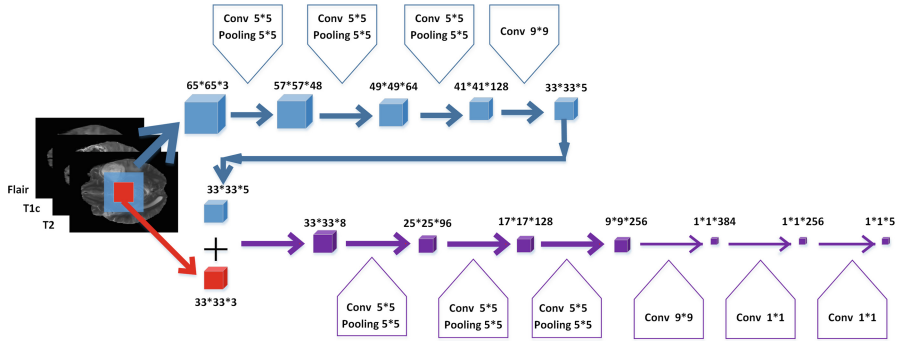

Fig. 2. The structure of the FCNN model used in this paper [19]

Fuse Segmentation Results Obtained in 3 Different Views. As described in the last subsection, 3 FCNN models are trained to segment brain images from 3 different views. During testing, their segmentation probability maps are fused by averaging to make better use of the 3D information provided by the 3D MRI scans.

Let P_a, P_c, and P_s denote the outputs of 3 the FCNN segmentation models respectively, $P_a = \{P_a^u | u = 0, 1, 2, 4\}$, $P_c = \{P_c^u | u = 0, 1, 2, 4\}$, $P_s = \{P_s^u | u = 0, 1, 2, 4\}$, where u denotes one of the four labels (enhancing tumor: label = 4; edema: label = 2; necrosis and non-enhancing tumor: label = 1; others: label = 0). P_a^u, P_c^u, and P_s^u are 3D probability images which have the same size as the 3D MRI scans to be segmented. The intensity value of each voxel in P_a^u, P_c^u, and P_s^u denotes the predicted probability of assigning label u to this voxel. We fuse these 3 segmentation results by averaging, that is $P = (P_a + P_c + P_s)/3$. P consists of

4 components $\{P^u|u = 0, 1, 2, 4\}$. Each $P^u = \{p_i^u\}$, where i is the index of voxel v_i in the 3D MRI scans and p_i^u is the fused predicted probability of assigning label u to voxel v_i.

We fuse the segmentation results as a post-processing step rather than fuse the 3 FCNN networks into one deep network, aiming to make sure that each FCNN model still has the ability to segment brain images slice by slice for efficiency. In this way, this method improves the efficiency of integrating multi-view 2D CNNs while achieves better accuracy than using a single 2D CNN network.

Optimize the Fused Segmentation Results by 3D CRF. To make sure the appearance and spatial consistency of segmentation results, we use fully connected 3D CRF [12] to optimize our segmentation results. CRF optimize the segmentation results by minimizing an energy function, which contains a unary term and a pairwise term. The energy function is shown as following:

$$E(a) = \sum_i \Phi(a_i^u) + \sum_{\forall i,j,i<j} \Psi(a_i^u, a_j^o) \tag{1}$$

where, i and j are indexes of two voxels v_i and v_j; a_i^u denotes assigning label u to voxel v_i, so does a_j^o; $\Phi(a_i^u)$ is the unary term, denoting the cost of assigning label u to voxel v_i; $\Psi(a_i^u, a_j^o)$ is the pairwise term, denoting the cost of assigning label u and o to voxels v_i and v_j respectively.

The unary term is calculated by $\Phi(a_i^u) = -\log p_i^u$, where, p_i^u denotes the fused probability of assigning label u to voxel v_i, which has been described in the last subsection. The pairwise term is formulated as a linear combination of Gaussian kernels [12]:

$$\Psi(a_i^u, a_j^o) = \mu(u, o)[\omega^{(1)} k^{(1)}(f_i, f_j) + \omega^{(2)} k^{(2)}(f_i, f_j)] \tag{2}$$

$$k^{(1)}(f_i, f_j) = exp(-\sum_d \frac{|s_{i,d} - s_{j,d}|^2}{2\sigma_{\alpha,d}^2}) \tag{3}$$

$$k^{(2)}(f_i, f_j) = exp(-\sum_d \frac{|s_{i,d} - s_{j,d}|^2}{2\sigma_{\beta,d}^2} - \sum_c \frac{|I_{i,c} - I_{j,c}|^2}{2\sigma_{\gamma,c}^2}) \tag{4}$$

$\mu(u, o)$ indicates the compatibility of labels u and o, $\mu(u, o) = [u \neq o]$; f_i and f_j denote feature vectors of v_i and v_j respectively, including their intensity $I_{i,c}$, $I_{j,c}$, and coordinates $s_{i,d}$, $s_{j,d}$; $c = flair$,$t1Gd$,or $t2$ indicates different MRI modalities; $d = x$, y, or z denotes different axles; $(s_{i,x}, s_{i,y}, s_{i,z})$ indicates the coordinate of voxel v_i, so does $(s_{j,x}, s_{j,y}, s_{j,z})$. Other parameters, such as $\omega^{(1)}$, $\omega^{(2)}$, $\sigma_{\alpha,d}$, $\sigma_{\beta,d}$, and $\sigma_{\gamma,c}$, are optimized by grid searching. The values of these parameters used in our experiments will be shown in Table 2 in Subsect. 3.1.

Post-process. We remove small isolated areas and correct some voxels' labels to post-process our segmentation results automatically by a simple thresholding method [10]. The thresholding parameters used in this paper have the same values as the parameters used in our previous paper [10].

3 Experiment

Our FCNN models are built upon caffe [20], and we use a computing server with multiple Tesla K80 GPUs and Intel E5-2620 CPUs. The dataset provided by BraTS 2017 are used to train and test our segmentation method. The details of this dataset have been described in Subsect. 2.1.

We use 80% of BraTS 2017 HGG training cases, including 168 HGG cases (hereinafter called 168 HGGs), to train our segmentation models, and use the rest of HGG training cases, including 42 HGG cases (hereinafter called 42 HGGs), as a validation dataset. Our FCNN networks are trained by patches and tested by slices. During training, we train 3 FCNN models, using 2D patches extracted from axial, coronal, and sagittal slices respectively. The numbers of extracted patches for different classes are equal. In our experiments, for the training of each view network, we extract $1000 * 4$ patches from each training case. Our FCNN networks need two different sizes of input patches, as shown in Fig. 2. The size of the larger patch is $65 * 65 * 3$, and the size of the smaller patch is $33 * 33 * 3$. Images in these 3 channels are extracted from pre-processed Flair, T1Gd, and T2 respectively. T1 scans are not used, because experiments in [10] showed that T1 scans couldn't improve segmentation performance. In our experiments, the training batch size is set to 300, and the initial learning rate is set to 10^{-4}. The learning rate is divided by 10 after each 20 epochs. During testing, slices are padded with 0 intensities before segmentation to make sure that the outputs of FCNN models have the same size as the original slices. For example, if the size of an original slice is $240 * 240$, we pad it with 0 intensities before segmentation to make two larger slices with sizes of $(240 + 64) * (240 + 64)$ and $(240 + 32) * (240 + 32)$ respectively. These two larger slices are used as the inputs of FCNN network, and then FCNN network outputs its prediction result with the size of $240 * 240$. The stride of each layer in our FCNN networks has been set to 1. That is why the outputs of our FCNN networks can have the same resolution as the original slices just by padding the original slices with 0 intensities before segmentation.

In our primary experiments that are tested on the 42 HGGs, we use Dice, Positive Predictive Value (PPV), and Sensitivity to measure the performance of our segmentation models. These 3 metrics are commonly used in BraTS evaluation websites: BraTS 2012[1], BraTS 2013[2], and BraTS 2015[3]. These metrics are calculated as following: $Dice(P_*, T_*) = \frac{|P_* \cap T_*|}{(|P_*| + |T_*|)/2}$, $PPV(P_*, T_*) = \frac{|P_* \cap T_*|}{|P_*|}$, $Sensitivity(P_*, T_*) = \frac{|P_* \cap T_*|}{|T_*|}$, where $*$ indicates complete, core, or enhancing regions. T_* denotes the true region of $*$. P_* denotes the segmented $*$ region. $|P_* \cap T_*|$ denotes the overlap area between P_* and T_*. $|P_*|$ and $|T_*|$ denotes the areas of P_* and T_* respectively. Particularly, the complete region includes enhancing core, edema, no-enhancing core and necrosis; the core region includes enhancing core, non-enhancing core and necrosis; the enhancing region only

[1] https://www.virtualskeleton.ch/BRATS/Start2012.
[2] https://www.virtualskeleton.ch/BRATS/Start2013.
[3] https://www.virtualskeleton.ch/BRATS/Start2015.

includes enhancing core. The 42 HGGs come from BraTS 2017 training dataset, and their ground truths are available. Therefore, we calculate the evaluation scores of the 42 HGGs by ourselves.

We also report our models' Dice scores on BraTS 2017 Validation dataset and Testing dataset. BraTS 2017 doesn't provide the ground truths of these two datasets. In this paper, the evaluation scores of Validation dataset are calculated on its evaluation website[4], and the evaluation scores of Testing dataset are provided by BraTS 2017 organizer. Even though there exist a number of LGG cases in BraTS 2017 Validation and Testing datasets, we still use our segmentation models trained only by the 168 HGGs to segment all the cases in these two datasets. Segmentation results of validation and testing datasets show that our models also work well for segmenting LGG cases. Therefor we only use HGG cases to train our segmentation models.

3.1 Primary Experiments Tested on the 42 HGGs

The evaluation scores of our models on the 42 HGGs are shown in Table 1, where the method called Fusing(FCNNs)+3D CRF fuses the segmentation results of 3 FCNN models by averaging, and then uses 3D CRF to optimize the fused results, as shown in Fig. 1; the method called Fusing(FCNNs+3D CRF) uses 3D CRF to optimize the segmentation results of each FCNN model, and then fuses the optimized results by voting.

Table 1. The average evaluation scores of 42 HGG cases

Methods	Dice			PPV			Sensitivity		
	Comp.	Core	Enh.	Comp.	Core	Enh.	Comp.	Core	Enh.
FCNNs(axial)	0.623	0.701	0.633	0.480	0.602	0.530	0.971	**0.916**	0.877
FCNNs(coronal)	0.666	0.738	0.675	0.578	0.664	0.594	0.963	0.896	0.865
FCNNs(sagittal)	0.662	0.703	0.653	0.526	0.607	0.574	0.957	0.912	0.846
Fusing(FCNNs)	0.713	0.805	0.733	0.585	0.753	0.663	**0.972**	0.912	0.887
Fusing(FCNNs)+3D CRF	0.865	0.867	0.821	0.912	**0.906**	0.800	0.841	0.856	0.884
Fusing(FCNNs)+3D CRF +post-process	**0.873**	**0.868**	**0.830**	**0.925**	0.904	**0.819**	0.843	0.860	0.877
FCNNs(axial)+3D CRF	0.857	0.843	0.787	0.873	0.840	0.731	0.857	0.875	**0.895**
FCNNs(coronal)+3D CRF	0.862	0.843	0.800	0.898	0.869	0.765	0.845	0.850	0.880
FCNNs(sagittal)+3D CRF	0.845	0.848	0.797	0.887	0.853	0.762	0.827	0.869	0.874
Fusing(FCNNs+3D CRF)	0.865	0.864	0.816	0.906	0.894	0.784	0.845	0.861	0.887
Fusing(FCNNs+3D CRF) +post-process	**0.873**	**0.868**	0.828	0.920	0.895	0.813	0.846	0.865	0.879

The scores in Table 1 indicate that fusing the segmentation results obtained from different views obviously improves the segmentation accuracy, no matter the fusing operation is performed before or after 3D CRF. But from the

[4] https://ipp.cbica.upenn.edu/.

view of Dice scores, Fusing(FCNNs)+3D CRF performs slightly better than Fusing(FCNNs+3D CRF). Note that, the performance of Fusing(FCNNs)+3D CRF reported in this paper is slightly different from the performance that we have reported in our pre-conference short paper of BraTS 2017 [21]. It is because that, at the beginning of our experiments, the same values of parameters in 3D CRF, which were optimized based on the experiments of Fusing(FCNNs+3D CRF), were applied in all of our experiments. But now, the parameters of 3D CRF are optimized separately in our experiments of Fusing(FCNNs)+3D CRF and Fusing(FCNNs+3D CRF). The values of parameters in 3D CRF used in our experiments are shown in Table 2. In addition, in [21], Fusing(FCNNs) fused segmentation results of FCNNs by voting, while in this paper, Fusing(FCNNs) fuses segmentation results of FCNNs by averaging.

Table 2. The values of parameters in 3D CRF used in our experiments (in this table, $d = x$, y,or z, and $c = flair$, $t1Gd$,or $t2$)

Experiments	$\omega^{(1)}$	$\omega^{(2)}$	$\sigma_{\alpha,d}$	$\sigma_{\beta,x}$	$\sigma_{\beta,y}$	$\sigma_{\beta,z}$	$\sigma_{\gamma,c}$
Fusing(FCNNs)+3D CRF	2.5	4.0	24	17	12	10	8
Fusing(FCNNs+3D CRF)	3.0	4.0	24	17	12	10	8

To show the effectiveness of integrating multiple FCNNs and multiple FCNNs+3D CRF, we show some segmentation examples in Fig. 3. Images in the first and second columns are used to show the effectiveness of fusing multiple FCNNs. Images in the third and fourth columns are used to show the effectiveness of fusing multiple FCNNs+3D CRF. Figure 3 shows that, fusing the 3 segmentation results of multi-view models can remove some obvious false positives, which just appear in one of the three results and do not appear in the other two results.

To show the effectiveness of 3D CRF and post-processing steps, we show some segmentation results in Fig. 4. Images in the first and second rows show that 3D CRF makes the segmentation labels have appearance and spatial consistency, and post-processing steps can remove false positives by removing small isolated areas. Images in the third and fourth rows show that 3D CRF also has the ability to remove many false positives caused by bias field. In this case, even though the segmentation result which has been optimized by 3D CRF still has much difference from the ground truth, it is already much better than the segmentation result without 3D CRF.

3.2 Segmentation Performance on BraTS 2017 Validation Dataset

We test our segmentation results of BRATS 2017 Validation dataset on its evaluation website. The Dice scores are shown in Table 3. Table 3 shows that, on BraTS 2017 Validation dataset, Fusing(FCNNs)+3D CRF+post-process has a better performance on enhancing region and core region, while Fusing(FCNNs+3D CRF)+post-process has a slightly better performance on complete tumor region.

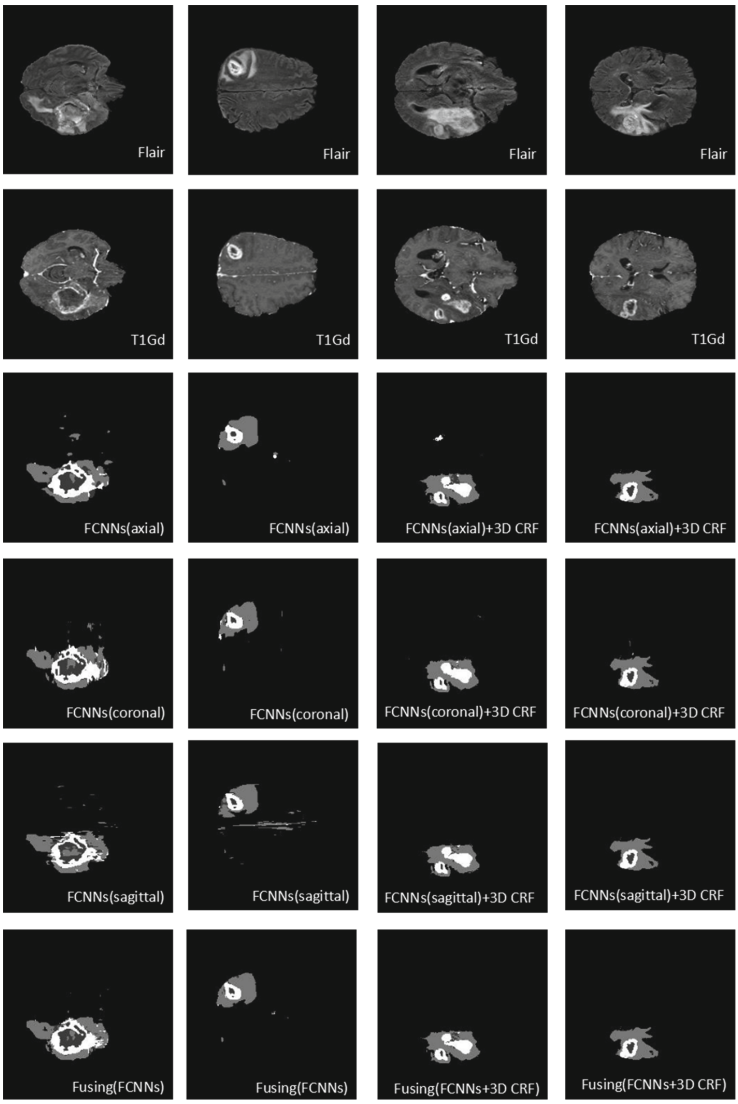

Fig. 3. Some segmentation examples to show the effectiveness of integrating multiple FCNNs and FCNNs+3D CRF. The type of each image has been labeled on the image. Images in the first and second columns are used to show the effectiveness of fusing multiple FCNNs. Images in the third and fourth columns are used to show the effectiveness of fusing multiple FCNNs+3D CRF. From left to right, the subjects ID are: Brats17_CBICA_ABE_1, Brats17_CBICA_AOZ_1, Brats17_CBICA_AME_1, Brats17_CBICA_AQD_1. In the segmentation results, each gray level represents a tumor class, from high to low: enhancing core, edema, necrosis and non-enhancing core.

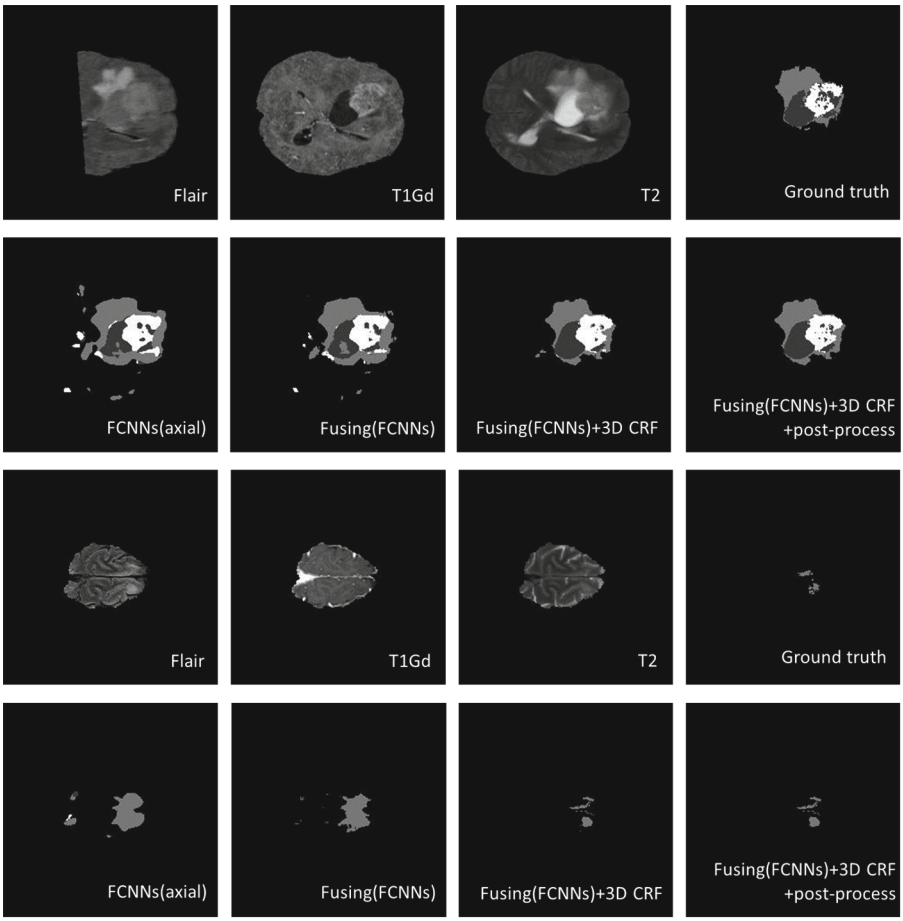

Fig. 4. Some segmentation examples to show the effectiveness 3D CRF and post-processing step. The type of each image has been labeled on the image. Images in the first and second rows come from the subject of Brats17_2013_20_1, and the images in the third and fourth rows come from the subject of Brats17_CBICA_ARZ_1. In the segmentation results, each gray level represents a tumor class, from high to low: enhancing core, edema, necrosis and non-enhancing core.

3.3 Segmentation Performance on BraTS 2017 Testing Dataset

The ground truths of BraTS 2017 Testing dataset are not available. The evaluation scores of our method on BraTS 2017 Testing dataset are provided by its organizer. We show the Dice scores of our method in Table 4.

During BraTS 2017, we used the segmentation model of Fusing(FCNNs)+3D CRF, not only because Fusing(FCNNs)+3D CRF has a slightly better performance, but also because Fusing(FCNNs)+3D CRF has a higher segmentation efficiency than Fusing(FCNNs+3D CRF). Fusing(FCNNs)+3D CRF only

Table 3. The average Dice scores of BRATS 2017 validation dataset (46 cases)

Methods	Dice		
	Comp.	Core	Enh.
FCNNs(axial)	0.648	0.601	0.543
FCNNs(coronal)	0.669	0.605	0.544
FCNNs(sagittal)	0.650	0.606	0.553
Fusing(FCNNs)	0.717	0.665	0.613
Fusing(FCNNs)+3D CRF	0.882	0.694	0.702
Fusing(FCNNs)+3D CRF+post-process	0.887	**0.794**	**0.754**
FCNNs(axial)+3D CRF	0.878	0.683	0.657
FCNNs(coronal)+3D CRF	0.871	0.677	0.647
FCNNs(sagittal) +3D CRF	0.870	0.694	0.668
Fusing(FCNNs+3D CRF)	0.881	0.694	0.696
Fusing(FCNNs+3D CRF)+post-process	**0.888**	0.792	0.749

Table 4. The average Dice scores of BRATS 2017 testing dataset (146 cases)

Methods	Dice		
	Comp.	Core	Enh.
Fusing(FCNNs)+3D CRF+post-process	0.876	0.752	0.764

performs 3D CRF once, while Fusing(FCNNs+3D CRF) performs 3D CRF three times. The 3D CRF is performed on CPU. Performing 3D CRF once costs about 3 min per case.

4 Conclusion

In this paper, 3D brain images are segmented by integrating the segmentation results of multiple 2D FCNNs, which are trained to segment brain images from axial, coronal, and sagittal views respectively. Each of the 2D FCNN networks is tested slice by slice, guaranteeing the segmentation efficiency. In addition, 3D CRF is used to optimize our fused segmentation results. Experimental results show that 3D CRF and the integrating strategy help a lot to improve segmentation accuracy. Moreover, this integrating method is not limited to fuse the 3 segmentation results from 3 different views. It can be extended to fuse the more from more views.

Acknowledgements. This work was supported by the National High Technology Research and Development Program of China (2015AA020504) and the National Natural Science Foundation of China under Grant No. 61572499, 61421004.

References

1. Bauer, S., Wiest, R., Nolte, L.-P., Reyes, M.: A survey of MRI-based medical image analysis for brain tumor studies. Phys. Med. Biol. **58**, 97–129 (2013)
2. Menze, B.H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., Wiest, R., et al.: The multimodal brain tumor image segmentation benchmark (BRATS). IEEE Trans. Med. Imaging **34**, 1993–2024 (2015)
3. Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., van der Laak, J.A., van Ginneken, B., Snchez, C.I.: A survey on deep learning in medical image analysis (2017). arXiv preprint arXiv:1702.05747
4. Chang, P.D.: Fully convolutional deep residual neural networks for brain tumor segmentation. In: Crimi, A., Menze, B., Maier, O., Reyes, M., Winzeck, S., Handels, H. (eds.) BrainLes 2016. LNCS, vol. 10154, pp. 108–118. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-55524-9_11
5. 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
6. Prasoon, A., Petersen, K., Igel, C., Lauze, F., Dam, E., Nielsen, M.: Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network. In: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (eds.) MICCAI 2013. LNCS, vol. 8150, pp. 246–253. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40763-5_31
7. Fritscher, K., Raudaschl, P., Zaffino, P., Spadea, M.F., Sharp, G.C., Schubert, R.: Deep neural networks for fast segmentation of 3D medical images. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9901, pp. 158–165. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46723-8_19
8. Mortazi, A., Karim, R., Rhode, K., Burt, J., Bagci, U.: *CardiacNET*: Segmentation of left atrium and proximal pulmonary veins from MRI using multi-view CNN. In: Descoteaux, M., Maier-Hein, L., Franz, A., Jannin, P., Collins, D.L., Duchesne, S. (eds.) MICCAI 2017. LNCS, vol. 10434, pp. 377–385. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66185-8_43
9. Wang, G., Li, W., Ourselin, S., Vercauteren, T.: Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks (2017). arXiv preprint arXiv: 1709.00382
10. Zhao, X., Wu, Y., Song, G., Li, Z., Zhang, Y., Fan, Y.: A deep learning model integrating FCNNs and CRFs for brain tumor segmentation. Med. Image Anal. **43**, 98–111 (2018)
11. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.H.: Conditional random fields as recurrent neural networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1529–1537 (2015)
12. Kamnitsas, K., Ledig, C., Newcombe, V.F., Simpson, J.P., Kane, A.D., Menon, D.K., Rueckert, D., Glocker, B.: Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. Med. Image Anal. **36**, 61–78 (2017)
13. Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J., Freymann, J., Farahani, K., Davatzikos, C.: Segmentation Labels and Radiomic Features for the Pre-operative Scans of the TCGA-GBM collection. The Cancer Imaging Archive (2017). https://doi.org/10.7937/K9/TCIA.2017.KLXWJJ1Q

14. Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J., Freymann, J., Farahani, K., Davatzikos, C.: Segmentation Labels and Radiomic Features for the Pre-operative Scans of the TCGA-LGG collection. The Cancer Imaging Archive (2017). https://doi.org/10.7937/K9/TCIA.2017.GJQ7R0EF
15. Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J.S., Freymann, J.B., Farahani, K., Davatzikos, C.: Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features. Sci. Data **4** (2017)
16. Tustison, N.J., Avants, B.B., Cook, P.A., Zheng, Y., Egan, A., Yushkevich, P.A., Gee, J.C.: N4ITK: improved N3 bias correction. IEEE Trans. Med. Imaging **29**, 1310–1320 (2010)
17. Pereira, S., Pinto, A., Alves, V., Silva, C.A.: Brain tumor segmentation using convolutional neural networks in MRI images. IEEE Trans. Med. Imaging **35**, 1240–1251 (2016)
18. Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Pal, C., Jodoin, P.-M., Larochelle, H.: Brain tumor segmentation with deep neural networks. Med. Image Anal. **35**, 18–31 (2017)
19. Zhao, X., Wu, Y., Song, G., Li, Z., Fan, Y., Zhang, Y.: Brain tumor segmentation using a fully convolutional neural network with conditional random fields. In: Crimi, A., et al. (eds.) Brainlesion Glioma Multiple Sclerosis Stroke and Traumatic Brain Injuries 2016. LNCS, vol. 10154, pp. 75–87. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-55524-9_8
20. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM International Conference on Multimedia, pp. 675–678. ACM (2014)
21. Zhao, X., Wu, Y., Song, G., Li, Z., Zhang, Y., and Fan, Y.: 3D brain tumor segmentation through integrating multiple 2D FCNNs. In: Pre-conference Proceedings of MICCAI-BraTS (Brain Tumor Segmentation Challenge), pp. 321–327 (2017)