IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 3, 2020, accepted January 23, 2020, date of publication February 13, 2020, date of current version February 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2973707

An Encoder-Decoder Neural Network With 3D Squeeze-and-Excitation and Deep Supervision for Brain Tumor Segmentation

PING LIU[1,3], QI DOU[2], (Member, IEEE), QIONG WANG[3], AND PHENG-ANN HENG[2], (Senior Member, IEEE)

[1]Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences, Shenzhen 518055, China
[2]Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong
[3]Shenzhen Key Laboratory of Virtual Reality and Human Interaction Technology, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China

Corresponding author: Qiong Wang (wangqiong@siat.ac.cn)

This work was supported in part by the National Natural Science Foundation of China under Project U1813204, and in part by the Shenzhen Science and Technology Program under Grant JCYJ20170413162256793.

ABSTRACT Brain tumor segmentation from medical images is a prerequisite to provide a quantitative and intuitive reference for clinical diagnosis and treatment. Manual segmentation depends on clinicians' experience, and is laborious and time-consuming. To tackle these issues, we proposed an encoder-decoder neural network, i.e. deep supervised 3D Squeeze-and-Excitation V-Net (DSSE-V-Net) to segment brain tumors automatically. We modified V-Net by adding batch normalization and using bottom residual block to make the network deeper. Then we incorporated a squeeze & excitation(SE) module in the modified V-Net by adding the SE block in each stage of the encoder and decoder, respectively. We also integrated 3D deep supervision seamlessly into the network to accelerate convergence. We evaluated our model on the public BraTS 2017 dataset for brain tumor segmentation. Our model outperformed both 3D U-Net and modified V-Net, and obtained highly competitive performance compared with those methods winning in the BraTS 2017 challenge.

INDEX TERMS Brain tumor segmentation, v-net, squeeze-and-excitation.

I. INTRODUCTION

Accurate brain tumor segmentation from medical images is a prerequisite to provide a quantitative and intuitive reference for clinical diagnosis and treatment of diseases. It is also the basis of quantitative disease progression assessment, treatment planning, and virtual surgery training systems. Magnetic resonance imaging (MRI) is a popular auxiliary means for diagnosis of brain tumors because of its high-quality images with high tissue contrast. Each MRI modality is good at differentiating some tissues. For example, the brain tissue structure can be clearly seen in T1c (T1 enhancement), while brain tumor boundaries are significantly enhanced in FLAIR. To make full use of these omplementary information, multi-model 3D MRIs are usually acquired.

At present, brain tumors are mainly manually labeled by the clinicians slice by slice. Manual segmentation relys on clinicians' experience. It is also tedious, time-consuming and of poor repeatability. Therefore, there is a demand for accurate and efficient automated brain tumor segmentation approaches. However, this is challenging as a result of the great tumor intensity changes, variable and irregular tumor shape, size, and localization, and unclear boundaries to normal brain tissues. The Multi-modal Brain Tumor Image Segmentation Benchmark (BraTS) challenges were hold along with MICCAI since 2012 [1] to foster sdudy in this area. The challenges indicated that deep learning based segmentation approaches showed superiority compared with traditional segmentation approaches [1], [2].

Specifically, convolutional neural networks (CNN) approaches especially V-Net [3] and U-Net [4], [5] based architectures demonstrated amazing performance in medical image segmentation tasks. These architectures both adopted an encoder-decoder structure with several stages and skip connection in the same stage. The whole structure with skip connection allowed the feature map to incorporate more

The associate editor coordinating the review of this manuscript and approving it for publication was K. C. Santosh.

low-level features, which also enables the fusion of features of different scales for multi-scale prediction. Recently, squeeze & excitation(SE) module [6] was put forward to enhance important features by recalibrating the feature maps adaptively. It improves the result in ImageNet classification competition greatly when added to existing neural network architectures, such as ResNet-50 and Inception. It also improved segmentation accuracy as demonstrated in [7].

Motivated by these works, we presented a Deep supervised 3D Squeeze-and-excitation V-Net (DSSE-V-Net) for automated brain tumor segmentation from multi-model magnetic resonance images (MRIs). Similar to standard V-Net, our modified V-Net follows the encoder-decoder structure of CNN. We added batch normalisation and replaced some residual block [8] in the original V-Net with the bottom residual block in some stages to make the network deeper. We incorprated squeeze & excitation(SE) module in the modified V-Net by including a SE block in each stage of the encoder and decoder, respectively. To accelerate network's convergence, we integrated deep supervision in the 3D DSSE-V-Net. We evaluated our model on the public BraTS 2017 datasets [9] in brain tumor segmentation. Our model outperformed both 3D U-Net and modified V-Net, and obtained highly competitive performance compared with those methods winning in the BraTS 2017 challenge.

II. RELATED WORKS

Most contemporary effective brain tumor segmentation methods utilized convolutional neural networks. These methods could be mainly classified into three categories. The first class focuses on single segmentation network based on encoder-decoder architecture to segment various tumor tissues simultaneously. Myronenko [10] put forward an encoder-decoder architecture and added another decoder path to recover the input image. The added decoder acted as a regularization of the shared encoder by imposing more constraints. They obtained the first place in the 2018 BraTS challenge. Isensee *et al.* [11] adopted the popular U-net architecture by making a little changes and used more training data collected in their own institution, resulting competitive performance. McKinley *et al.* [12] modified a U-net-like structure by incorporating a DenseNet [13] module with dilated convolutions. Pereira *et al.* [14] proposed a segmentation network with feature recalibration based on the idea of squeeze & excitation(SE).

The second type takes cascaded networks to segment tumor subregions sequentially. Wang *et al.* [15] designed a network structure and trained the network for the whole tumor first, then trained a similar network for a tumor subregion with cropped 3D bounding box of the entire tumor in the original image as input. At last they trained another similar network for another tumor subregion with cropped 3D bounding box of the above tumor subregion. In this cascaded way, they obtained a second in BraTS 2017 challenge. Lachinov [16] trained multiple 3D U-Net cascadely to segment tumor subregions sequentially. They used separate encoders for each individual input modality and introduced a method to merge encoded feature maps to solve the problem of heterogeneous input. The third type fusions several segmentation models together to obtain good performance. Kamnitsas *et al.* [17] proposed an ensemble of different models and achieved the first place in BraTS 2017 challenge. Zhou *et al.* [18] segmented three tumor subregions in a cascaded way and utilized an ensemble of different models as well.

III. METHODS

A. NETWORK ARCHITECTURE

We built our network 3D DSSE-V-Net based on the V-Net in [3]. We improved V-Net in a few ways to make it deeper and more robust. Motivated by the success of squeeze-and-excitation module on image object classification, we designed and included 3D squeeze-and-excitation (SE) block in our modified V-Net. In addition, we incorporated deep supervision in the DSSE-V-Net to accelerate convergence. Our network architecture is given in Fig. 1 and will be detailed in the following subsections. The detailed information such as the channels of each convolution about the network are presented in Table 1.

1) MODIFIED V-NET

The standard V-Net follows the encoder-decoder architecture. The encoder extracts multi-scale features at different stages, while the decoder focuses on the segmentation objects and reconstructs the features to the original size gradually. Similar to V-Net, our modified V-Net had four stages in addition to an input transform stage, and each stage operated at different resolutions. At each stage, the encoder progressively reduced image resolutions by a convolution and increased feature maps number by 2 at the same time. We added batch normalization after this convolution, and used ELU for nonlinear operation instead of the original PRelu, these operations were denoted as DownConvBnElu in Fig. 1. The convolution kernel size was $2 \times 2 \times 2$ and its stride was $2 \times 2 \times 2$. Following the DownConvBnElu, we used two residual blocks with an additive identity skip connection followed by an ELU operation, as seen in the green dotted rectangle in Fig. 1 and Fig. 2(a). The residual block consisted of a few ConvBnElus, where the size of all convolutions were $5 \times 5 \times 5$ and the stride was $1 \times 1 \times 1$. The residual connections in the residual block encourage training more deep networks [19]. The batch normalization and ELU were utilized to make the network easier for training.

To make the network deeper to extract more useful features, we used a bottom residual block replacing a residual block in the 2-4 stage. The bottom residual block consists several bottom convolution units, where each unit has three ConvBnElu, as seen in Fig. 2(b). And kernel sizes of the three convolutions were $1 \times 1 \times 1$, $3 \times 3 \times 3$ and $1 \times 1 \times 1$, respectively.

The decoder's structure was analogous to the encoder. The only difference was that each decoder stage began with a 3D transposed convolution to increase the resolution of feature

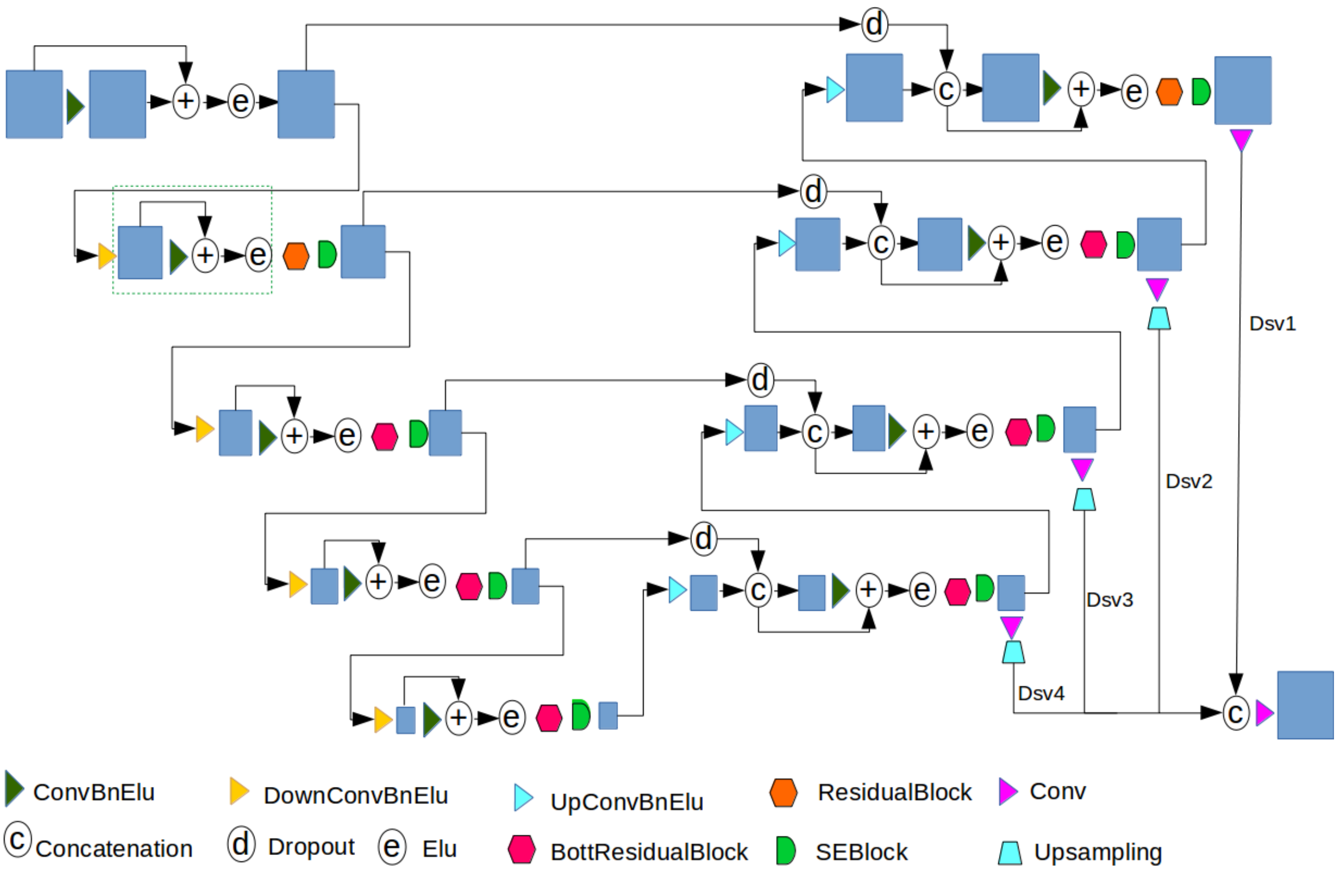

FIGURE 1. The network architecture of our 3D DSSE-V-Net.

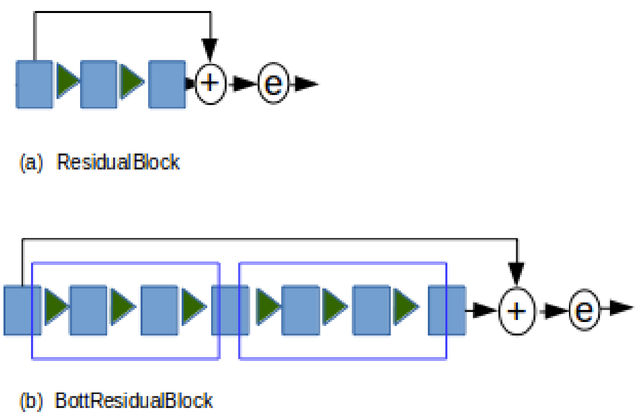

FIGURE 2. (a) A residual block with two ConvBnRelu, (b) A bottom residual block with bottom convolution unit, and each unit has three ConvBnRelu.

maps by 2 while decrease the feature maps number by 2. And a skip connection, i.e, a concatenation of each encoder stage's output with a dropout and the feature maps of the corresponding decoder stage was performed to incorporate more low-level features. Then, the same bottom residual block as in the corresponding encoder stage was used.

2) MODIFIED V-NET WITH SQUEEZE-AND-EXCITATION(SE) BLOCK

SE block was initially presented in [6], which improved channel interdependencies at almost no computational cost. It is a feature recalibration process to selectively enhance useful feature maps through adaptively adjusting the weighting of each feature map. The SE block is shown in Fig. 3(a) (For clarity, 2D feature maps are used). For any given CNN feature maps U, a global pooling operation was used to obtain a weight matrix sized $1 \times 1 \times 1 \times ch$. ch denotes the the convolutional channels number. Then an excitation operation was performed. The excitation operation consisted a fully connected layer, a ReLU adding necessary nonlinearity, and a sigmoid function following a second fully connected layer to rescale the activations to [0, 1]. This produced a gathering of per-channel adjusting parameters. The output of the SE block was obtained by multiplying the parameters to U.

The above SE block generated per-channel modulation weights, so it could be denoted as cSE. Motivated by cSE, spatial Squeese and Excitation Block (sSE) was presented in [7]. sSE was a recalibration by squeezing the feature maps spatially. This process enhanced the salient spatial locations. The sSE block is given in Fig. 3 (b). For more details, please refer to [7]. As illustrated in Fig.3 (c), [7] also combined the

TABLE 1. The detailed information such as the channels of each convolution about the 3D DSSE-V-Net. Batch normalization, ELU operation and the SEBlock in each stage of the encoder and decoder are not listed in this table for simplicity.

Stage	Layer name	In channel size	Out channel size	Kernel size	Stride	Padding	Output size
InTr	Input data	4	-	-	-	-	$128 \times 128 \times 128 \times 4$
	Conv1	4	16	$5 \times 5 \times 5$	$1 \times 1 \times 5$	-	$128 \times 128 \times 128 \times 16$
DownTr32	DownConv	16	32	$2 \times 2 \times 2$	$2 \times 2 \times 2$	-	$128 \times 128 \times 128 \times 4$
	Conv1	4	16	$5 \times 5 \times 5$	$1 \times 1 \times 1$	-	$64 \times 64 \times 64 \times 32$
	ResdualBlock1: Conv1	32	32	$5 \times 5 \times 5$	$1 \times 1 \times 1$	$2 \times 2 \times 2$	$64 \times 64 \times 64 \times 32$
	ResdualBlock2: Conv1	32	32	$3 \times 3 \times 3$	$1 \times 1 \times 1$	$1 \times 1 \times 1$	$64 \times 64 \times 64 \times 32$
DownTr64	DownConv	32	64	$2 \times 2 \times 2$	$2 \times 2 \times 2$	-	$32 \times 32 \times 32 \times 64$
	ResdualBlock1:Conv1	64	64	$5 \times 5 \times 5$	$1 \times 1 \times 1$	$2 \times 2 \times 2$	$32 \times 32 \times 32 \times 64$
	ResdualBlock1:Conv2	64	64	$5 \times 5 \times 5$	$1 \times 1 \times 1$	$2 \times 2 \times 2$	$32 \times 32 \times 32 \times 64$
	BottResidualBlock:Conv11	64	16	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$32 \times 32 \times 32 \times 16$
	BottResidualBlock:Conv12	16	16	$3 \times 3 \times 3$	$1 \times 1 \times 1$	$1 \times 1 \times 1$	$32 \times 32 \times 32 \times 16$
	BottResidualBlock:Conv13	16	64	$3 \times 3 \times 3$	$1 \times 1 \times 1$	-	$32 \times 32 \times 32 \times 64$
	BottResidualBlock:Conv21	64	16	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$32 \times 32 \times 32 \times 16$
	BottResidualBlock:Conv22	16	16	$3 \times 3 \times 3$	$1 \times 1 \times 1$	$1 \times 1 \times 1$	$32 \times 32 \times 32 \times 16$
	BottResidualBlock: Conv23	16	64	$3 \times 3 \times 3$	$1 \times 1 \times 1$	-	$32 \times 32 \times 32 \times 64$
DownTr128	DownConv	64	128	$2 \times 2 \times 2$	$2 \times 2 \times 2$	-	$16 \times 16 \times 16 \times 128$
	ResdualBlock1:Conv1	128	128	$5 \times 5 \times 5$	$1 \times 1 \times 1$	$2 \times 2 \times 2$	$16 \times 16 \times 16 \times 128$
	ResdualBlock1:Conv2	128	128	$5 \times 5 \times 5$	$1 \times 1 \times 1$	$2 \times 2 \times 2$	$16 \times 16 \times 16 \times 128$
	BottResidualBlock: Conv11	128	32	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$16 \times 16 \times 16 \times 32$
	BottResidualBlock: Conv12	32	32	$3 \times 3 \times 3$	$1 \times 1 \times 1$	$1 \times 1 \times 1$	$16 \times 16 \times 16 \times 32$
	BottResidualBlock: Conv13	32	128	$3 \times 3 \times 3$	$1 \times 1 \times 1$	-	$16 \times 16 \times 16 \times 128$
	BottResidualBlock: Conv21	128	32	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$16 \times 16 \times 16 \times 32$
	BottResidualBlock: Conv22	32	32	$3 \times 3 \times 3$	$1 \times 1 \times 1$	$1 \times 1 \times 1$	$16 \times 16 \times 16 \times 32$
	BottResidualBlock: Conv23	32	128	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$16 \times 16 \times 16 \times 128$
DownTr256	DownConv	128	256	$2 \times 2 \times 2$	$2 \times 2 \times 2$	-	$8 \times 8 \times 8 \times 256$
	ResdualBlock1:Conv1	256	256	$5 \times 5 \times 5$	$1 \times 1 \times 1$	$2 \times 2 \times 2$	$8 \times 8 \times 8 \times 256$
	ResdualBlock1:Conv2	256	256	$5 \times 5 \times 5$	$1 \times 1 \times 1$	$2 \times 2 \times 2$	$8 \times 8 \times 8 \times 256$
	BottResidualBlock:Conv11	256	64	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$8 \times 8 \times 8 \times 64$
	BottResidualBlock:Conv12	64	64	$3 \times 3 \times 3$	$1 \times 1 \times 1$	$1 \times 1 \times 1$	$8 \times 8 \times 8 \times 64$
	BottResidualBlock:Conv13	64	256	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$8 \times 8 \times 8 \times 256$
	BottResidualBlock:Conv21	256	64	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$8 \times 8 \times 8 \times 64$
	BottResidualBlock:Conv22	64	64	$3 \times 3 \times 3$	$1 \times 1 \times 1$	$1 \times 1 \times 1$	$8 \times 8 \times 8 \times 64$
	BottResidualBlock:Conv23	64	256	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$8 \times 8 \times 8 \times 256$
UpTr256	UpConv	256	128	$2 \times 2 \times 2$	$2 \times 2 \times 2$	-	$16 \times 16 \times 16 \times 256$
	ResdualBlock1:Conv1	256	256	$5 \times 5 \times 5$	$1 \times 1 \times 1$	$2 \times 2 \times 2$	$16 \times 16 \times 16 \times 256$
	ResdualBlock1:Conv2	256	256	$5 \times 5 \times 5$	$1 \times 1 \times 1$	$2 \times 2 \times 2$	$16 \times 16 \times 16 \times 256$
	BottResidualBlock: Conv11	256	64	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$16 \times 16 \times 16 \times 64$
	BottResidualBlock: Conv12	64	64	$3 \times 3 \times 3$	$1 \times 1 \times 1$	$1 \times 1 \times 1$	$16 \times 16 \times 16 \times 64$
	BottResidualBlock: Conv13	64	256	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$16 \times 16 \times 16 \times 256$
	BottResidualBlock: Conv21	256	64	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$16 \times 16 \times 16 \times 64$
	BottResidualBlock: Conv22	64	64	$3 \times 3 \times 3$	$1 \times 1 \times 1$	$1 \times 1 \times 1$	$16 \times 16 \times 16 \times 64$
	BottResidualBlock: Conv23	64	256	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$16 \times 16 \times 16 \times 256$
UpTr128	UpConv	256	64	$2 \times 2 \times 2$	$2 \times 2 \times 2$	-	$32 \times 32 \times 32 \times 128$
	ResdualBlock1:Conv1	128	128	$5 \times 5 \times 5$	$1 \times 1 \times 1$	$2 \times 2 \times 2$	$32 \times 32 \times 32 \times 128$
	ResdualBlock1:Conv2	128	128	$5 \times 5 \times 5$	$1 \times 1 \times 1$	$2 \times 2 \times 2$	$32 \times 32 \times 32 \times 128$
	BottResidualBlock:Conv11	128	32	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$32 \times 32 \times 32 \times 32$
	BottResidualBlock:Conv12	32	32	$3 \times 3 \times 3$	$1 \times 1 \times 1$	$1 \times 1 \times 1$	$32 \times 32 \times 32 \times 32$
	BottResidualBlock:Conv13	32	128	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$32 \times 32 \times 32 \times 128$
	BottResidualBlock:Conv21	128	32	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$32 \times 32 \times 32 \times 32$
	BottResidualBlock:Conv22	32	32	$3 \times 3 \times 3$	$1 \times 1 \times 1$	$1 \times 1 \times 1$	$32 \times 32 \times 32 \times 32$
	BottResidualBlock:Conv23	32	128	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$32 \times 32 \times 32 \times 128$
UpTr64	UpConv	128	32	$2 \times 2 \times 2$	$2 \times 2 \times 2$	-	$64 \times 64 \times 64 \times 64$
	ResdualBlock1:Conv1	64	64	$5 \times 5 \times 5$	$1 \times 1 \times 1$	$2 \times 2 \times 2$	$64 \times 64 \times 64 \times 64$
	ResdualBlock1:Conv2	64	64	$5 \times 5 \times 5$	$1 \times 1 \times 1$	$2 \times 2 \times 2$	$64 \times 64 \times 64 \times 64$
	BottResidualBlock:Conv11	64	16	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$64 \times 64 \times 64 \times 16$
	BottResidualBlock:Conv12	16	16	$3 \times 3 \times 3$	$1 \times 1 \times 1$	$1 \times 1 \times 1$	$64 \times 64 \times 64 \times 16$
	BottResidualBlock:Conv13	16	64	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$64 \times 64 \times 64 \times 64$
	BottResidualBlock:Conv21	64	16	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$64 \times 64 \times 64 \times 16$
	BottResidualBlock:Conv22	16	16	$3 \times 3 \times 3$	$1 \times 1 \times 1$	$1 \times 1 \times 1$	$64 \times 64 \times 64 \times 16$
	BottResidualBlock:Conv23	16	64	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$64 \times 64 \times 64 \times 64$
UpTr32	UpConv	64	16	$2 \times 2 \times 2$	$2 \times 2 \times 2$	-	$128 \times 128 \times 128 \times 32$
	ResdualBlock1:Conv1	32	32	$5 \times 5 \times 5$	$1 \times 1 \times 1$	$2 \times 2 \times 2$	$128 \times 128 \times 128 \times 32$
	ResidualBlock2:Conv1	32	32	$3 \times 3 \times 3$	$1 \times 1 \times 1$	$1 \times 1 \times 1$	$128 \times 128 \times 128 \times 32$
Dsv4	Conv1+Upsample	256	4	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$128 \times 128 \times 128 \times 4$
Dsv3	Conv1+Upsample	128	4	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$128 \times 128 \times 128 \times 4$
Dsv2	Conv1+Upsample	64	4	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$128 \times 128 \times 128 \times 4$
Dsv1	Conv1	32	4	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$128 \times 128 \times 128 \times 4$
Final	Conv1	16	4	$1 \times 1 \times 1$	$1 \times 1 \times 1$	-	$128 \times 128 \times 128 \times 4$

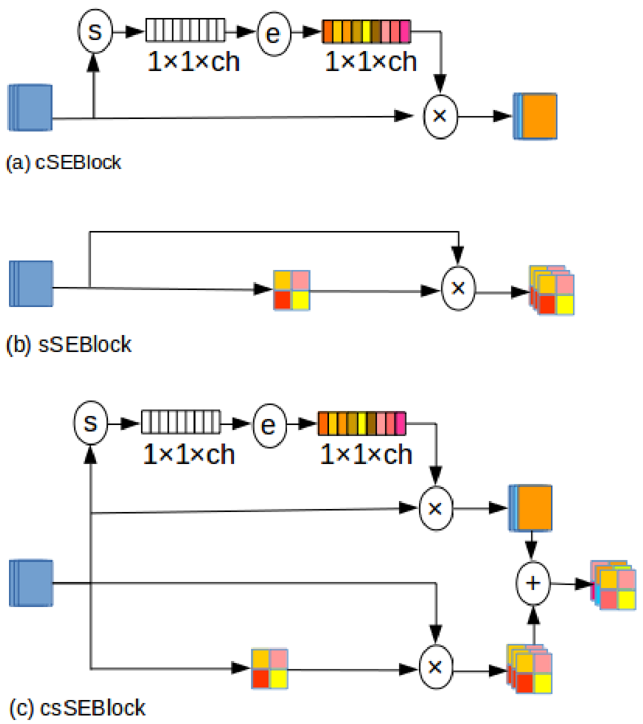

FIGURE 3. (a) Channel Squeeze & Excitation Block (cSE), (b) Space Squeeze & Excitation Block (sSE), (c) Space and channel Squeeze & Excitation. For clarity, 2D feature mas are used in the Figure.

cSE block and sSE block to generate a scSE block by adding the channel and spatial excitation element-wisely. The scSE block is supposed to concurrently encourage the network to learn more relevant features both spatially and channel-wise.

We implemented 3D cSE block, sSE block and csSE block, and added them after the second residual block or bottom residual block in each encoder and decoder stage, respectively, as seen in Fig.1. The SE block was supposed to facilitate the network focus on more informative features maps in each scale and improve segmentation accuracy.

3) MODIFIED V-NET WITH SE BLOCK AND DEEP SUPERVISION

Relatively deep networks are usually used to encode highly representative features. Deep supervision is helpful to reduce overfitting and facilitate network convergence when training a deep neural network [20]. It is also useful to extract more meaningful features [21]. Here, deep supervision were utilized in each stage, so that the output of the middle stage can be directly utilized as a supervision. Specifically, the outputs of each decoder stage were resized to the same size as the training patches by upsampling, resulting in Dsv4, Dsv3, Dsv2, and Dsv1, as seen in Fig. 1. Then, Dsv4, Dsv3, Dsv2, and Dsv1 with the same size were concatenated, and a convolution with kernel size $1 \times 1 \times 1$ was used to fusion all the outputs with different scales. Finally, the probability map was obtained by a softmax operation. The loss could be calculated between ground truth and the softmax output. In this way, the outputs of the middle stages and the final stage all contributed to the loss and gradients back propagation implicitly, which forced the update process of the middle stage filters to favor highly discriminative features.

B. LOSS FUNCTION

The model was supposed to predict whether a pixel was a specific brain tumor tissue or background. Our model output was the segmentation mask of the input. The cross-entropy (CE) loss was popularly used to train CNN segmentation models. Because brain tumor especially the tumor core are usually rather small in the whole image, the CE loss is not good at this unbalanced segmentation problem. The well-known Dice overlap coefficient was also adopted as a regional loss function, outperforming CE in this kind of tasks [3]. The Dice coefficient for multi-class segmentation can be calculated by:

$$L_{Dice} = 1 - \sum_{c=1}^{C} \frac{2\omega_c \sum_{i=1}^{N} pm(c,i)gt(c,i)}{\sum_{i=1} pm(c,i)^2 + gt(c,i)^2} \tag{1}$$

where N is the voxel number,$pm(c, i) \in [0, 1]$ and $gt(k, i) \in \{0, 1\}$ represents the softmax output and the one-hot encoder of the ground truth label for class c, respectively. C is the class of brain tumor tissues. $\sum c\omega_c = 1$ where ω_c are the class weights. It was set to $\omega_c = 1/C$ empirically.

We also tried the Lovasz loss, differentiable surrogate for optimizing Intersection-over-union(IoU) recently proposed in [22], but found no clear performance improvement compared with Dice.

IV. EXPERIMENTS AND DISCUSSIONS

We first did experiments with different model designs on BraTS 2017 dataset. Then we reported and discussed results of our best model and those of the published top approaches.

A. EVALUATION DATASETS

BraTS 2017 dataset consisted multi-modal MRIs from multiple institutions. It aimed for intrinsically heterogeneous brain tumors segmentation. There were a training, validation and test dataset. The training dataset included 285 samples with manually annotated and confirmed ground truth labels. For each sample, four modalities, T1, T1c, T2 and FLAIR sized $240 \times 240 \times 155$ and the corresponding annotations were provided. The BraTS 2017 validation dataset consisted 46 cases without given the annotations.

B. DATA PREPROCESSING AND AUGMENTATION

The MR images have artifacts because of different imaging protocols and equipments [23]. The same as in [2], we used N4BiasFieldCorrection algorithm in ITK [24] for the T1, T1c and T2 modalities for bias correction. Then normalization was performed the same as that in [25] for each modality respectively.

Considering the GPU memory limit, we randomly cropped patches sized $128 \times 128 \times 128$ within the brain tissue mask.

We adopted a simple data augmentation strategy by randomly flipping each patch along a randomly selected axis with a probability 0.5.

C. IMPLEMENTATION DETAILS

We adopted PyTorch [26] to implement all models. Our workstation was equipped with four NVIDIA Titan 1080 TI 11GB GPUs. As there were four MRI modalities, we concatenated the images obtaining a four-channel image as input. With the random crop operation, our input patch size was $4 \times 128 \times 128 \times 128$.

As previously described, the BraTS 2017 dataset included 285 training samples and 46 validation samples. We trained all models on the same 275 cases and evaluated their performance on the validation cases. The ground truth of the validation samples were not offered. So we submitted our segmentations of all models to the BraTS 2017 on line evaluation system and obtained quantitative evaluations in term of Dice and Hausdorff distances of each tumor tissue class to compare with other methods.

We trained the network from scratch with the He normal weight initialization [27] for the convolutional kernel parameters. We used adam optimizer and dice loss for optimization. The initial learning rate was $lr = 2e4$. The batch size was 4 according to our GPU number to enable data parallelism. The input images were shuffled to ensure that each training case was selected once per epoch. The dropout before the SE block in each decoder stage was to 0.2. The hyper-parameters were chosen experimentally. We trained all models with the same hyper-parameters. The total training epoch was 1000. It took about 3 minutes per epoch and about 43 hours for training the DScSE-V-Net model on four NVIDIA GPUs.

For prediction, each sample was cropped regularly with stride $16 \times 16 \times 16$. There were 192 patches for a sample sized $4 \times 255 \times 255 \times 255$. We combined outputs of all the patches of a sample to reconstruct probability maps to the same volume size of the sample, and obtained the segmentation masks from the reconstructed probability maps.

D. EXPERIMENT RESULTS

We implemented 3D U-Net, 3D U-Net with deep supervision (DS), and the modified V-Net with DS. We also tried the attention-gated U-Net in [28] and added the attention gates in V-Net as similar as that in U-Net. Finally, we integrated the squeeze & excitation(SE) module in our model by adding the cSE block and the csSE block in each encoder and decoder stage, respectively. Thus there were seven models in total and we called them (1) 3D U-Net, (2) DS-U-Net, (3) DS-V-Net, (4) DS-Att-U-Net, (5) DS-Att-V-Net, (6) DS-cSE-V-Net (the modified V-Net with DS and cSE block), and (7) DS-csSE-V-Net (the modified V-Net with DS and csSE block), respectively. Table 2 lists the Dice and Hausdorff distances of each tumor tissue class of the implemented models. Wt denotes whole tumor, Et represents enhancing tumor while Tc is short for tumor core.

TABLE 2. The quantitative results on BraTS 2017 validation datasets of different models. Wt denotes the whole tumor, Et represents the enhancing tumor while Tc is short for tumor core. The implemented models include: (1) 3D U-Net, (2) DS-U-Net, (3) DS-V-Net, (4) DS-Att-U-Net, (5) DS-Att-V-Net, (6) DScSE-V-Net and (7) DScsSE-V-Net.

Models	Dice			Hausdorff(mm)		
	Et	Wt	Tc	Et	Wt	Tc
(1)	0.7204	0.8799	0.7693	3.6407	16.9156	8.6511
(2)	0.6789	0.8953	0.7828	4.8848	8.9592	11.5662
(3)	0.7343	0.8959	0.7716	5.8534	6.5209	8.12
(4)	0.7362	0.8851	0.7636	4.184	4.9525	8.1045
(5)	0.7149	0.8812	0.7944	6.6581	6.496	7.5223
(6)	0.7474	0.8928	0.8005	4.1977	4.5295	5.5724
(7)	0.7079	0.8901	0.7979	3.5612	5.1424	8.8293

Comparing results of (1) and (2) (The first two rows in Table 2, we observed that the Dices of WT and TC of DS-U-Net increased to 0.8953 and 0.7828 from 0.8799 and 07693 of 3D U-Net, respectively, and the Hausdorff distances of WT was also decreased by adding deep supervision. The improvement of most of the metrics demonstrated that deep supervision was helpful to extract more discriminative features. Comparing results of (2) and (3), we observed that the Dices of all tumor tissue types of DS-V-Net were better than those of DS-U-Net. This indicated that V-Net outperformed U-Net in this task. There were no clear improvements between the results of DS-Att-U-Net and DS-Att-V-Net and the corresponding models without attention gates. The results of DS-cSE-V-Net outperformed DS-V-Net with improvements of almost all metrics (the Dices of ET and TC increase to 0.7474 and 0.8005 from 0.7343 and 07716, and the Hausdorff distance of ET, WT and TC reduce to 4.1977, 4.5295, and 5.5724 from 5.8534, 6.5209 and 8.12, respectively). This demonstrated the effectiveness of the cSE block by global pooling to adaptively enhance salient features and ignore unimportant ones by adjusting the weighting of each feature map of CNN features. There was no much performance gain of the results of DS-csSE-V-Net, perhaps because the large patch size in 3D makes the relative importance of a spatial pixel in the csSE block not notable. One segmentation example of our best model (DS-cSE-V-Net) is shown in Fig.4.

E. COMPARISON WITH STATE-OF-THE-ART ON BRATS 2017 VALIDATION DATASET

Table 3 gives the Dice and Hausdorff distances of each tumor tissue class of our best model and those of the top published methods on the BraTS 2017 validation dataset. The methods in [15], [17] and [11] are the top three methods on test data of BraTS 2017. Kamnitsas' s method [17], EMMA, demonstrated the effectiveness of ensemble of different models. Wang's method [15] took the first place on the BraTS 2017 validation dataset, showing the power of cascaded networks. However, it had to train several networks, and errors produced by the forward network could not be corrected by the following networks. Isensee's method [11] shown a U-net architecture with small changes and more training data could achieve competitive performance. Our model modified the

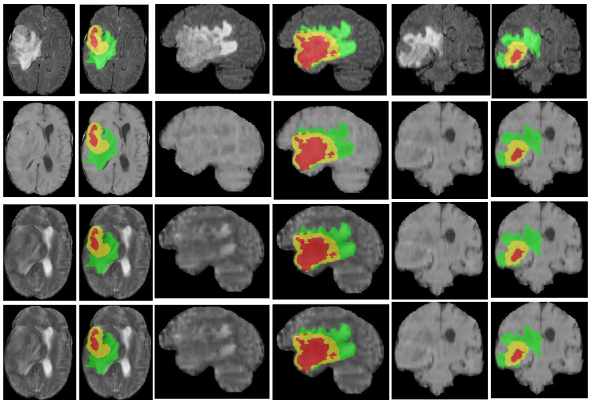

FIGURE 4. A MRIs (FLAIR, T1, T2, T1c) and its segmentation of our top performing model DS-cSE-V-Net in the BraTS validation data-set in three orientations viewed with ITKSNAP [29]. The first, third and fifth column images are slices of the input images, while the second, fourth and last column images are the segmentation overlapped on the corresponding input slice. (Axial:the first two column images, Sagittal: the third and fourth column images, Coronal: the last two column images),and the whole tumor (green, yellow, red), tumor core (yellow and red), enhanced tumor (yellow).

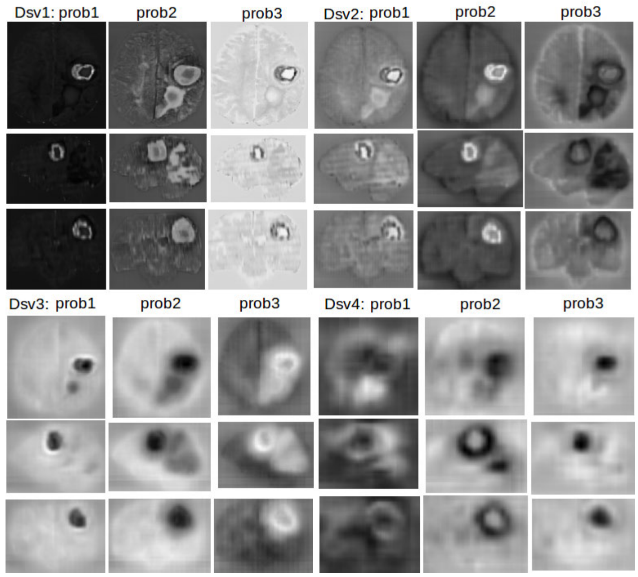

FIGURE 5. Multi-scale output feature maps of each decoder stage of DS-cSE-V-Net in prediction on BraTS validation dataset.

TABLE 3. Comparison of our best model with the top published approaches on BRATS'17 validation dataset. Wt denotes whole tumor, Et represents enhancing tumor while Tc is short for tumor core.

Models	Dice			Hausdorff(mm)		
	Et	Wt	Tc	Et	Wt	Tc
[14]	0.719	0.884	0.771	6.702	10.215	6.202
[17]	0.738	0.901	0.797	4.5	4.23	6.56
[15]	0.786	0.905	0.838	3.28	3.89	6.48
[11]	0.732	0.896	0.797	4.55	6.97	9.48
Ours	0.7474	0.8928	0.8005	4.1977	4.5295	5.5724

standard V-Net, incorporated deep-supervision and Squeeze-and-Excitation modules, achieving comparable results with those of the top published methods. The method in [14] also adopted a SE block for feature recalibration. The results in Table 3 show that our model outperforms [14] in both Dice score and Hausdorff distance. Compared to the full convolutional network (FCN) used in [14], our base network VNet is able to reconstruct more multi-scale features. In addition, our network incorporated deep surpervision in each stage of VNet, obtaining a better performance than the method in [14].

F. DISCUSSIONS

Brain tumors have great tumor intensity changes, variable and irregular tumor shape, size, and localization, and unclear boundaries to normal brain tissues. Most recent semantic segmentation approaches utilized multi-scale feature fusion or encoder-decoder structures to improve segmentation accuracy of multi-scale objects. We presented an encoder-decoder neural network DS-cSE-V-Net with 3D squeeze-and-excitation and deep supervision for automated brain tumor segmentation.

We extracted the output feature maps of each decoder stage (Dsv4, Dsv3, Dsv2, and Dsv1) of DS-cSE-V-Net in prediction on BraTS validation dataset, the same as we did in the deep supervision training process. Since the brain tumor tissues were classified to 3 classes, each decoder stage output had 4 feature maps. Fig. 5 displays the output feature maps of all stages. For clarity, we combined outputs of all the patches of a sample to reconstruct probability maps with the same volume size of the sample, the same as we did in the prediction. Fig. 5 indicated that the encoder-decoder structure with multiple stages did learn multi-scale feature maps. The deep supervised loss calculated from them was helpful to extract more discriminative features.

Besides, the feature maps of each decoder stage composited the feature maps from the next decoder stage and those from the corresponding encoder stage by a concatenation followed by a residual block, a residual block or a bottom residual block and a SE block. The SE block were added both in the encoder stage and the decoder stage to adaptively enhance salient features and ignore unimportant ones by adjusting the weighting of each feature map of CNN features. Fig. 6 shows prediction results with the DS-cSE-V-Net at training epochs. The prediction results got better with the training epoch increasing. Fig.7 shows visual segmentation results of DS-V-Net and DS-cSE-V-Net overlapped on the FLAIR in the BraTS validation dataset in three orientations viewed with ITKSNAP [29]. From Fig.7 we could see, DS-cSE-V-Net outperformed DS-V-Net. The prediction results of DS-cSE-V-Net fitted more to the tumor boundaries, and little false positive tumors were removed in the prediction results of DS-cSE-V-Net.

Although we had achieved promising segmentation performance by our DS-cSE-V-Net, there were still a few bad prediction cases. And our best model didn't outperformed the cascaded method [15] on most metrics. One limitation of our model is the receptive field problem, which is the general

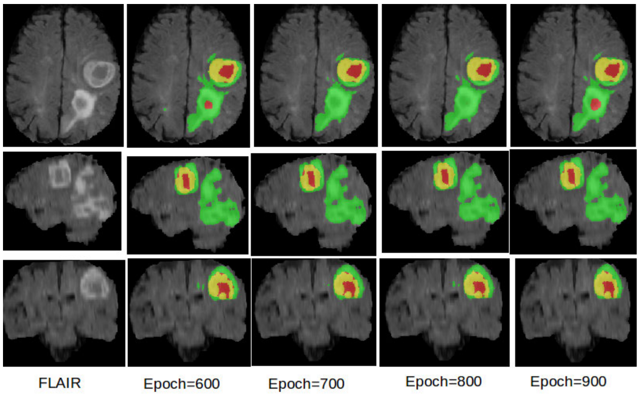

FIGURE 6. Segmenation result of DS-cSE-V-Net at different epochs.

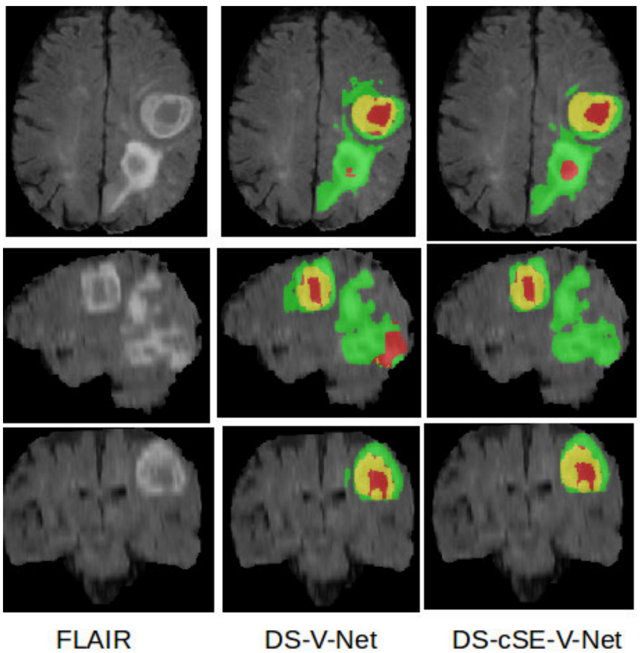

FIGURE 7. A segmentation result of DS-V-Net and DS-cSE-V-Net overlapped on FLAIR in the BraTS validation dataset in three orientations viewed with ITKSNAP [29] (Axial: images in the first row, Sagittal: images the middle row, Coronal: images in the bottom row), Wt (green, yellow, red), Tc (yellow and red), Et (yellow).

drawback of CNN based segmentation. The model lacks large context information due to the limited size of CNN kernels. We will try to incorporate context information in the model. In addition, pixels relationships can be extracted and utilized in the model to obtain more robust results.

V. CONCLUSION

We presented a DSSE-V-Net for automated brain tumor segmentation from multi-model MRIs. We enhanced the original V-Net with a few modifications. The adoption of deep supervision forces the middle stage filters to favor highly discriminative features and accelerates network convergence. The introduction of Squeeze-and-excitation into encoder and decoder stage facilitates the model to focus on more informative features. Experimental results on BraTS 2017 validation dataset showed our model outperformed the traditional encoder-decoder network, and was also highly competitive compared with those methods winning the BraTS 2017 challenge.

REFERENCES

[1] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, and R. Wiest, "The multimodal brain tumor image segmentation benchmark (BRATS)," *IEEE Trans. Med. Imag.*, vol. 34, no. 10, pp. 1993–2024, Oct. 2015.
[2] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle, "Brain tumor segmentation with deep neural networks," *Med. Image Anal.*, vol. 35, pp. 18–31, Jan. 2017.
[3] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 565–571.
[4] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.
[5] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning dense volumetric segmentation from sparse annotation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2016, pp. 424–432.
[6] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
[7] A. G. Roy, N. Navab, and C. Wachinger, "Concurrent spatial and channel 'squeeze & excitation' in fully convolutional networks" in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2018, pp. 421–429.
[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[9] *CBICA Image Processing Portal*. Accessed: Nov. 11, 2018. [Online]. Available: https://ipp.cbica.upenn.edu/
[10] A. Myronenko, "3D MRI brain tumor segmentation using autoencoder regularization," in *Proc. Int. MICCAI Brainlesion Workshop*, 2018, pp. 311–320.
[11] F. Isensee, P. Kickingereder, W. Wick, M. Bendszus, and K. H. Maier-Hein, "Brain tumor segmentation and radiomics survival prediction: Contribution to the BRATS 2017 challenge," in *Proc. Int. MICCAI Brainlesion Workshop*, 2017, pp. 287–297.
[12] R. McKinley, R. Meier, and R. Wiest, "Ensembles of densely-connected CNNs with label-uncertainty for brain tumor segmentation," in *Proc. Int. MICCAI Brainlesion Workshop*, 2018, pp. 456–465.
[13] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.
[14] S. Pereira, V. Alves, and C. A. Silva, "Adaptive feature recombination and recalibration for semantic segmentation: Application to brain tumor segmentation in MRI," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2018, pp. 706–714.
[15] G. Wang, W. Li, S. Ourselin, and T. Vercauteren, "Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks," in *Proc. Int. MICCAI Brainlesion Workshop*, 2017, pp. 178–190.
[16] D. Lachinov, E. Vasiliev, and V. Turlapov, "Glioma segmentation with cascaded Unet," 2018, *arXiv:1810.04008*. [Online]. Available: http://arxiv.org/abs/1810.04008
[17] K. Kamnitsas, W. Bai, E. Ferrante, S. McDonagh, M. Sinclair, N. Pawlowski, M. Rajchl, M. Lee, B. Kainz, D. Rueckert, and B. Glocker, "Ensembles of multiple models and architectures for robust brain tumour segmentation," in *Proc. Int. MICCAI Brainlesion Workshop*, 2017, pp. 450–462.
[18] C. Zhou, S. Chen, C. Ding, and D. Tao, "Learning contextual and attentive information for brain tumor segmentation," in *Proc. Int. MICCAI Brainlesion Workshop*, 2018, pp. 497–507.
[19] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 630–645.
[20] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Artificial Intelligence and Statistics*. 2015, pp. 562–570.
[21] Q. Dou, L. Yu, H. Chen, Y. Jin, X. Yang, J. Qin, and P.-A. Heng, "3D deeply supervised network for automated segmentation of volumetric medical images," *Med. Image Anal.*, vol. 41, pp. 40–54, Oct. 2017.

[22] M. Berman, A. R. Triki, and M. B. Blaschko, "The Lovasz-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4413–4421.
[23] G. Collewet, M. Strzelecki, and F. Mariette, "Influence of MRI acquisition protocols and image intensity normalization methods on texture classification," *Magn. Reson. Imag.*, vol. 22, no. 1, pp. 81–91, Jan. 2004.
[24] [Online]. Available: https://itk.org/
[25] Y. Qin, K. Kamnitsas, S. Ancha, J. Nanavati, G. Cottrell, A. Criminisi, and A. Nori, "Autofocus layer for semantic segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2018, pp. 603–611.
[26] [Online]. Available: https://pytorch.org/
[27] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.
[28] J. Schlemper, O. Oktay, M. Schaap, M. Heinrich, B. Kainz, B. Glocker, and D. Rueckert, "Attention gated networks: Learning to leverage salient regions in medical images," *Med. Image Anal.*, vol. 53, pp. 197–207, Apr. 2019.
[29] P. A. Yushkevich, J. Piven, H. C. Hazlett, R. G. Smith, S. Ho, J. C. Gee, and G. Gerig, "User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability," *NeuroImage*, vol. 31, no. 3, pp. 1116–1128, Jul. 2006.

PING LIU received the B.S. and M.S. degrees in biomedical engineering from Chongqing University, China. She is currently pursuing the Ph.D. degree with Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences, Shenzhen, China. She is currently an Assistant Researcher with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen. Her research interests include medical image analysis and virtual surgical planning.

QI DOU (Member, IEEE) received the bachelor's degree in biomedical engineering from Beihang University, Beijing, China, in 2014, and the Ph.D. degree from the Department of Computer Science and Engineering, The Chinese University of Hong Kong, in 2018. She is currently an Assistant Professor with the Department of Computer Science and Engineering, The Chinese University of Hong Kong. Her research interests are in the interdisciplinary fields of medical image analysis and artificial intelligence, for improving lesion detection, anatomical structure computation, and surgical robotics perception, with an impact to advance disease diagnosis and robot-assisted intervention via machine intelligence.

QIONG WANG is currently an Associate Researcher with the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences. Her research interests include virtual reality applications in medicine, visualization, medical imaging, human–computer interaction, and computer graphics.

PHENG-ANN HENG (Senior Member, IEEE) received the Ph.D. degree in computer science from Indiana University, Indianapolis, IN, USA. He is currently a Professor with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, where he is also the Director of the Virtual Reality, Visualization, and Imaging Research Centre. He is also the Director of the Research Center for Human–Computer Interaction, Shenzhen Institute of Advanced Integration Technology, Chinese Academy of Sciences, Shenzhen, China. His research interests include virtual reality applications in medicine, visualization, medical imaging, human–computer interfaces, rendering and modeling, interactive graphics, and animation.