A Computation-Efficient CNN System for High-Quality Brain Tumor Segmentation

Yanming Sun

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montreal, Quebec, Canada

August 2020

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Yanming Sun

Entitled: A Computation-Efficient CNN System for High-Quality Brain Tumor Segmentation

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Electrical and Computer Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

____ Chair
Dr. W.-P. Zhu

____ External Examiner
Dr. A. Ben Hamza (CIISE)

____ Internal Examiner
Dr. W.-P. Zhu

____ Supervisor
Dr. C. Wang

Approved by: ___________________________________
Dr. Y.R. Shayan, Chair
Department of Electrical and Computer Engineering

_______________ 20___

____
Dr. Mourad Debbabi, Interim Dean,
Gina Cody School of Engineering and Computer Science

Abstract

A Computation-Efficient CNN System for High-Quality Brain Tumor Segmentation

Yanming Sun

Brain tumor diagnosis is an important issue in health care. Automated brain tumor segmentation can help timely diagnosis. It is, however, very challenging to achieve high-quality segmentation results, because the shapes, sizes, textures and locations of brain tumors vary from patient to patient. To develop a Convolutional Neural Network (CNN) system for a high-quality brain tumor segmentation at the lowest computation cost, the CNN should be custom-designed to extract efficiently sufficient critical features particularly related to the tumors from brain images for the multi-class segmentation of tumor areas.

In this thesis, a CNN system is proposed for brain tumor segmentation. The system consists of three parts, a pre-processing block to reduce the data volume, an application-specific CNN (ASCNN) to segment tumor areas precisely, and a refinement block to detect false positive voxels. The CNN, designed specifically for the task, has 7 convolution layers, and the number of output channels per layer is no more than 16. The convolutions combined with max-pooling in the first half of the CNN are performed to localize brain tumor areas. Two convolution modes, namely depthwise convolution and standard convolution, are performed in parallel in the first 2 layers to extract elementary features efficiently. In the second half of the CNN, the convolutions combined with upsampling are to segment different tumor areas. For a fine classification of pixel-wise precision, the feature maps are modulated by adding the weighted local feature maps generated in the first half of the CNN. The system has only 11716 parameters to be trained and, for a patient case of (240x240x155 x3) voxels, it requires only 21.14G Flops to complete the test. Hence, it is likely the simplest CNN system, so far reported, for brain tumor segmentation.

The performance of the proposed system has been evaluated by means of CBICA Image Processing Portal with samples from dataset BRATS2018. Requiring a very low computation volume, the proposed system delivers a high segmentation quality indicated by its average Dice scores of 0.75, 0.88 and 0.76 for enhancing tumor, whole tumor and tumor core, respectively, and the median Dice scores of 0.85, 0.92, and 0.86. Its processing quality is comparable to the best ones so far reported. The consistency in system performance has also been measured, and the results have demonstrated that the system is able to reproduce almost the same output to the same input after retraining.

In conclusion, the proposed CNN system has been designed to meet the specific needs to segment brain tumors or other kinds of tumors in medical images. In this way, the redundancy in computation can be minimized, the information density in data flow increased, and the computation efficiency/quality improved. This design demonstrates that a CNN system can be made to perform a high-quality processing, at a very low computation cost, for a specific application. Hence, ASCNN is an effective approach to lower the barrier of computation resource requirement of CNN systems in order to make them more implementable and applicable for general public.

Acknowledgements

I would like to express my best appreciation to my supervisor Dr. Chunyan Wang for her support during my study in Concordia University. She always gives me brilliant guidance. Moreover, her focus on details in research influences me deeply. It is my great honor and fortune to work under her supervision.

I would also like to thank my colleagues, Yunlong Ma, Bao Zhu, and Mingze Ni, for the help in the last two years.

Additionally, thank Compute Canada and Calcul Quebec for the powerful servers.

Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

Adam	Adaptive moment estimation
ASCNN	Application-Specific Convolutional Neural Network
BN	Batch Normalization
BRATS	The Multimodal Brain Tumor Image Segmentation Benchmark
CBICA	Center for Biomedical Image Computing and Analytics
CNN	Convolutional Neural Network
ET	Enhancing tumor
FCN	Fully Convolutional Network
FDR	False Discovery Rate
FLAIR	Fluid-attenuated inversion recovery
FLOPs	Floating-Point Operations
FNR	False Negative Rate
HGG	High-grade gliomas
LGG	Low-grade gliomas
MRI	Magnetic Resonance Imaging
ReLU	Rectified Linear Unit
SGD	Stochastic gradient descent
T1	T1-weighted imaging
T1c	T1-weighted contrast-enhanced imaging
T2	T2-weighted imaging
TC	Tumor core
VGG	Visual Geometry Group
WT	Whole tumor

List of Symbols

N_{sl}	The number of slices per 3D images.
N_{vET}	The number of voxels in ET areas
P_0	Number of voxels in the tumor-free regions in a predicted image
P_1	Number of voxels in the tumor regions in a predicted image
$StdDev$	Standard deviation
T_0	Number of voxels in the tumor-free regions in the ground truth
T_1	Number of voxels in the tumor regions in the ground truth
$w(k_1, k_2)$	2D kernel
$x_{00}, x_{01}, \ldots, x_{33}$	Input of pooling operation
x_1	The distance between the left boundary of a slice and the leftmost point of the brain area.
x_2	The distance between the right boundary of a slice and the rightmost point of the brain area.
$x(i, j)$	Input image of 2D convolution
$y_{00}, y_{01}, y_{10}, y_{11}$	Output of pooling operation
y_1	The distance between the upper boundary of a slice and the uppermost point of the brain area.
y_2	The distance between the lower boundary of a slice and the lowermost point of the brain area.
y_i	The predicted probability
$y(i, j)$	Output image of 2D convolution
y_t	The label of the ground truth
x_{1m}, x_{2m}, y_{1m} and y_{2m}	A fixed set of x_1, x_2, y_1 and y_2.
α	A small positive slope in the negative area of Leaky-ReLU

1. Introduction

1.1 Background and Challenges

Brain tumors pose a serious problem to human health, and brain tumor segmentation is a critical step for the diagnosis and treatment of the disease. Manual segmentation is very time-consuming and often causes delays. It is thus important to develop fully automated computer vision systems for brain tumor segmentation to facilitate timely diagnosis. This development is, however, a very challenging task.

Human brain is likely the most complex organ in human body. From anatomy point of view, it is composed of many interconnected areas, each of which has different structures and specialized for different functions, as illustrated in Figure 1.1. Brain tumors can have many different types, growing in different parts of a brain. Each type of tumors can have different appearances in different stages of its development.

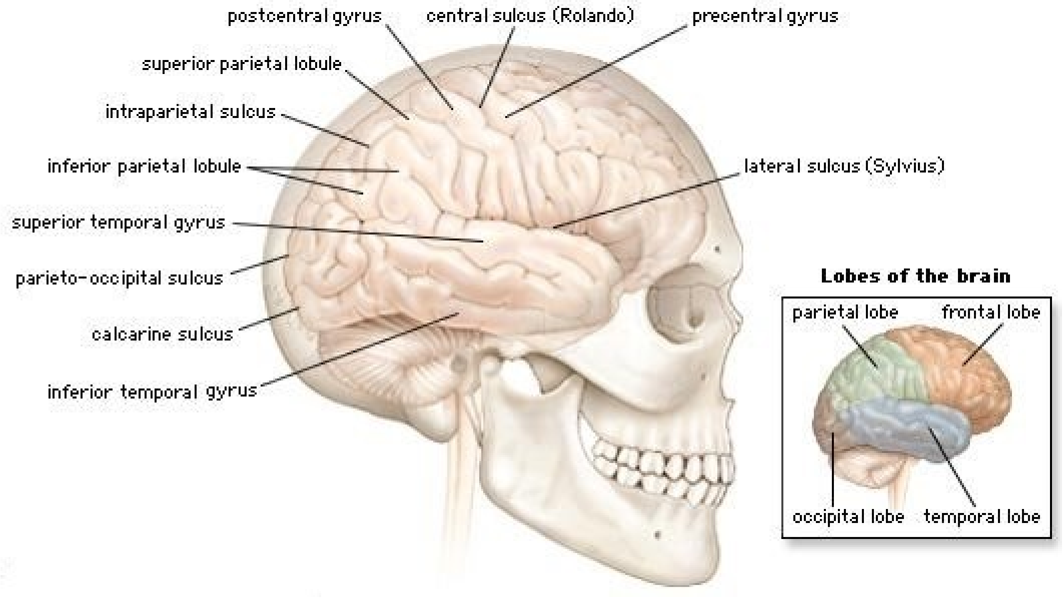
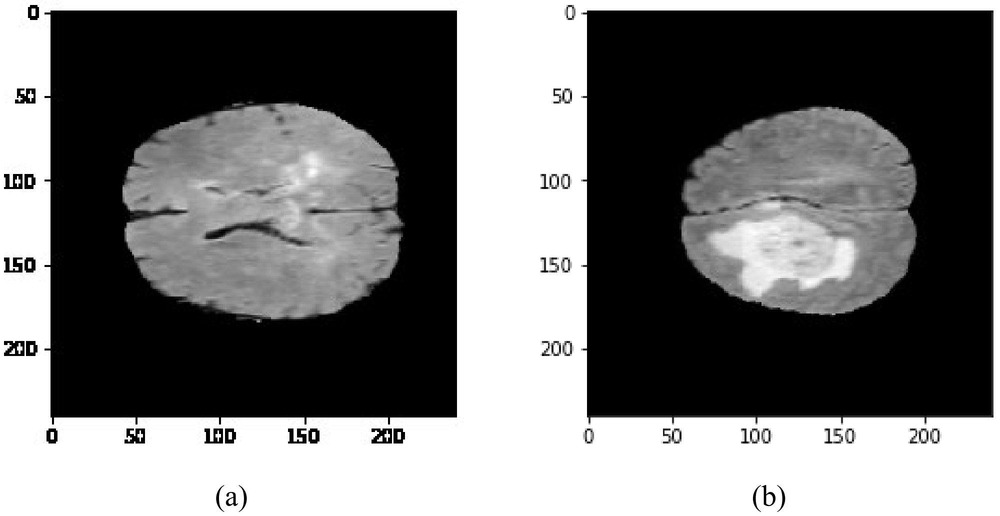


Figure 1.1 Complicated structure of human brain [1].

From computer vision point of view, human brain is a complicated 3D structure and is often represented as a large number of slices from different cross-sections. Voxels in the 3D structure are represented by the pixels in the 2D slices. The structures of tissues in slices are interpreted as different textures, as examples illustrated in Figure 1.2.



(a) (b)

Figure 1.2 T2 slices from two patient cases. (a) HGG/Brats18_2013_2_1, No. 91st slice. (b) HGG/Brats18_2013_7_1, No. 101st slice.

A computer vision program for brain tumor segmentation needs to be able to distinguish tumor tissue from healthy tissue. In practice, tumor tissues in one location can have similar texture to that of healthy tissues in another location, whereas two tumor areas in different locations may have very different patterns, as illustrated in Figure 1.3. One cannot identify tumor areas by applying a simple pattern detection method. Moreover, for meaningful applications in brain tumor diagnosis, the computer vision program needs to define whole-tumor regions and to classify the voxels in these regions into multiple classes, corresponding to three types of intra-tumoral structures, namely edema, non-enhancing (solid) core/necrotic (or fluid-filled) core and enhancing core [2]. This multi-class classification in voxel-wise precision makes the development of a fully automated systems for brain tumor segmentation even more challenging.

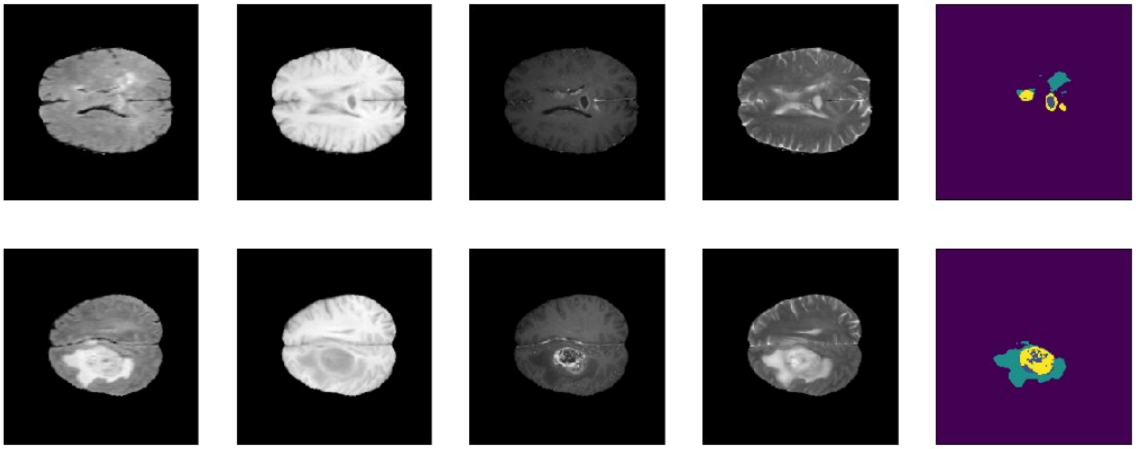


Figure 1.3 Brain images and ground truth of tumor from two patient cases. The images in the first row are FLAIR, T1, T1c, T2 and ground truth of patient Brats18_2013_2_1; The images in the second row are those of patient Brats18_2013_7_1.

1.2 Motivation and Objective

Like most of object detection or recognition programs, a computer vision program for brain tumor segmentation involves 2 basic functions of image processing, feature extraction and classification. The feature extraction is often based on 2D filtering, and the critical issue is how to catch effectively all the image features useful for this specific classification, while these features are often mixed up with everything else. The quality of the feature extraction can have a critical effect on the quality of the classification, though the latter is also related to other elements in the system design.

Brain tumor segmentation can be done by means of filtering and/or morphological operations. High-pass and/or band-pass filters can be applied to extra features. The segmentation method reported in [3] is based on feature extracted by applying Sobel high-pass filtering method. Tumor features can also be extracted by applying Gabor filters, and the data are used to classify the pixels with extremely randomized trees [4]. As healthy brains have certain degree of symmetry [5], brain tumor detection can be done by analysing the symmetry of MRI brain images [6]. Implementations of these methods do not require a large number of computation resources. However, due to the complexity of the brain tumor variations, it is difficult to precisely classify the pixels of intra-tumoral structures.

Convolutional Neural Networks (CNNs) are widely used for feature extraction and classification in various data processing applications, including those for medical image segmentations. A large number of research papers are found in literature about CNN systems for brain tumor segmentation. The processing quality reported in most of the papers is superior, with respect to that achieved by conventional filtering methods, however, CNN systems require much more computation resources. Most of them have millions, if not tens of millions, of parameters. Considering the large amount of data in a 3D input image, a huge volume of computation is required for such a system to perform a test of a single case, and its training process is even much more computation consuming. Hence, to train and to use the systems, one needs a powerful computing facility, which can limit the implementation and applications of the CNN systems for brain tumor segmentation.

The objective of the work presented in this thesis is to design a fully automated computer vision system for high-quality brain tumor segmentation with low computation requirement in order to facilitate its implementation. The system involves a CNN block for feature extraction and classification. The design of the CNN aims at a simplicity in structure and efficiency in its operations. To this end, the CNN system should be an Application Specific CNN (AS-CNN), i.e., custom-designed for specific applications, instead of general-purpose.

1.3 Scope and Organization

To extract various features from the input images and precisely classify the pixels/voxels for a specific application, without requiring a large amount of computation, the CNN should be custom-designed to target the specific features in the input data and to generate the output data useful in the application. The CNN system to be proposed in this thesis will consist of 3 blocks.

- Pre-processing block for data reduction so that the data processed in the succeeding CNN block will have a much smaller amount and a higher density of feature information with respect to the original input data.
- CNN for feature extraction and classification.
- Post-processing block for refinement.

The pre and post processing blocks will be designed aiming at reducing the computation burden of the CNN so that it can be made to perform the feature extraction and classification more effectively and efficiently.

The thesis is organized as follow. Description of CNN basics and a brief review of the existing systems for brain tumor segmentation are found in Chapter 2. The proposed CNN system is presented in Chapter 3 with detailed description of the design of each functional block. Chapter 4 is dedicated to a comprehensive performance evaluation of the proposed system. The training process and the test results are presented. The performance of the system is compared to those of the CNNs reported in recent years, which is also found in Chapter 4. Chapter 5 is a conclusion of the work presented in this thesis.

2. Background and Relevant Work

2.1 Introduction

Since brain tumor segmentation is very challenging, many efforts are devoted to this task. A large number of methods for this task have been developed, include filtering systems/morphological operations and Convolutional neural networks. In general, Filtering systems/morphological operations are very computation-efficient, but it is difficult to detect a variety of features in brain images. On the contrary, CNNs require huge amount of computation volume but have high potential to handle the problem if there are adequate training samples. Thus, the work in this thesis is focused on computation-efficient CNN system.

In this chapter, the fundamentals of CNN are shown in sub-chapter 2.2. The CNNs for brain tumor segmentation reported recent years are given in sub-chapter 2.3. A brief summary is presented in sub-chapter 2.4.

2.2 Fundamentals of CNN

A CNN is mainly composed of convolution operations, combined with other critical elements, e.g., normalization, non-linear activation function and pooling. Different from traditional convolution operations, the kernels of the convolutions in a CNN are not deterministic. They shall be trained by using a large number of samples.

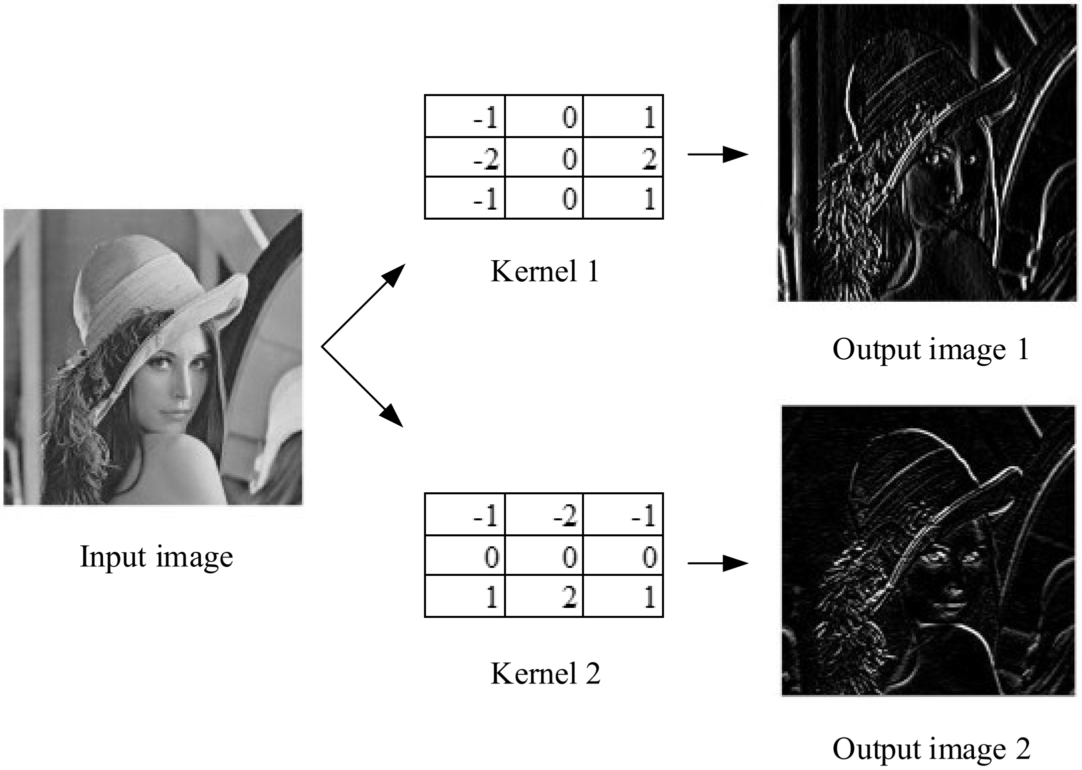
2.2.1 2D Convolution and Convolution Modes

A 2D convolution is in fact a filtering operation, it is defined as follow:

$$y(i,j) = \sum_{k_1=-n}^{n} \sum_{k_2=-m}^{m} w(k_1,k_2)x(i-k_1,j-k_2) \qquad (2.1)$$

where w is a 2D kernel; x and y are input and output and their dimensions are ($2n$+1, $2m$+1).

There are three types of 2D filter, high-pass filter, low-pass filter and band-pass filter. An example of a high-pass filter (Sobel filter) is shown in Figure 2.1.



-1	0	1
-2	0	2
-1	0	1

-1	-2	-1
0	0	0
1	2	1

Figure 2.1 An example of 2D convolutions.

Deterministic kernels can extract specific features in the input images, e.g., the two kernels in Figure 2.1 are to extract vertical edges and horizontal edges of the input image, respectively. It is not easy to design a large number of different deterministic kernels to extract variant features in brain images. However, a CNN has multi-layer convolutions, and there are multiple channels in each layer. More importantly, unlike the deterministic convolutions in Figure 2.1, the kernels in the CNN are trainable, and they can be determined by a training process. Thus, CNN is a more feasible choice for brain tumor segmentation.

As mention above, the convolution in CNN has multiple channels in each layer. There are three types of convolution modes for the channels in a layer. Assume the kernel size is 3×3, and the numbers of input and output channels are 16 and 32, respectively:

(1) Standard convolution. In this mode, each of the output channels is produced by all of the input channels, so the number of weights in this layer is 4608 (3×3×16×32).

(2) Group convolution. Assume the input channels are divided into 4 groups, then a group convolution is similar to 4 small-scale standard convolutions. The number of weights in this layer is 1152 ($3 \times 3 \times \frac{16}{4} \times \frac{32}{4} \times 4$).

(3) Depthwise convolution. In this mode, each of the output channels is produced by only one of the input channels. Since the number of the output channels is twice that of the input channels, an input channel should produce two output channels. The number of weights in this layer is 288 (3×3×32).

2.2.2 Critical Components for CNN

Except the convolutions, there are many other components in CNN architecture.

(1) **Normalization**. A normalization is to uniform the range and the distribution of the data in each layer, thereby facilitating the convergence of a network.

(i) Batch normalization (BN) [7]. A BN normalizes the feature maps by using the mean and standard deviation of a whole batch of data.

(ii) Channel-wise normalization. The data of each channel are normalized with the mean and the standard deviation of the channel.

(2) **Non-linear activation function**. The relationship between the input and the output in image segmentation is non-linear. However, the convolution is a linear function. Thus, it is necessary to introduce non-linearity to help a network learn complex data and to provide accurate predictions.

(i) Rectified Linear Unit (ReLU) [8]. ReLU is a non-linear activation function defined as $f(x) = max\ (0, x)$. It is the most computation-efficient activation function so far. The curve of ReLU is shown in Figure 2.2. It should be noted that only positive values can be perform backpropagation.

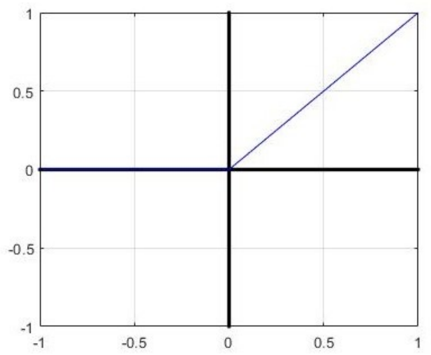


Figure 2.2 Curve of ReLU.

(ii) Leaky ReLU [9]. Leaky ReLU is defined as $f(x) = max\ (\alpha x, x)$, where α is a constant and $0 < \alpha < 1$. It has a small positive slope α in the negative area, so it enables backpropagation for both positive values and negative values, as shown in Figure 2.3.

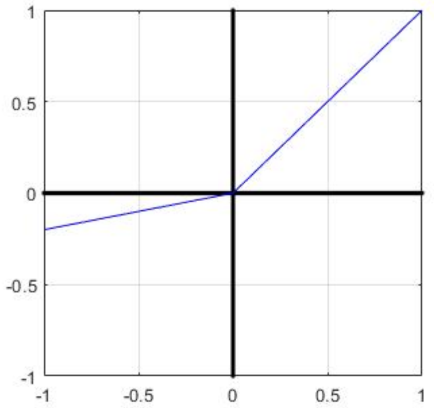


Figure 2.3 An example curve of Leaky ReLU, $\alpha = 0.2$.

(iii) Sigmoid [10]. Sigmoid function is also called Logistic Activation Function. It is defined as $f(x) = \frac{1}{(1+e^{-x})}$. The curve of Sigmoid is shown in Figure 2.4. The range of the values of sigmoid function is from 0 to 1. It leads to a vanishing gradient problem when the input data has very high absolute value.

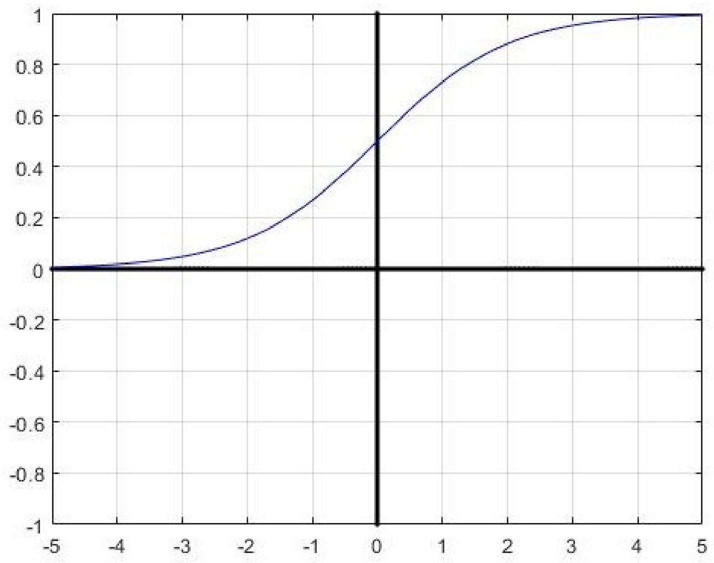


Figure 2.4 Curve of Sigmoid function.

(3) **Pooling**. A pooling operation is in fact a downsampling. It is to increase signal density, to reduce data volume and to decrease computation complexity. Max pooling and average pooling are widely used in practical.

(i) Max pooling. Assume the stride is equaled to 2, a maximum value of four adjacent pixels will be reserved, and the other three pixels will be removed, so the number of pixels of the output is a quarter of that of the input. As shown in Figure 2.5, the output is $y_{00} = max\ (x_{00}, x_{01}, x_{10}, x_{11})$, …, $y_{11} = max\ (x_{22}, x_{23}, x_{32}, x_{33})$.

(ii) Average pooing. An average pooling means that an average value of four adjacent pixels will be reserved. As showed in Figure 2.5, the output is $y_{00} = (x_{00} + x_{01} + x_{10} + x_{11})/4$, …, $y_{11} = (x_{22} + x_{23} + x_{32} + x_{33})/4$.

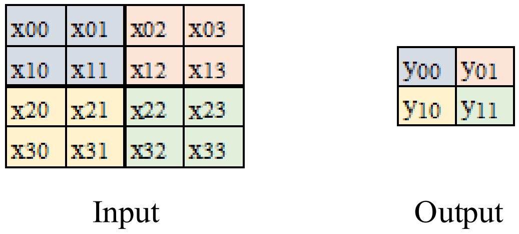


Figure 2.5 Input and Output of pooling, stride 2.

(4) **Upsampling**. The size of the output image of a CNN for image segmentation is the same as that of the input image. After pooling the input data, upsampling operations should be used to recover the dimension of the data. There are two widely used upsampling methods, nearest-neighbor interpolation and bilinear interpolation.

2.2.3 Basic Blocks for CNN

In this sub-chapter, serval widely used blocks of CNN are given.

(1) **VGG** [11]. VGG is proposed by Visual Geometry Group at University of Oxford. There are two versions of VGG architecture. The first one is VGG16, i.e., there are 13 convolution layers and 3 fully connected layers, as shown in Figure 2.6. The other one is VGG19, which has 19 layers. All the convolution layers in VGG blocks are standard convolution with 3×3 kernel. In addition, ReLU and max pooling are applied to the blocks. The VGG block is widely used in image segmentation, for example, SegNet [12]. It has a so-call encoder-decoder architecture, which is shown in Figure 2.7. The encoder is in fact a VGG16 without fully connected layer. The decoder is a flipped VGG16 without fully connected layer.

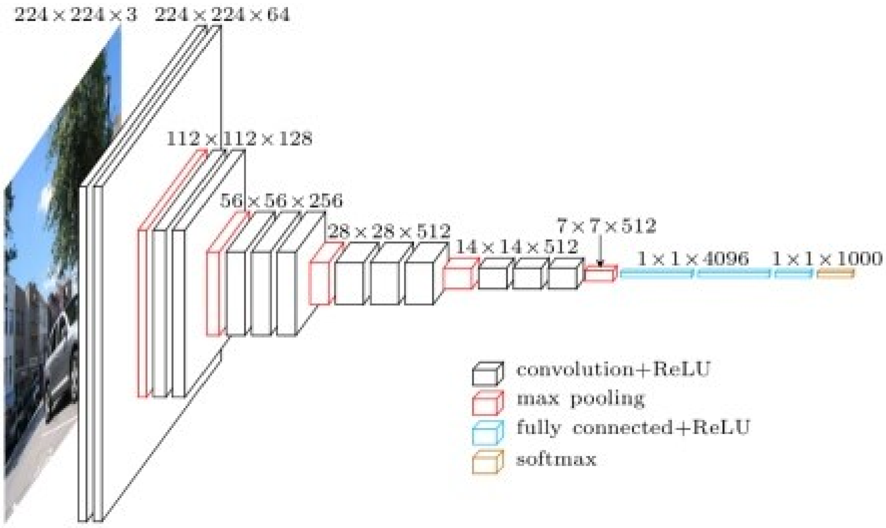


Figure 2.6 Architecture of VGG [11].

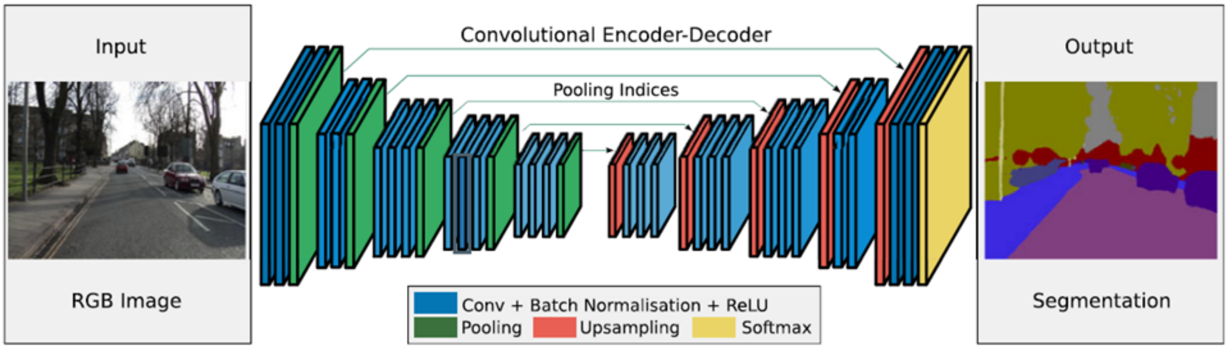


Figure 2.7 Architecture of SegNet [12].

(2) Inception block [13][14]. There are three kernels with different sizes performed in parallel. Large kernel is to extract global features, and small kernel is to extract local features.

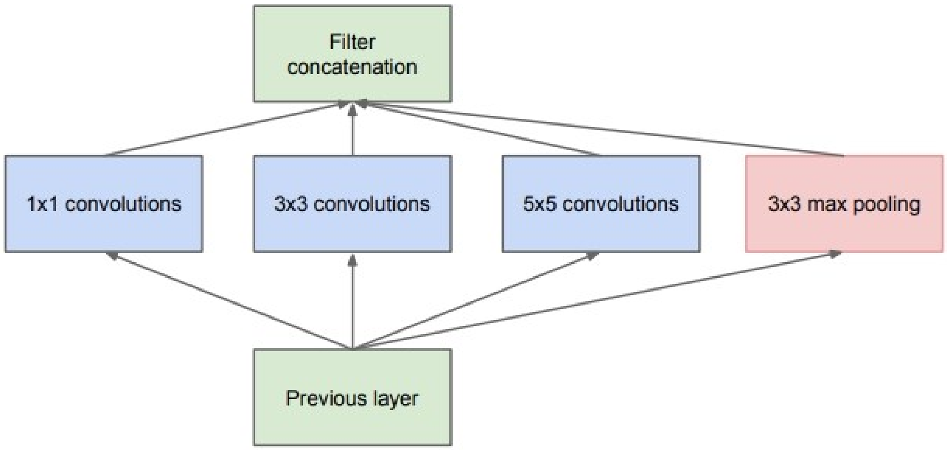


Figure 2.8 Architecture of inception block [13].

(3) Deep residual block [15]. Very deep neural networks are difficult to train and will cause low predicted accuracy. Deep residual block has shortcut connections to enable a network to be deeper. Its architecture is given in Figure 2.9.

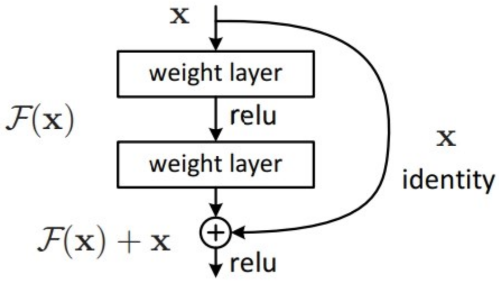

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 2.9 Architecture of deep residual block [15].

(4) Densely connected block [16]. The architecture of densely connected block is shown in Figure 2.10, the feature-maps of all preceding layers are used as inputs of all subsequent layers, so that the connection between layers is very dense. Thus, the network can be deeper.

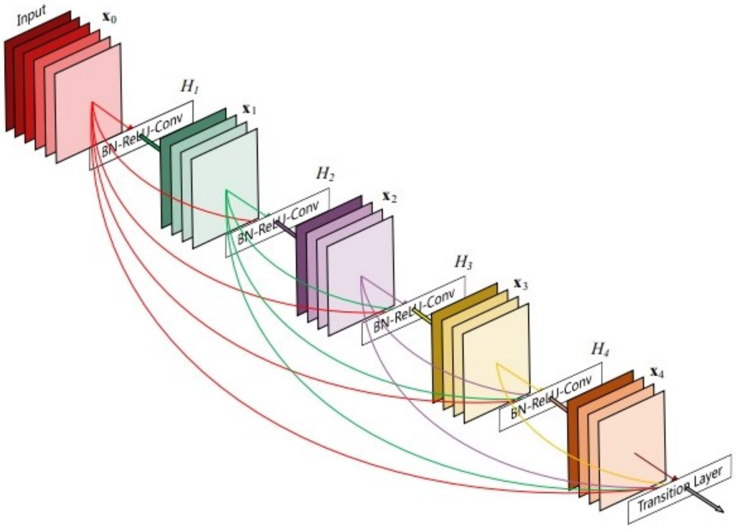


Figure 2.10 Architecture of the densely connected block [16].

2.2.4 Training Process and Important Setups

After building a CNN architecture, the trainable parameters should be determined by training process. The training process includes forward propagation and backward propagation. (i) Forward propagation means that the CNN produces a predicted result based on the input data. (ii) Backward propagation means that a loss is calculated based on the predicted results and the ground truth, and the trainable parameters in the CNN are updated according to the loss.

There are many critical setups for training processing.

(1) Batch size. A batch of input images should be plentiful enough to contain variance features. However, the upper limit of the batch size is related to the volume of a dataset. In other word, the dataset should be divided into adequate batches to let the network update step by step.

(2) Loss function. Cross-Entropy loss is widely used in CNN [17]. It is defined as

$$Loss = \sum -y_t log\ (y_i) \tag{2.2}$$

where y_t is the label of the ground truth, and y_i is the predicted probability.

(3) The number of training epochs. The training processing cannot be stopped before the loss curve converges.

(4) Learning rate. Learning rate should be large in the initial stages to adjust the network coarsely, and it should be small in the last stages to finely adjust the network.

(5) Data augmentation [18]. To train a CNN meaningfully, the training set should contain adequate features. However, the number of samples in training set is limited in most datasets. Thus, data augmentation can be performed to increase the number of samples for training. It includes flipping and rotating the input images.

(6) Optimizer. Stochastic Gradient Descent (SGD) [19] and Adaptive Moment Estimation (Adam) [20] are widely used in CNN. Adam is computation-efficient and is suitable for large network and huge data volume.

2.3 CNNs for Brain Tumor Segmentation

A lot of CNN systems for brain tumor segmentation have been reported. Many of them are designed based on the famous U-net.

2.3.1 U-Net Based CNNs for Brain Tumor Segmentation

U-net, designed based on FCN [21], has been widely used in medical image processing, including brain tumor segmentation [22]. The architecture of U-net is shown in Figure 2.11. U-net consists of a VGG-like contracting path, and an expansive path, in which copies of the feature data generated in the contracting path are included, by concatenation, in the convolutions.

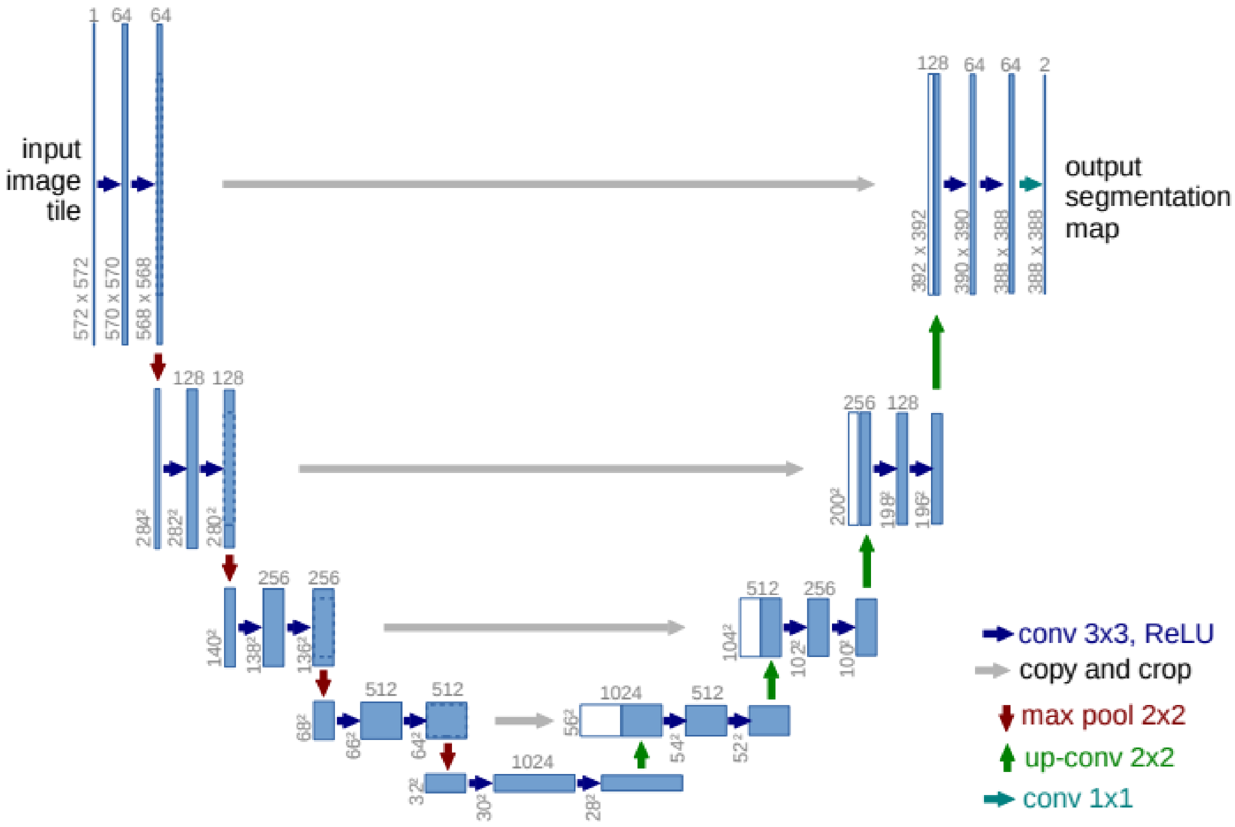


Figure 2.11 Architecture of U-net [22].

A large number of U-net-based CNN systems for brain tumor segmentation have been reported in recent years, attempting to improve the segmentation quality.

In the CNN presented in [23], standard convolution blocks in the original U-net are replaced by inception blocks. Also, up skip connection is used to enhance network connectivity. The details of the CNN are shown in Figure 2.12.

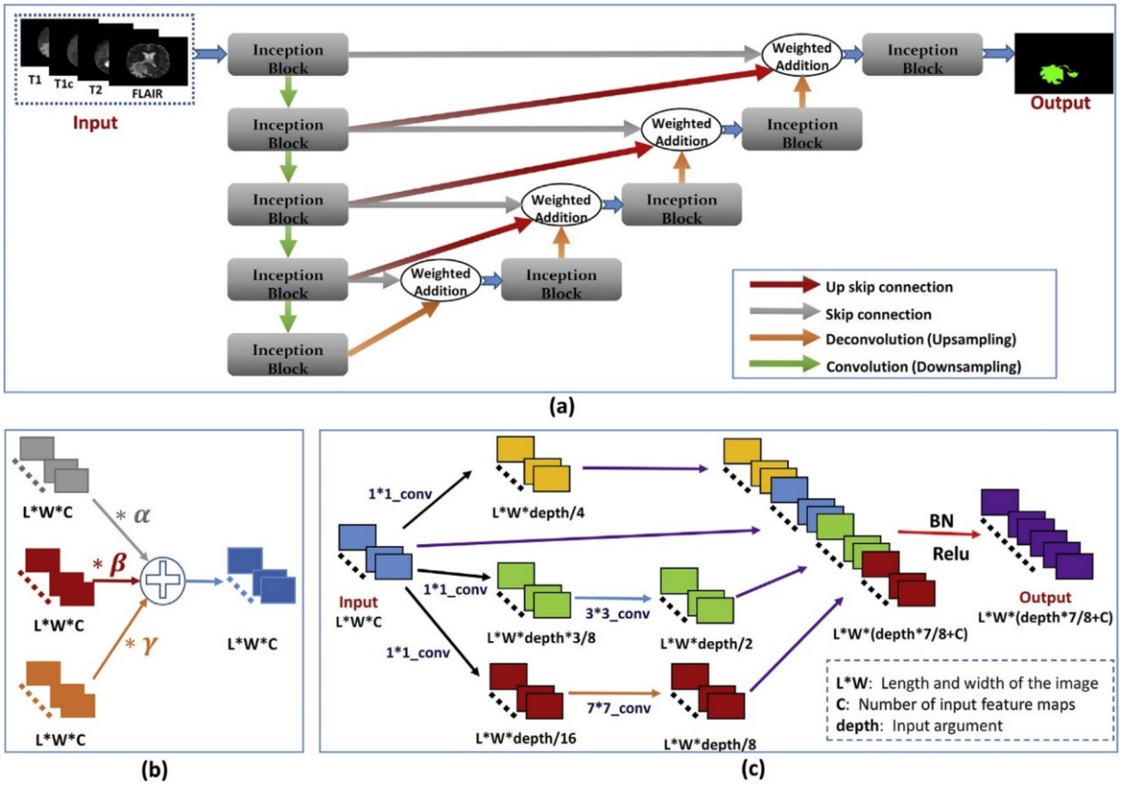


Figure 2.12 Details of the CNN in [23]. (a) Basic architecture. (b) Details of the weighted addition. (c) Details of the inception block.

U-net structure can be also used as a basic block, and multiple blocks are stacked to seek a better processing quality [24]. The architecture of this network is given in Figure 2.13. These basic blocks are simplified from U-net, and the so call ‘bridge connections’ are used to connect these blocks.

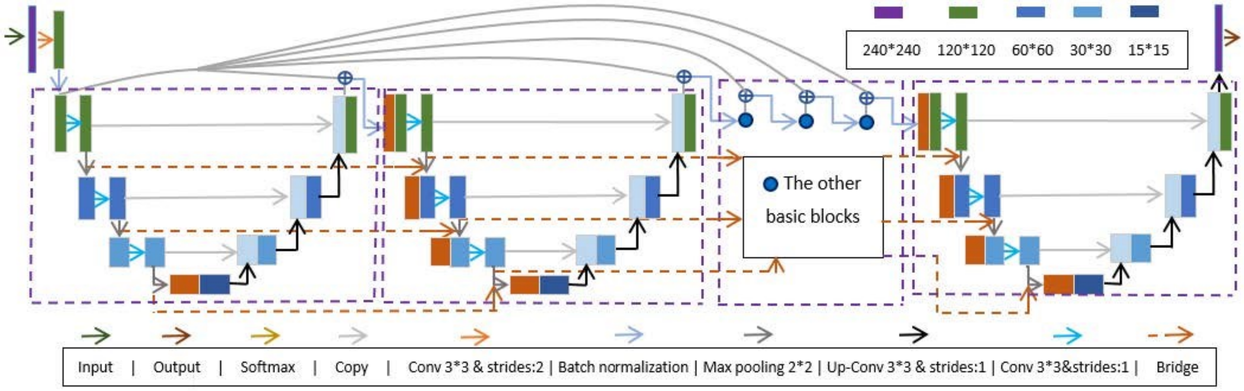
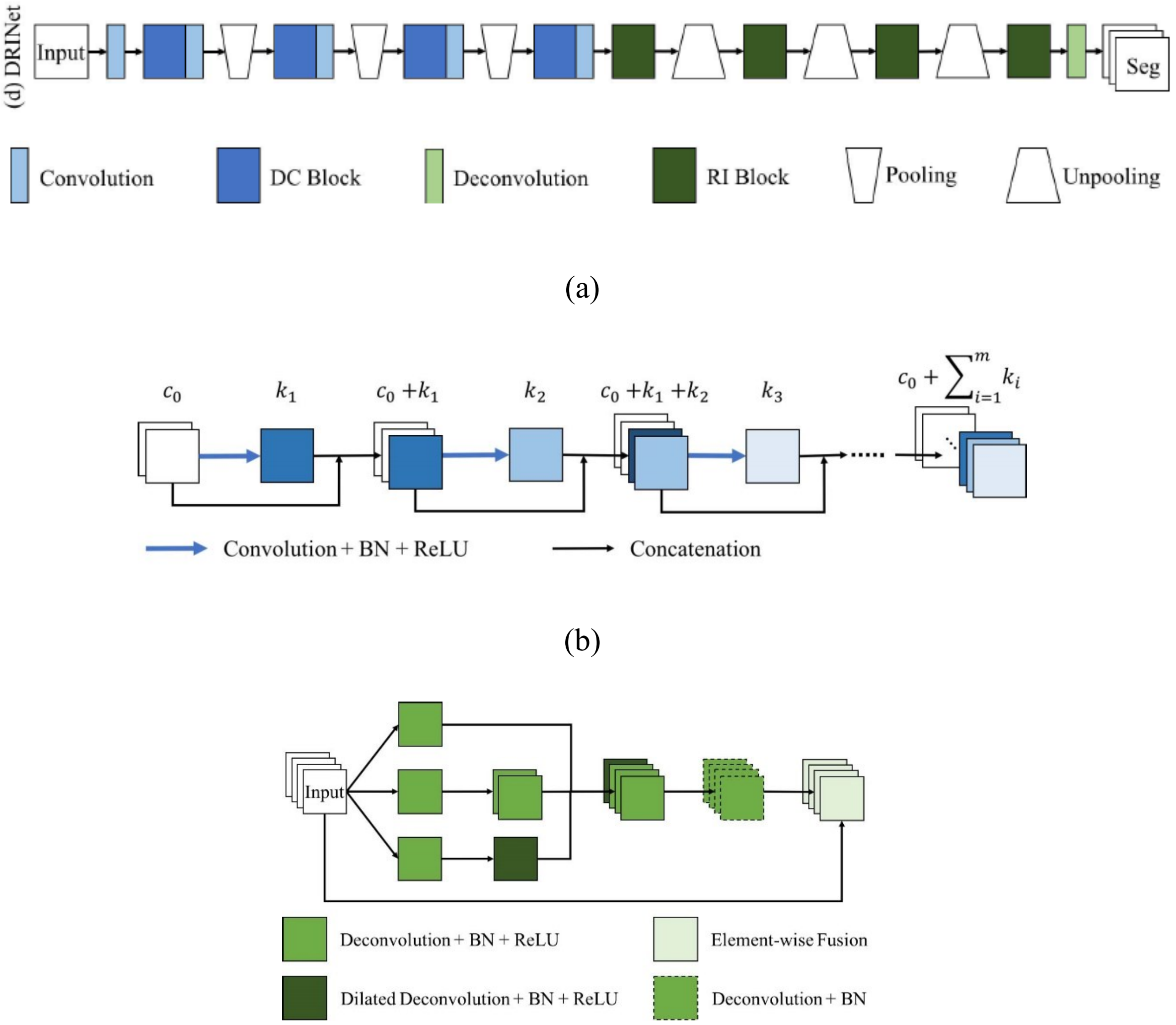


Figure 2.13 Stacked U-net [24].

In DRINet [25], which details are shown in Figure 2.14, standard convolution blocks in the first layers are replaced by dense connection blocks, and those in the last layers are replaced by residual inception blocks, whereas the copy/crop part (also called long skip connection) is not used.

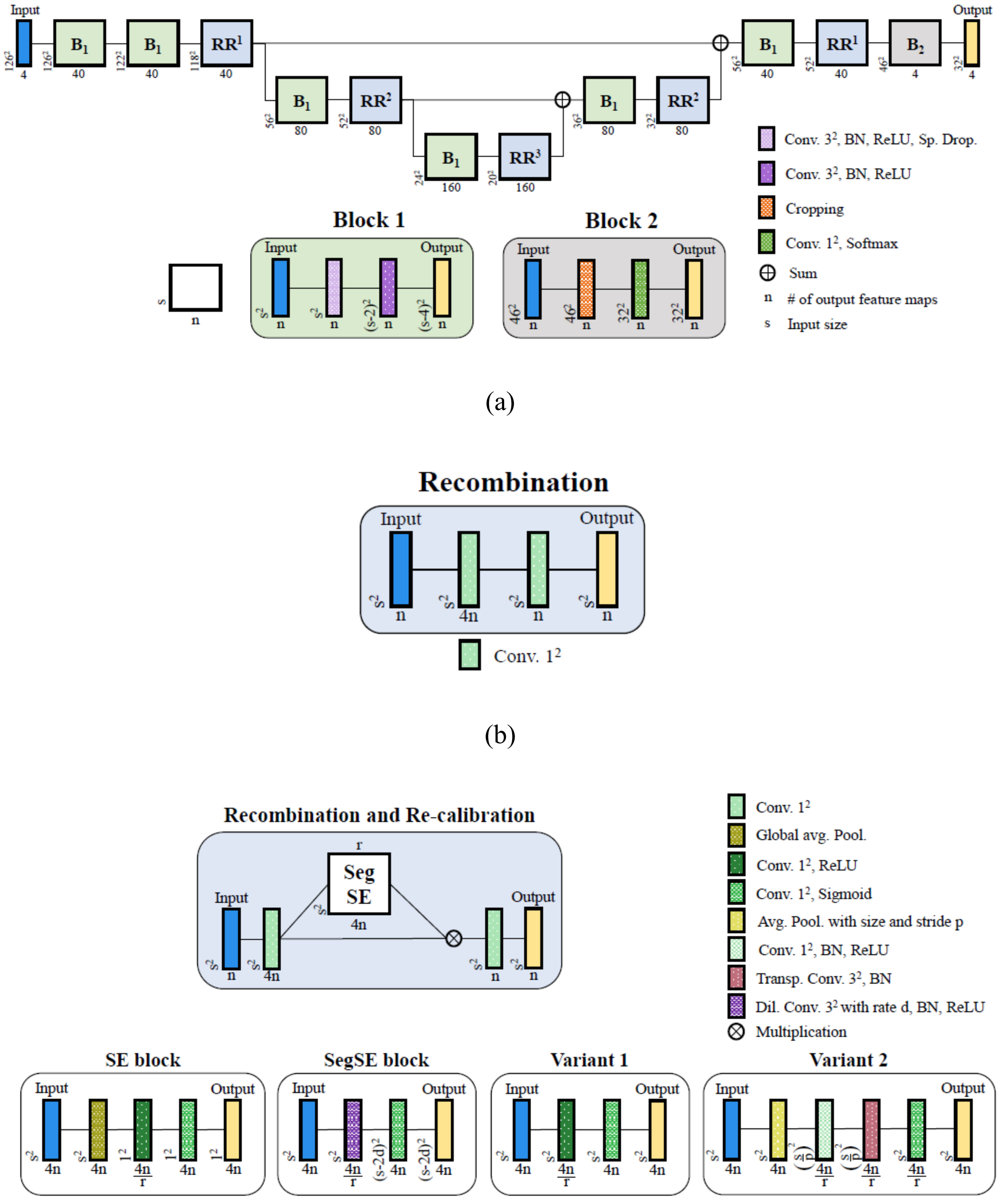


(a)

(b)

Figure 2.14 Details of DRINet [25]. (a) Basic architecture. (b) Dense connection block. (c) Residual inception block.

The network presented in [26] was designed to have recombination of features and spatially adaptive recalibration block to improve the segmentation quality. The details of the CNN are presented in Figure 2.15.



(a)

(b)

(c)

Figure 2.15 Details of the CNN in [26]. (a) Basic architecture. (b) Recombination block. (c) Recombination and Recalibration block.

2.3.2 Other Forms of CNN for Brain Tumor Segmentation

Except the U-net-based CNNs, other forms of CNN can also be adopted for brain tumor segmentation.

In the network reported in [27], the structure of deep residual network is used and combines with spatial fusion blocks. Also, the middle supervision block is applied at the last stage. The architecture is shown in Figure 2.16.

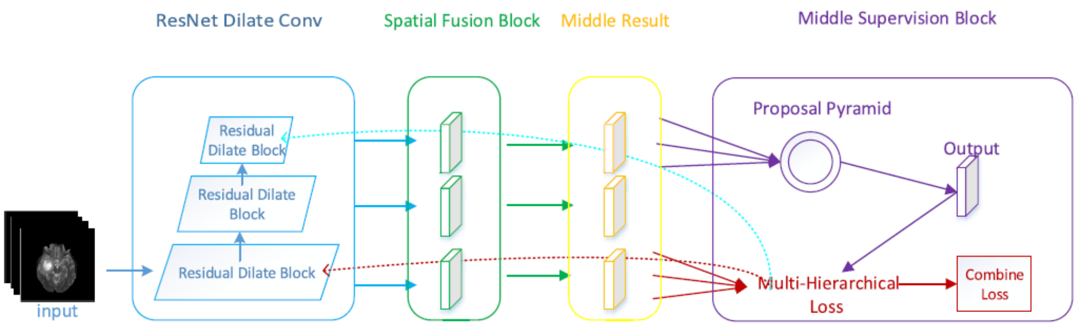


Figure 2.16. Architecture of the network presented in [27].

The network in [28] is an architecture of multiple convolution pathways, which is presented in Figure 2.17. It is made to operate with patches obtained from axial, coronal and sagittal views of the 3D MRI brain images.

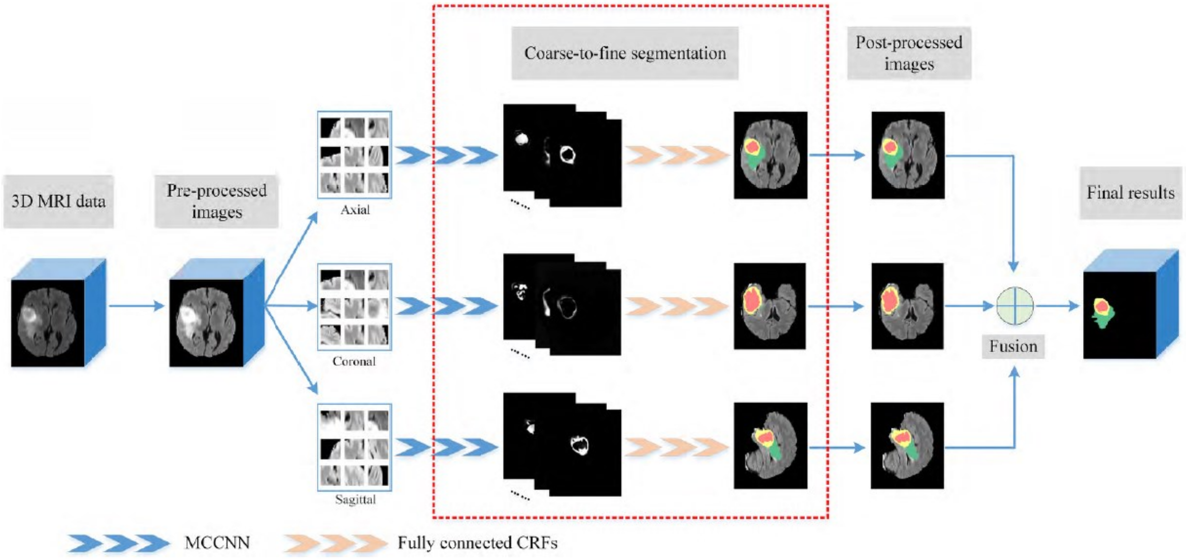


Figure 2.17 Flowchart of the CNN presented in [28].

DeepMedic [29] and DF-MLDeepMedic network [30] are CNNs with two 3D convolution pathways, to extract local and larger contextual features, respectively. The flowchart is shown in Figure 2.18. The 3D system presented in [31] involves dense connectivity structure, feature pyramid module, and deep supervision mechanism to improve the segmentation quality. It should be noted that, the convolutions in these 3D systems are performed with 3D input data, and thus 4D kernels are used in a convolution involving multiple channels, which requires more parameters to be trained with respect to standard 2D convolution.

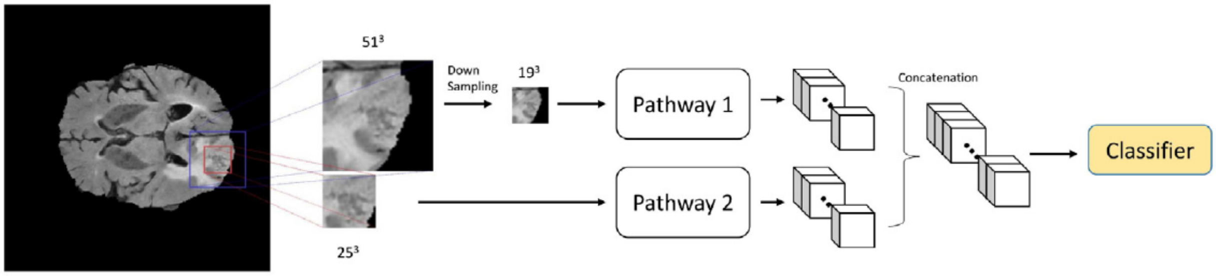


Figure 2.18 Flowchart of the CNN reported in [29][30].

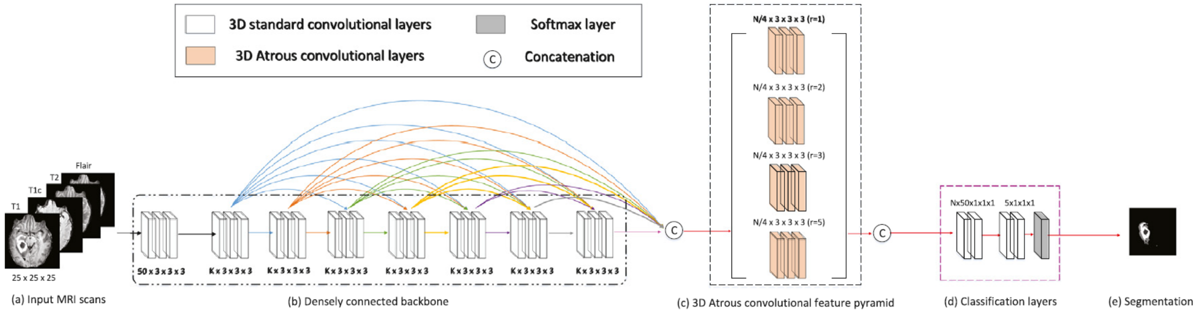


Figure 2.19. Flowchart of the CNN reported in [31].

2.4 Summary

In this chapter, a brief description of the fundamentals of CNN is given. It includes (i) the principle of 2D convolution and three convolution modes; (ii) the critical components of CNN, e.g., normalization, non-linear activation function, pooling and upsampling; (iii) basic blocks for CNN, like VGG, inception block, deep residual block and densely connected block, and (iv) training process and important setups for training.

In addition, the CNNs for brain tumor segmentation reported in recent years are presented in this chapter. Many of them are U-net-based network. Most of them increase the computation complexity in order to improve the segmentation quality.

Based on these investigation results, the work in this thesis will adopt the basic architecture of U-net, but the details are custom-design based on the characters of input signals to yield high-quality segmentation under low computation complexity.

3. Proposed System

3.1 Introduction

Automated brain tumor segmentation is an important processing for timely diagnosis in order to optimize treatments. In the design of the proposed system, the emphasis is to achieve a good segmentation quality at the lowest computation cost in order to facilitate its implementation. The basic block diagram of the system is shown in Figure 3.1.

The input of the proposed system is 3D brain images. The case of each patient is represented by four 3D images, namely FLAIR, T1, T1c and T2, as shown in Figure 3.1. Each of the 3D images is sliced into 2D slices. Hence, a voxel in a 3D image becomes a pixel in a 2D slice, and the brain tumor segmentation of 3D images is done by segmenting these 2D slices. An example of the 2D slices is shown in Figure 3.2 (a)~(d), and the ground truth of the brain tumor segmentation corresponding to these slices is shown in Figure 3.2 (e). It is difficult to find a deterministic model that can be used to detect various brain tumors in a pixel-wise precision. CNN can be potentially useful to handle such a problem if there is an appropriate dataset for training and testing the system.

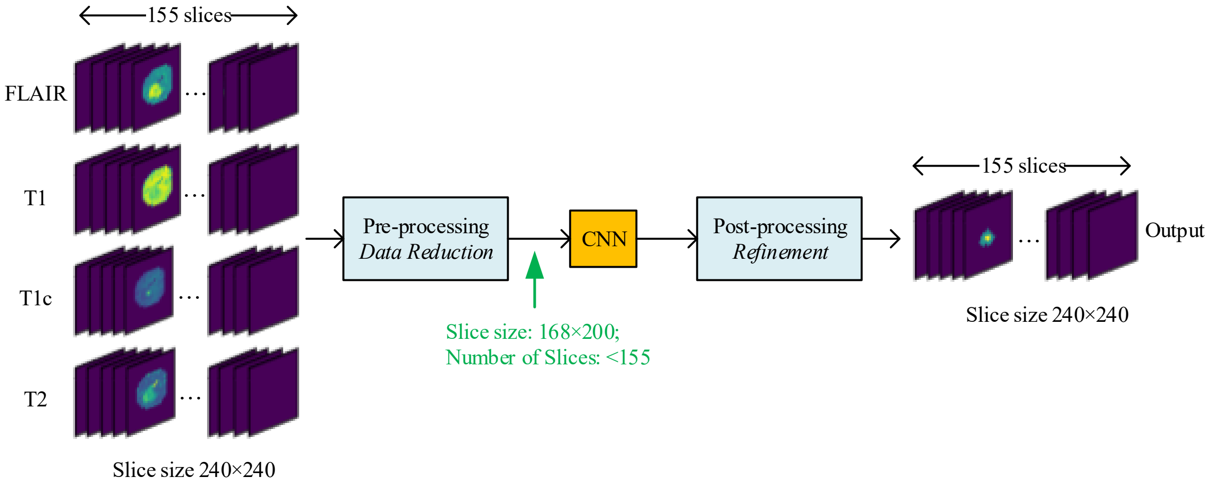


Figure 3.1 Overview of the proposed system. There are four 3D images namely FLAIR, T1, T1c and T2 in each patient case, and the size of the 3D brain images is 240×240×155.

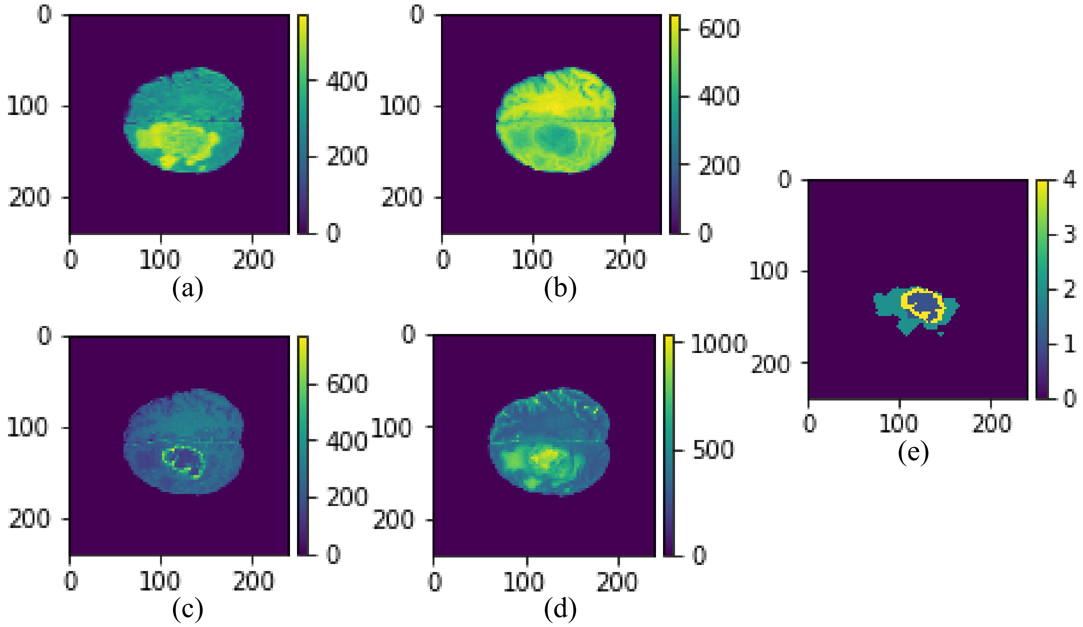


Figure 3.2 (a) FLAIR slice. (b) T1 slice. (c) T1c slice. (d) T2 slice. (e) Ground true.

In general, CNN systems require a very large amount of computation. In this design, a simple CNN is proposed to detect various features of brain tumors and to segment different forms of brain tumor areas. To facilitate the computation in the CNN, pre- and post-processing blocks are used, as shown in Figure 3.1. The functions of the three parts are as follow.

- Pre-processing. It is to reduce the volume of input data applied to the succeeding CNN, without any risk of losing feature data of brain tumors.
- CNN. It is composed of filtering layers to detect features of brain tumors and to generate maps indicating the pixels of candidates of brain tumors in each 2D slice.
- Post-processing. It is to remove possible false positive pixels.

The details of the pre-processing, the CNN and the post-processing are shown in sub-chapters 3.2, 3.3 and 3.4, respectively. In addition, a summary of this chapter is given in sub-chapter 3.5.

3.2 Pre-Processing – Reduction of Data Volume

The system is designed to operate with 3D brain images. The pre-processing block is made to reduce the data volume of the 3D images to facilitate the computations in the CNN. The dimensions of the 3D images from commonly used datasets, such as BRATS2017 and BRATS2018, are 240×240×155, resulting from a post-acquisition registration.

The brain image segmentation is, in fact, to identify the pixels in different tumor areas. As a result, the tumor-free brain areas will merge with the background, and the pixels in these locations will be classified in the same tumor-free class. Evidently, a large majority of the pixels are found in this class, and some of them can be very easily identified. Excluding these pixels, or a large part of them, from the operations in the CNN will significantly reduce the amount of data to be processed without causing any signal loss, which will result in not only a better computation efficiency but also a lower risk of false positive result.

In the pre-processing block, the reduction of the data volume is done effectively with very insignificant amount of computation, thanks to the 2 facts. Firstly, there is a wide margin in each side of a brain slice. Excessive margins can be easily identified and removed. Secondly, out of the 155 slices in each 3D image, some slices are free of tumor areas, and they can be removed without affecting the quality of the segmentation.

The pre-processing is done in 2 steps. The first step is to remove the excessive margins in each slice, and the second is to identify and remove tumor-free slices. The details in the design are found in the following sub-sections.

3.2.1 Detection and Removal of Excessive Margins in Slices

In each slice of a 3D brain image, the brain area occupies only a small portion of the space, due to a wide margin in each side, and one such example is shown in Figure 3.3 (a). It is very easy to detect the widths of the 4 margins, as the pixels in the background have the gray-level of zero. The preprocessing is to cut off excessive margins in order to reduce the size of the slices.

Let x_1 denote the distance between the left boundary of a slice and the leftmost point of the brain area, x_2, y_1 and y_2, denote the distances, respectively, to the other three boundaries of the slice, as shown in Figure 3.3 (a). The values of x_1, x_2, y_1 and y_2 change from slice to slice. Let x_{1m}, x_{2m}, y_{1m} and y_{2m}, to be the widths of the 4 rectangle margins to be cut-off. To apply them to all the slices without risk of data loss due to over-cutting, their values are determined by the minimum distances found in all the slices of the 3D images.

In case of the dataset BRATS2018, the minimum value of x_1, x_2, y_1 and y_2 are 29, 16, 38 and 38 pixels, respectively, which can be found in the histograms in Figure 3.4. Taking all the issues mentioned above, x_{1m}, x_{2m}, y_{1m} and y_{2m} are chosen to be 26, 14, 36 and 36, respectively, and the size of the slices is changed from 240×240 to 168×200 after the removal, as the examples showed in Figure 3.3 (b). The reduction of the data volume in this case is 42%. This simple procedure can be applied to the input from different datasets. In many cases, it helps to significantly reduce the computation volume in CNN processes.

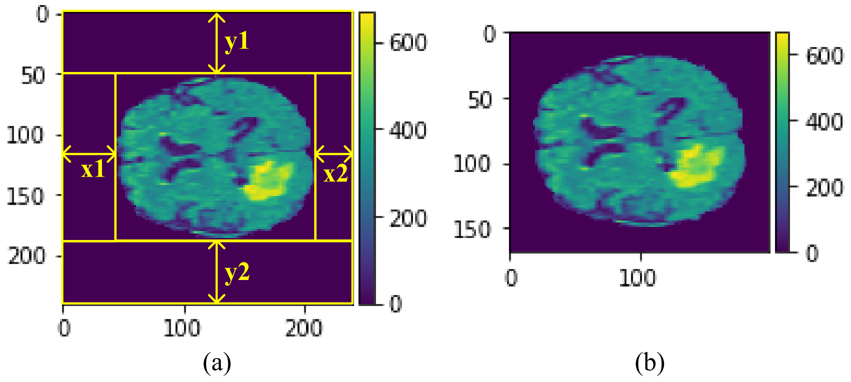


Figure 3.3 (a) Example of the original input slices sized 240×240 pixels. (b) Slice after the margin removal, sized 168×200 pixels.

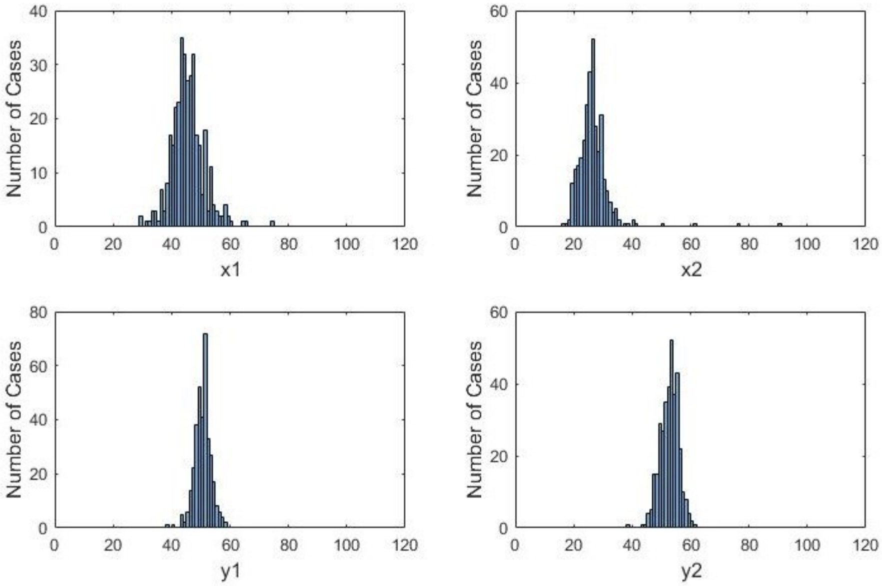


Figure 3.4 Histograms of x_1, x_2, y_1 and y_2 of BRATS2018 training set

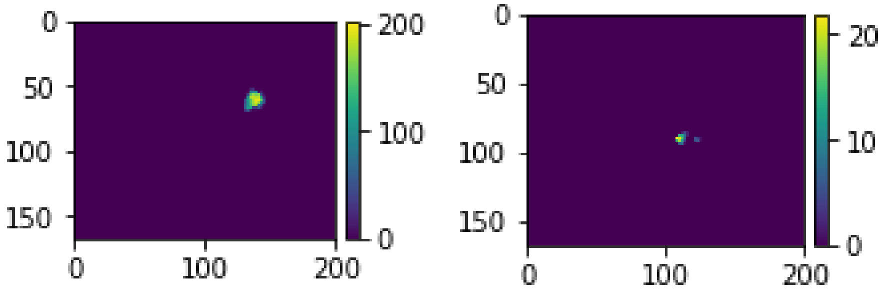
3.2.2 Detection and Removal of Tumor-Free Slices

There are three kinds of tumor-free slices in brain images.

i. In the sequence of the 155 slices in each 3D image, brain tumor areas are seldom found in the slices located at the 2 ends of the sequence. The slices of this kind can be identified by their very high percentage of background pixels. Two examples of such slices are shown in Figure 3.5.

ii. The slices of the second kind are of healthy section of a brain. Healthy brains have natural left-right symmetry, which is reflected to the upper-lower symmetry in brain images [32][33], as an example shown in Figure 3.6 (a)(c). A tumor causes an asymmetry, as a slice with tumor shown in Figure 3.6 (b)(d). The development of a brain tumor increases the asymmetry between the upper and lower halves of the slices. Thus, a tumor-free slice of the second kind can be identified by a high degree of upper-lower symmetry of the pixel data in the brain area.

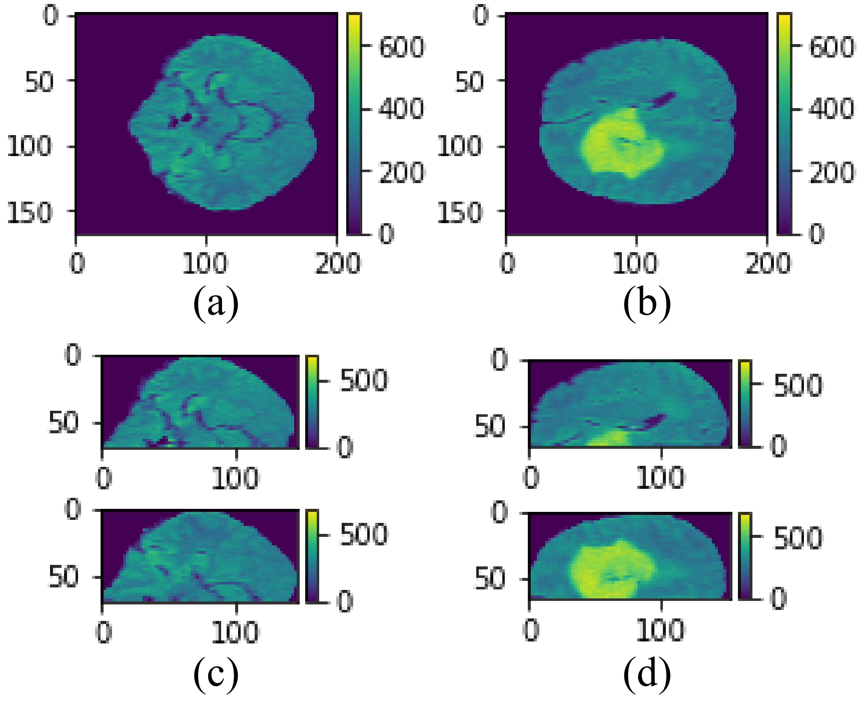
iii. Some slices have only incomplete brain areas appearing, as examples shown in Figure 3.7 (a) and (b), due to imperfection in medical image acquisition. As these slices are located in the areas where it is very rare to find brain tumors, they are considered tumor-free. A slice of this kind may not have particularly high percentage of background pixels, but a significant asymmetry between the upper half and the lower halves of the brain outlines, as shown in Figure 3.7 (c) and (d). Hence one can identify it by calculating the degree of symmetry of the outlines.



(a)

(b)

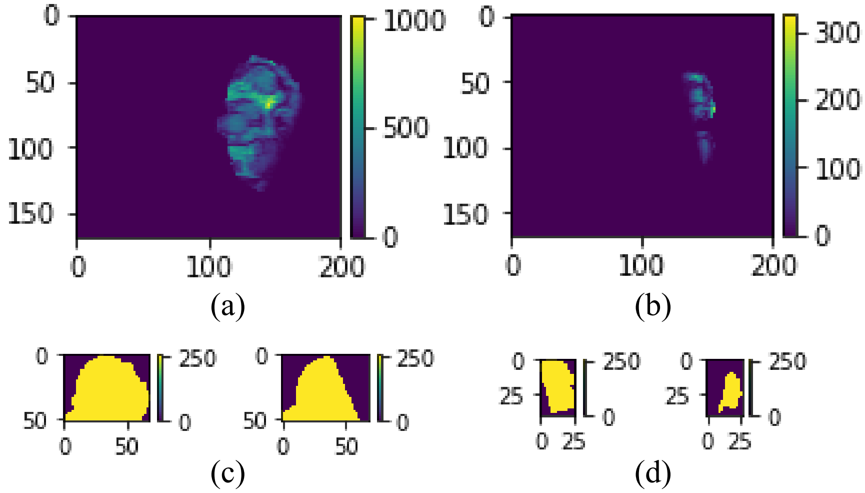
Figure 3.5 Examples of slices with high percentage of background pixels.



(a) (b)

(c) (d)

Figure 3.6 (a) An example of a FLAIR slice without tumor. (b) An example of a FLAIR slice with tumor. (c) Upper and lower halves of the contents in (a). (d) Upper and lower halves of the contents in (b).



(a) (b)

(c) (d)

Figure 3.7 (a) (b) Examples of slices with incomplete brain areas. (c) Upper and lower halves of the binary image of (a) that highlights the outline of the brain area. (d) Upper and lower halves of the binary image of (b).

Evidently, a tumor-free slice can be identified by a simple method to detect (1) a very high percentage of background pixels, (2) a high degree of symmetry of pixel data in the upper-lower halves of the brain area, or (3) a low degree of symmetry of the upper-lower brain outlines. Once identified, it is removed from the input slices, as all of its pixels are in the tumor-free class, without need for computation.

There are various approaches to detect the similarity between the upper and lower halves of brain area contents or brain area outlines. Structural similarity (SSIM) [34] is one of the commonly used approach for this purpose. It involves the mean, variance and covariance in the calculation.

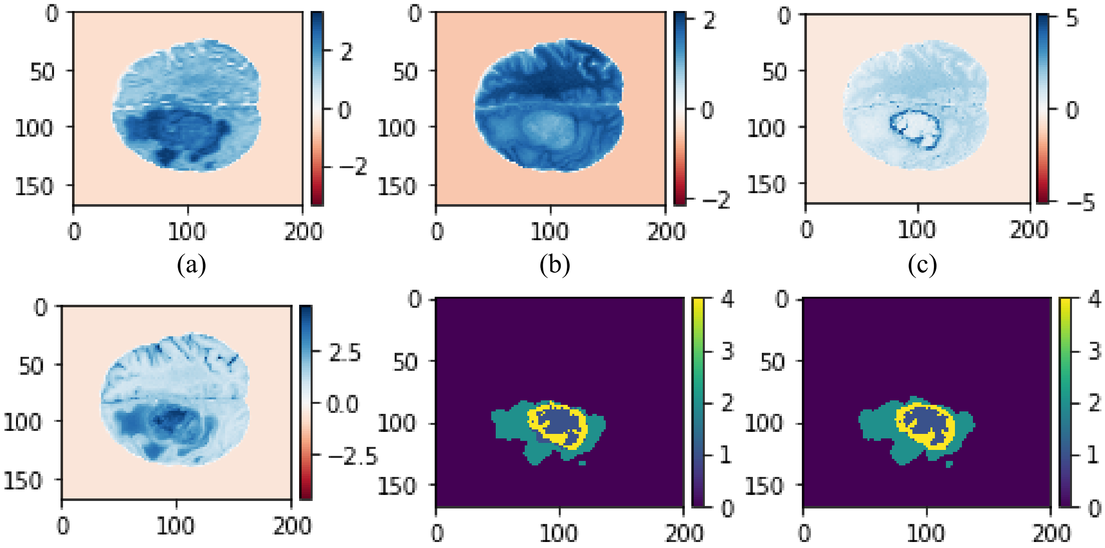
Applying the method to the samples from BRATS2018, one can find that the number of tumor-free slices per case is between 13 and 43, i.e., 8.4% ~ 28% of the 155 slices, which is not negligible. Combining it with the removal of the excessive margins in each slice, a reduction of more than 50% data volume can be achieved.

In conclusion, by cutting off the excessive margins in the slices and detecting/removing the tumor-free slices, one can expect a significant reduction of the data to be processed in the CNN stage.

3.3 CNN

The CNN has its input signal of four channels, given by the four 3D images, namely FLAIR, T1, T1c and T2, but the dimensions of the channels are smaller because of the pre-processing. Each of the four channels consists of 2D slices sized 168×200 pixels and indexed from 1 to N_{sl}. As the sizes of brains appearing in the 3D images are different, N_{sl}, the number of slices per channel, can vary, e.g., from 119 to 149, depending on cases of patients. Figure 3.8. (a)~(d) illustrates 4 slices of the same index number from the 4 input channels.

As mentioned previously, the function of the CNN is to identify the pixels in the four areas, namely the peritumoral edema (ED), the necrotic/non-enhancing tumor core (NCR/NET), the GD-enhancing tumor (ET) and the tumor-free areas. The classification results are used to indicate the areas of whole tumor (WT), tumor core (TC) and enhancing tumor (ET) as shown in Figure 3.8. (e).



(a) (b) (c)

(d) (e) (f)

Figure 3.8 Example of the input and output of the proposed CNN. (a)~(d) FLAIR, T1, T1c and T2 slice samples. In each slice, the intensity values are channel-wisely normalized. (e) Ground truth. (f) Example of predicted result.

In order to achieve a good segmentation accuracy at the lowest computation cost, instead of a general-purpose CNN, one needs an application-specific CNN (ASCNN), i.e., a CNN custom-designed for the task. To this end, an investigation of the input data is needed.

As the 3D brain images may not be acquired under the exactly same condition, the intensity range of the input data is not uniformed [35][36]. Thus, the input data need to be normalized before being applied to the convolution layers. However, unlike batch normalizations commonly used in CNNs, this normalization needs to be "channel-wise", performed to the data of each input channel, i.e., N_{sl} slices from each of the four 3D images of one given patient case. To be more specific, the data of each channel are normalized with the mean and the standard deviation of the channel. This kind of normalization uniforms the data range of all the channels, while minimizing the risk of attenuating critical feature information in channels of low-intensity levels.

As mentioned previously, there are a lot of variations in brain images. From microscopic point of view, it is hard to differentiate the gray level variations in the tumor areas from those in healthy areas. However, from a macroscopic point of view, the tumor textures look somehow different from those in the healthy parts. Hence, the detection of that difference needs image features extracted from a relatively large neighborhood. Moreover, as a tumor can grow in any part of a brain, a division of a slice from a 3D brain image can result in a division of brain tumor neighborhoods, which may affect the quality of the texture detection. Therefore, the proposed CNN is designed to operate with undivided slices of 3D brain images.

Based on the investigation of the input data, the strategy of this design is for the CNN to perform 2 functions in sequence. First function is to extract elementary features and to localize brain tumor areas, and the second is a fine classification of the pixels in the areas. The former requires a detection of both "fine" signal variations and "coarse" brain tissue textures, and the latter needs local feature information in the tumor areas. To do so, a simple CNN is proposed as illustrated in Figure 3.9. Though it is based on the main frame of U-net, for a better performance without need for more computation, the proposed network is designed to have different characters, such as i) one sole convolution per layer in all of its 7 layers, ii) uniformly 16 kernels per layer, except the last layer, iii) modulating the upsampled data directly by weighted early feature data by means of 1x1 convolution and addition, and iv) applying different convolution modes.

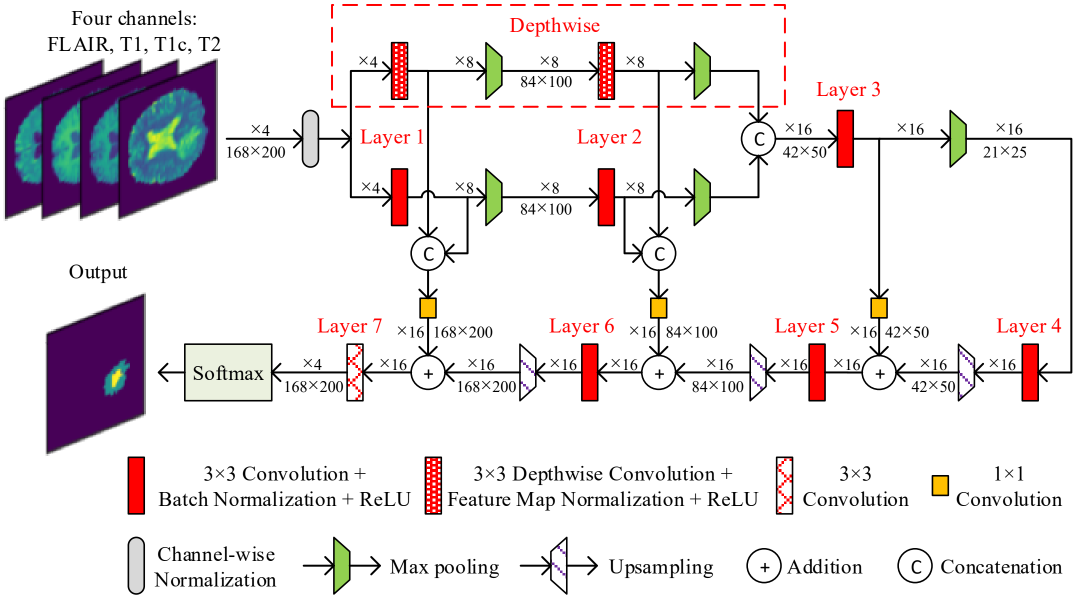


Figure 3.9. Detailed diagram of the proposed CNN.

Table 3.1 Details of the CNN configuration

Layer#	Kernel Size	Number of input channels	Number of output channels	Size of input channel	Size of output channel	Number of parameters
1	3×3	4	16	168×200	84×100	376
2	3×3	16	16	84×100	42×50	664
3	3×3	16	16	42×50	21×25	2320
4	3×3	16	16	21×25	42×50	2320
5	3×3	16	16	42×50	84×100	2320
6	3×3	16	16	84×100	168×200	2320
7	3×3	16	4	168×200	168×200	580
Three 1×1 Convolutions, 272 parameters each						816
Total						11716

The feature extraction is performed in the first three convolution layers and the operation in the 4th layer is to detect the tumor areas. As the inputs of the network are of simple gray scale images, not color ones, the number of output channels per layer is chosen to be 16. The following considerations have been taken in the design of these layers.

i. It should be mentioned that the input image data of the four 3D images, namely FLAIR, T1, T1c and T2, are acquired with different emphases on different lesion areas of a specific brain. Though there is a correlation among the four 3D images, each of them contains enhanced features of particular intra-tumoral structures. To make good use of the signals from the four slices taken from the same brain section, two different modes of convolutions are used in the first 2 layers to extract various elementary features. As shown in Figure 3.9, in the upper part of the first 2 layers, so-called depthwise convolutions, represented by red-white-polka-dot rectangles, are applied to individual 2D slices. For example, in the first layer, there are four 3×3 kernels in the depthwise convolution and each of them is applied to the data of a single 2D slice to generate a 2D feature map representing the particular features of the slice. Each depthwise convolution is followed by a feature map normalization applied to each individual feature map. Standard convolutions, indicated by solid-red rectangles in the diagram, are also used to generate 2D feature maps, and each of these maps is based on the data originated from the 2D slices taken from the FLAIR, T1, T1c and T2 channels. The two kinds of convolutions are performed in parallel in the first 2 layers, as shown in Figure 3.9, and the results are then concatenated before being applied to the 3rd convolution layer.

ii. Following each of the first 3 convolution layers, a pooling operation is performed to each of the 2D data maps, as shown in Figure 3.9. It is to "zoom out" the 2D map, so that the effective size of neighbourhood in the succeeding convolution will be larger than the size of the convolution kernels of 3×3 pixels. After the 3 layers of 3×3 convolution followed by pooling, the operation in the 4th layer will be performed with the input data originated from a quite large neighbourhood, i.e., in a quasi "macroscopic" scale.

iii. As each pooling operation is performed to the convolved data that are more likely the results of high-pass filtering, max pooling is the most suitable to minimize the information loss while the data volume being reduced significantly. It is also used to increase the information density of the data applied to the succeeding convolution layers.

The function of the first 4 layers combined can be seen as a coarse classification of tumor pixels and tumor-free ones. As the signal resolution is reduced by the pooling operations whereas the segmentation requires a pixel-wise precision, the second part of the proposed CNN is designed to have the following 2 characters.

- Upsampling operations are performed so that the dimensions of the output maps will grow to be the same as those of the input slices of the CNN. Bilinear upsampling is used for this purpose. However, the upsampling does not improve the precision of image signal.
- To precisely classify the pixels in each slice, the filtering results of the first convolution layers, i.e. local feature data, are included in the convolutions of the last 3 layers. Instead of concatenation, these data are first scaled by trainable coefficients and then added to the upsampled data, as shown in Figure 3.9. This addition can be seen as a modulation of the upsampled data by the local feature data, or vise versa. It results in an enhancement of the feature data in tumor areas, which implies an attenuation of those in tumor-free areas. Hence the filtering operations in the last 3 convolution layers are performed to the data well prepared with pertinent feature information of brain tumors for a fine classification.

The proposed CNN is designed specifically to suit characters of the 3D-four-channel gray-scale input signals of brain images to optimize the filtering operations with a view to achieving a good processing quality at the lowest computation cost. In fact, the specific measures taken to improve the processing quality are all helping to reduce the computation complexity: The number of kernels per layer is constantly 16, without argument over the layers, which helps to achieve a simple computation and a high signal density in each output channel. Half of the convolutions in the first 2 layers are "depthwise", requiring only 1/*n* of computation, compared with standard convolutions, where *n* is the number of the input channels. In the second half of the CNN, the additions, instead of concatenations, of the filtered data from earlier layers and the upsampled data yields another very significant reduction of computation volume.

As the input data in each layer are well prepared for an efficient filtering operation, the proposed CNN has only 7 convolution layers, and each generates sixteen 2D maps, except the last one, as shown in Figure 3.9. The details of the CNN configuration are presented in Table 3.1. The total number of parameters of the network is 11716. It is likely the simplest U-net-based CNN that has been designed so far. With this low computation complexity, the network can be very easily trained and implemented in a recourse-restricted environment. Moreover, its simplicity in structure may also be beneficial for the consistency of its performance and reproducibility of the segmentation results.

3.4 Post-Processing – Refinement

After the classification by the CNN, the post-processing block is placed to identify the pixels that are falsely classified as positive ones. The identification is based on the fact that a brain tumor and its enhanced tumor core, if it exists, are 3D objects, and the area of each of them must be found in a certain number of consecutive slices.

The thickness of a detectable whole tumor is considered to be at least 1/20 of the diameter of a brain. If a 3D brain image consists of 155 slices, this thickness corresponds to at least seven consecutive slices. If a whole tumor area appears in fewer than seven consecutive slices, the pixels in this area are likely falsely classified, and they will be re-classified as tumor-free pixels.

The identification of the false-positive pixels of enhancing tumor core is based on a similar principle that tumor cores have their own minimum size limit. Also, it is common sense that the minimum size of a tumor core is slightly smaller than that of a whole tumor. If a predicted tumor core area appears in fewer than six consecutive slices, instead of seven, the pixels in the area will be considered false-positive and then be reclassified as non-enhancing tumor pixels.

By applying the principles mentioned above, the post-processing block improves the precision of segmentation without adding a perceivable amount of computation.

3.5 Summary

The details of the proposed system are presented in this chapter. This system is composed of three parts, pre-processing block, CNN and refinement block. The pre-processing is to detect and remove the excessive margins in each slice and the tumor-free slices. The CNN is designed specifically according to the characters of the data in brain images, aiming at efficient feature extraction and precise classification. The refinement is to remove possible false positive voxels.

In conclusion, the CNN, which is designed specifically based on the characters of the MRI brain images, combined with the pre- and post- processing blocks, segments brain tumors precisely and efficiently.

4. Performance Evaluation

4.1 Introduction

The proposed system for brain tumor segmentation has been trained and tested with a commonly used dataset, namely BRATS2018, for its performance evaluation. Python and Tensorflow have been used in developing & testing the system, and a NVIDIA P100 Pascal GPU with 12GB HBM2 memory has been used to run the programs. As described in chapter 3, the proposed CNN has only 11716 parameters and requires only 21.14G FLOPs for the test of a patient case. Thus, it is very easy to be trained and tested. The details of the training and testing processes are presented in the following subsections. The performance of the proposed system has been assessed by CBICA Image Processing Portal and compared with those of methods reported in recent years.

This chapter is organized as follow. Sub-chapter 4.2 describes the details of dataset and training processing. Sub-chapter 4.3 presents and comments the testing results, and the results produced by the proposed system are compared with those by state-of-the-art methods. Sub-chapter 4.4 gives a brief summary.

4.2 Dataset and Training

4.2.1 Dataset

The dataset BRATS2018 includes all of the samples from BRATS2017 dataset and part of the samples from BRATS2015. The number of patient cases and the partition of training/test sets are shown in Table 4.1.

Table 4.1 Partition of the samples from dataset BRATS2018

Dataset	Number of patient cases for training		Number of patient cases for testing
	HGG	LGG	
BRATS2018	210	75	66

As the number of patient cases from the dataset is very limited, data augmentation has been performed to have a decent training process. It was done by up-down and left-right flipping each slice in all the 3D brain images in the training set. Moreover, in order to include slices of different texture pattern in each batch, the slices in the training set have been sorted by shuffling them randomly.

4.2.2 Training Details

In the proposed CNN, there are 11716 parameters to be determined by means of training process. A number of elements are critical for the quality of the training:

i. Batch size and the number of training epochs. The batch size is chosen to be 100, and the training process is completed after 50 epochs.

ii. Learning rate. Cosine Decay [40] is chosen to make the learning rate variable from 0.01 to 1×10^{-6}. In this way, the system loss is reduced coarsely and quickly during the first epochs and is then adjusted finely in the last epochs.

iii. Loss function. The loss function is chosen to be Cross Entropy [17].

iv. Optimizer. Adam (Adaptive Moment Estimation) [20] is used as the optimizer in the training process.

v. Initialization. The initial weights are chosen to be truncated normal distribution with 0.1 standard deviation, and the initial biases are chosen to be 0.1.

The loss curve of training process of the proposed system is shown in Figure 4.1. It has been confirmed that the loss is reduced quickly during the first 10 epochs, and only 50 epochs are needed to complete the training process.

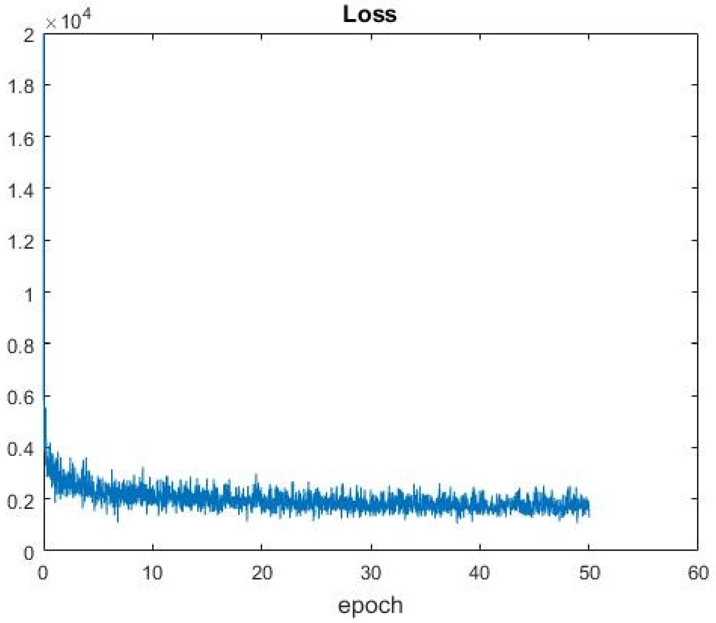


Figure 4.1 Loss curve in the training process of the proposed system.

4.3 Testing and Performance Evaluation

After determining the values of the trainable parameters in the CNN by the training process, the proposed system has been tested by using BRATS2018 test (validation) set. To have the results of evaluation formally recognized, the output data of the proposed system have been examined by CBICA Image Processing Portal [37], an online evaluation platform, where the assessment is a standard process with data from the Cancer Imaging Archive [38][39].

4.3.1 Performance Metrics

There are four commonly used metrics to evaluate the segmentation quality, namely Dice score (*Dice*), Sensitivity (*Sens*), Specificity (*Spec*) and Hausdorff95 distance (*Haus*) [2]. The first three metrics are expressed as follows.

$$Dice(P_1,\, T_1) = \frac{P_1 \wedge T_1}{(P_1 + T_1)/2} \tag{4.1}$$

$$Sens(P_1, T_1) = \frac{P_1 \wedge T_1}{T_1} \tag{4.2}$$

$$Spec(P_0,\, T_0) = \frac{P_0 \wedge T_0}{T_0} \tag{4.3}$$

where P_0 and P_1 are the predicted results, indicating the number of voxels in the non-tumor regions and that in the tumor regions, respectively, as shown in Figure 4.2, whereas T_0 and T_1 are those in the ground truth. In addition, the *Haus* is also used to indicate the distance between the predicted tumor boundaries and those of the ground truth [2].

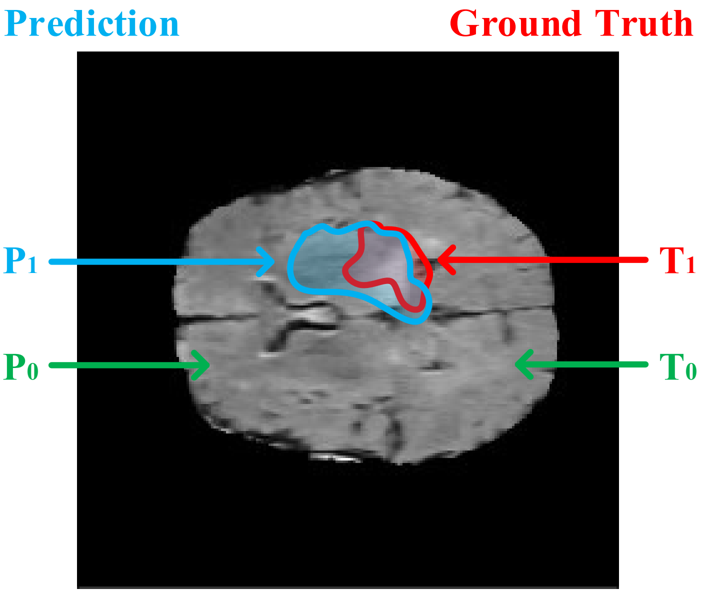


Figure 4.2 Slice of segmented brain image to indicate the lesion. T_1 is the number of pixels in the true lesion region, which is located in the red-contoured area. P_1 is the number of pixels in predicted lesion region, which is located in the blue-contoured area. T_0 and P_0 are the number of pixels in normal regions in the ground truth and the predicted maps, respectively.

As the tumor voxels take a very small space a 3D brain image, we have

$$P_1 \ll P_0, \qquad T_1 \ll T_0, \qquad P_0 \approx T_0 \approx P_0 \wedge T_0, \qquad Spec(P_0, T_0) \approx 1$$

As $Spec(P_0, T_0)$ is almost equal to unit in most of brain tumor segmentation cases, and it does not indicate sensitively a difference in identification of true/false negative voxels.

To better evaluate the quality of the segmentation in different aspects, the metrics of false discovery rate (*FDR*) [41] and false negative rate (*FNR*, or miss rate) [42] of the proposed system have also been measured. The false discovery rate is the ratio of the number of false positive voxels to the total number of predicted positive voxels, indicating how many voxels are falsely predicted to be positive. The false negative rate or miss rate is the ratio of the number of false negative voxels to the total number of positive voxels in the ground truth. They are expressed as follows.

$$FDR(P_1, T_1) = \frac{P_1 - (P_1 \wedge T_1)}{P_1} = 1 - \frac{1}{2/Dice(P_1, T_1) - 1/Sens(P_1, T_1)} \tag{4.4}$$

$$FNR(P_1, T_1) = \frac{T_1 - (P_1 \wedge T_1)}{T_1} = 1 - Sens(P_1, T_1) \tag{4.5}$$

The performance metrics of a CNN system also include the measure of the computation volume required to achieve the processing quality, as it is related to the computation efficiency, the feasibility of system implementations and the range of applications. The number of parameters in the CNN is an important indicator of the computation volume in both training and testing process. The number of floating-point operations (FLOPs) required to complete a test for one patient case is related to the applications of the system, as it determines where the system can be installed and how fast the process will be.

4.3.2 Testing Results

As mentioned previously, the performance assessment of the proposed system has been done with the dataset BRATS2018, and all the results have been generated by CBICA Image Processing Portal. Ten experiments have been conducted. Each experiment has been done by (i) training the system from the initial state and (ii) testing all the 66 test cases in the testing pool and generating 66 results.

If the system is functional, it should deliver the results of high processing quality in a consistent manner. The degree of the consistency reflects the degree of the reliability and confidence of the results. Therefore, the assessment of the consistency in system performance should be part of the validation of the results.

To assess the consistency, the 10 sets of 66-results have been examined in the aspects of the statistical feature data. Each of the 66 results from every single experiment includes the 3 *Dice* scores for ET, WT and TC. The 10 histograms, presented in Figure 4.3, illustrate the distributions of the *Dice* scores of ET obtained in the 10 experiments. To visualize the distributions in form of histogram, the score values are quantised with a precision of 0.02. The distributions of the *Dice* scores of WT and TC are illustrated in Figure 4.4 and Figure 4.5, respectively. The results are also visualized, as shown in Figure 4.6, in form of boxplots. The mean, median and mode values of each distribution are presented in Table 4.2 to give more statistical characters of the performance. The mean values of *Dice* score are also presented in Table 4.3, together with the mean values of Sensitivity (*Sens*), Specificity (*Spec*) and Hausdorff95 distance (*Haus*).

One can have the following observations of the distributions of the 3 *Dice* scores.

(i) The 10 distributions in each of the three categories of *Dice* scores, i.e., ET, WT, or TC, have a high degree of similarity. The 10 mean, or 10 median, or 10 mode, values are very close to one another with a small deviation of less than 0.013, as shown in Table 4.2 and Table 4.3. It indicates that, though the 10 individual training processes cannot set up the parameters of the system in an identical manner, the system is trained to have quasi-determined characteristics.

(ii) The *Dice* scores illustrate that the proposed system is able to deliver high-quality segmentation results. For enhancing tumors, the median Dice score is 0.85, and the value of the mode is 0.9. The mean values of all the 3 Dice scores are, consistently in all the 10 experiments, lower than the median values. In fact, in all the 10 experiments, only approximately 22% of the 66 patient cases, which is less than the first quantile, have got ET (or WT) *Dice* scores lower than the mean value, and in case of TC *Dice* scores, it has been around 32%. Thus, the mean scores are with a small minority of the cases.

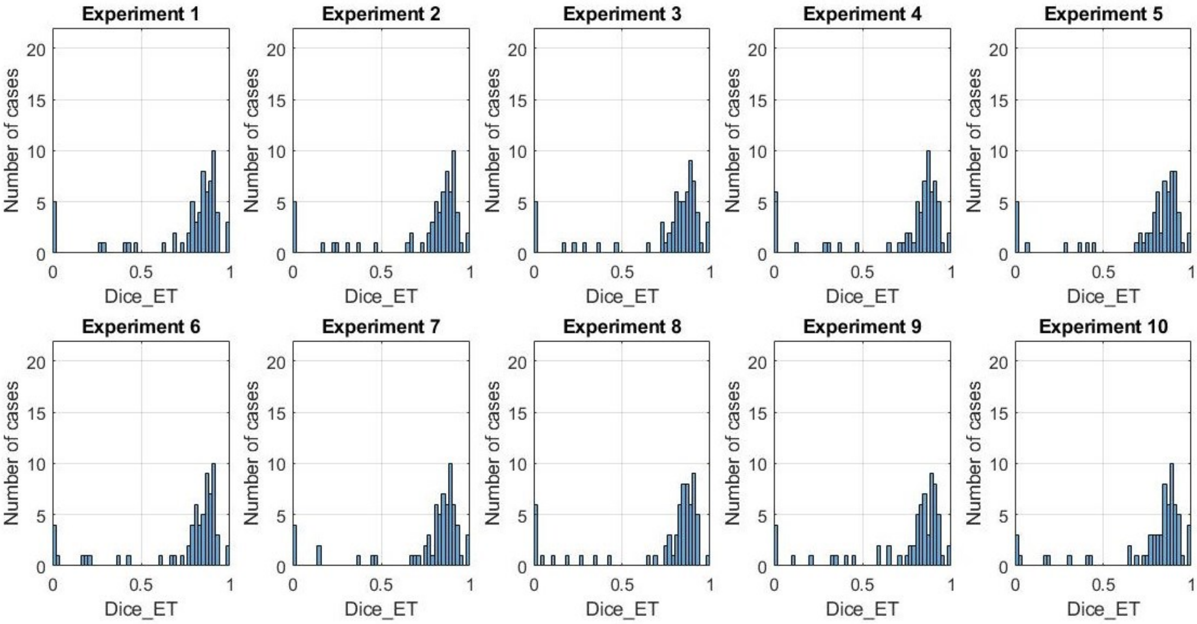


Figure 4.3 Distributions of *Dice* scores in ET areas obtained in the ten experiments.

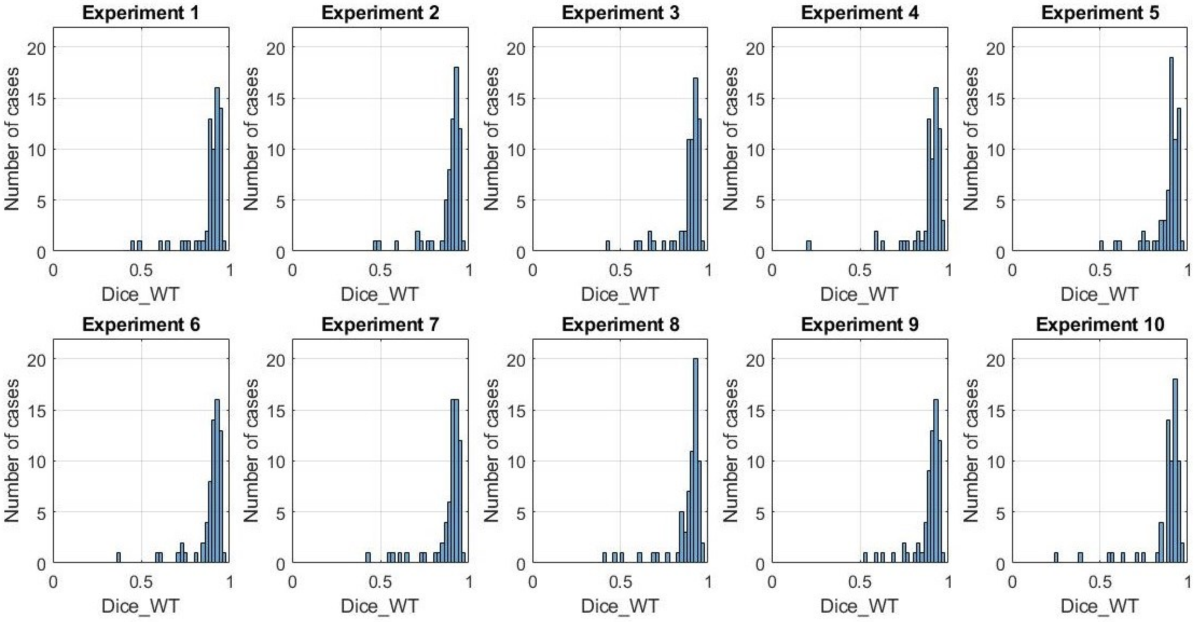


Figure 4.4 Distributions of *Dice* scores in WT areas obtained in the ten experiments.

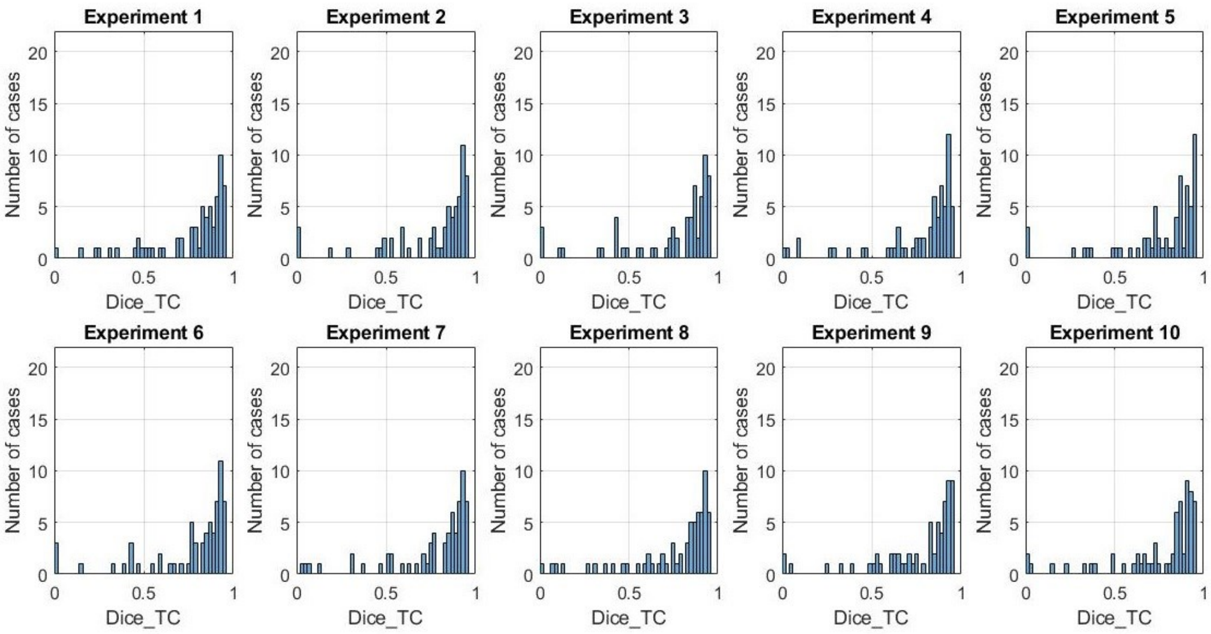


Figure 4.5 Distributions of *Dice* scores in TC areas obtained in the ten experiments.

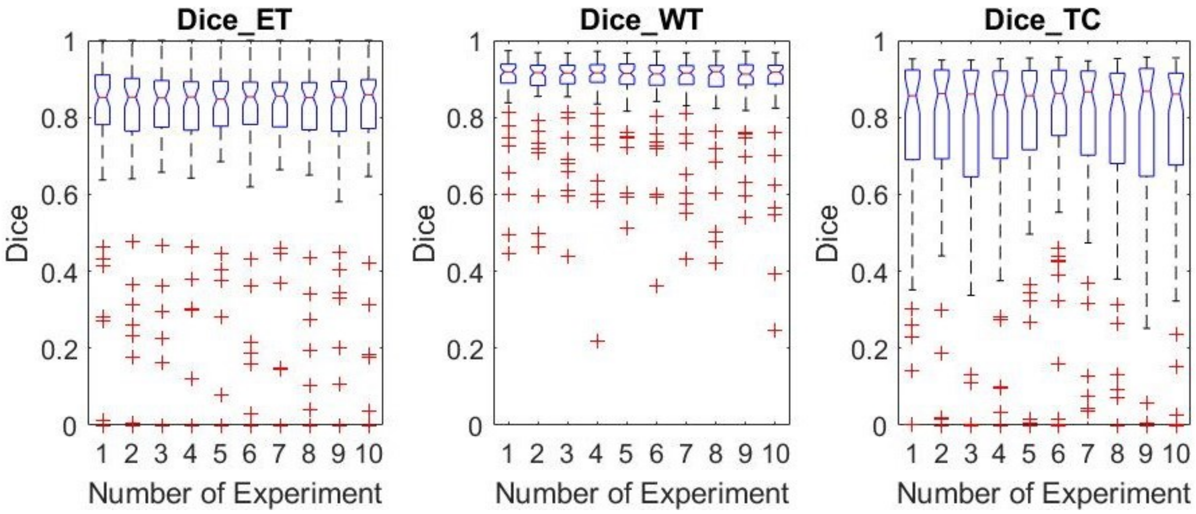


Figure 4.6 Boxplots of the test results obtained in the ten experiments.

Table 4.2 Statistical data of the ten experiments

Exp.	*Dice* - Mean			*Dice* - Median			*Dice* - Mode		
	ET	WT	TC	ET	WT	TC	ET	WT	TC
No. 1	0.7568	0.8869	0.7612	0.8518	0.9184	0.8564	0.91	0.93	0.93
No. 2	0.7423	0.8863	0.7656	0.8524	0.9161	0.8619	0.91	0.93	0.93
No. 3	0.7519	0.8851	0.7407	0.8504	0.9153	0.8607	0.89	0.93	0.93
No. 4	0.7390	0.8842	0.7615	0.8528	0.9157	0.8581	0.87	0.93	0.93
No. 5	0.7492	0.8892	0.7612	0.8473	0.9143	0.8564	0.89	0.91	0.95
No. 6	0.7456	0.8861	0.7629	0.8532	0.9132	0.8624	0.91	0.93	0.93
No. 7	0.7645	0.8835	0.7597	0.8555	0.9151	0.8662	0.89	0.91	0.93
No. 8	0.7222	0.8814	0.7542	0.8516	0.9185	0.8587	0.91	0.93	0.93
No. 9	0.7483	0.8895	0.7617	0.8523	0.9124	0.8678	0.89	0.93	0.93
No. 10	0.7663	0.8766	0.7545	0.8588	0.9174	0.8605	0.89	0.93	0.91
Average	0.7486	0.8849	0.7583	0.8526	0.9156	0.8609	0.90	0.93	0.93
STDEV	0.0128	0.0038	0.0071	0.0030	0.0020	0.0038	0.013	0.008	0.009

Table 4.3 Mean scores obtained in the ten experiments

Exp.	*Dice*			*Sensitivity*			*Specificity*			*Hausdorff95*		
	ET	WT	TC	ET	WT	TC	ET	WT	TC	ET	WT	TC
No. 1	0.7568	0.8869	0.7612	0.7873	0.8793	0.7697	0.9977	0.9949	0.9965	4.6582	7.9896	10.8728
No. 2	0.7423	0.8863	0.7656	0.7735	0.8765	0.7569	0.9980	0.9950	0.9972	4.5158	6.6366	11.5508
No. 3	0.7519	0.8851	0.7407	0.7720	0.8737	0.7481	0.9980	0.9950	0.9964	4.5532	6.4869	9.5567
No. 4	0.7390	0.8842	0.7615	0.7722	0.8763	0.7537	0.9979	0.9950	0.9971	5.7798	5.3184	10.6471
No. 5	0.7492	0.8892	0.7612	0.7796	0.8804	0.7596	0.9979	0.9948	0.9968	4.6720	4.6038	9.4975
No. 6	0.7456	0.8861	0.7629	0.7825	0.8787	0.7643	0.9978	0.9948	0.9968	4.6195	6.4392	10.5515
No. 7	0.7645	0.8835	0.7597	0.7889	0.8739	0.7632	0.9979	0.9948	0.9968	5.5765	7.9587	10.0286
No. 8	0.7222	0.8814	0.7542	0.7711	0.8710	0.7531	0.9978	0.9949	0.9970	4.8184	7.2094	10.9618
No. 9	0.7483	0.8895	0.7617	0.7829	0.8798	0.7540	0.9977	0.9948	0.9969	5.7496	6.5947	11.2316
No. 10	0.7663	0.8766	0.7545	0.7872	0.8670	0.7579	0.9977	0.9949	0.9968	5.2887	5.6498	12.1282
AVERAGE	0.7486	0.8849	0.7583	0.7797	0.8757	0.7580	0.9978	0.9949	0.9968	5.0232	6.4887	10.7027
STDEV	0.0128	0.0038	0.0071	0.0070	0.0043	0.0064	0.0001	0.0001	0.0003	0.5182	1.0850	0.8415

The system is trained with 285 cases in the training pool of BRATS2018, of which 210 are HGG cases, i.e., cases with more aggressive form of the disease, with respect to less aggressive low-grade gliomas (LGG) [2]. Hence, the system is tuned to operate better with HGG cases. However, one cannot give a certain answer to the question if the cases of low *Dice* scores in the experiments described above belong to LGG group, as the dataset does not give indication of HGG or LGG for any of the 66 testing cases. Nevertheless, one can check if these cases have characters commonly found in LGG cases.

In general, the size of enhancing tumors (ET) of HGG cases is bigger than that of LGG cases. The 66 patient cases can be divided into Big ET (BET) group and Small ET (SET) group so that one can evaluate the performance of the proposed system separately in each of the 2 groups. Since the ground truths of the test set are not provided to users, the cases are grouped by using the results of prediction.

The size of a 3D enhancing tumor area can be measured by the number of voxels in the area. An enhancing tumor can be considered as large, if it occupies 0.1% or more of the 3D brain image. This 0.1% is interpreted as 8928 voxels out of 240×240×155 in each brain image. In case of ground truths or predicted images, the identified tumor areas are usually slightly larger than the real one. Thus, the threshold used to group BET and SET is 9450, instead of 8928.

Applying the threshold described above to the 66 cases in the testing pool of BRATS2018 dataset, 44 cases are put in the BET group and the remaining 22 in the SET group. The scores of all the individual cases in BET group and those of SET group obtained in Experiment 5 are presented in Table 4.4 and Table 4.5, respectively. They illustrate that:

(i) The proposed system yields excellent results for Big ET cases. The average *Dice* scores of ET, WT and TC areas of Big ET cases are 0.8579, 0.9126, 0.8577, respectively, and the standard deviations of those are 8%, 3.76% and 9.93%.

(ii) The results of the Small ET cases are not good as those of the Big ET cases, because there are only 75 LGG training samples in the dataset BRATS2018, i.e., 26% of the training samples.

Table 4.4 Test results of 44 Big enhancing tumor (BET) cases in Experiment 5

Label	*Dice*			*Sensitivity*			*Specificity*			*Hausdorff95*		
	ET	WT	TC	ET	WT	TC	ET	WT	TC	ET	WT	TC
Brats18_CBICA_AAM_1	0.9352	0.8732	0.9379	0.9739	0.8036	0.9601	0.9988	0.9966	0.9989	1.4142	8.0623	1.4142
Brats18_CBICA_ALA_1	0.8727	0.8969	0.7322	0.9231	0.9415	0.9510	0.9988	0.9941	0.9950	2.8284	2.8284	14.3527
Brats18_CBICA_ALT_1	0.4452	0.8859	0.8624	0.7948	0.8352	0.8178	0.9771	0.9944	0.9947	3.6056	4.5826	7.2801
Brats18_CBICA_ALZ_1	0.7771	0.9318	0.9229	0.6755	0.9884	0.8818	0.9993	0.9889	0.9994	2.0000	2.0000	3.0000
Brats18_CBICA_AMF_1	0.8783	0.9192	0.9461	0.8888	0.9671	0.9584	0.9968	0.9885	0.9972	1.4142	2.0000	3.0000
Brats18_CBICA_AMU_1	0.8400	0.9261	0.9058	0.9912	0.9212	0.9780	0.9920	0.9906	0.9931	3.1623	4.8990	21.4709
Brats18_CBICA_APM_1	0.8498	0.9041	0.8733	0.8139	0.8451	0.8300	0.9972	0.9984	0.9971	1.4142	3.6056	4.3589
Brats18_CBICA_AUE_1	0.8938	0.8901	0.9444	0.8868	0.8829	0.9521	0.9970	0.9866	0.9968	1.4142	6.7082	2.8284
Brats18_CBICA_BHF_1	0.9247	0.9344	0.9164	0.9535	0.9433	0.8941	0.9988	0.9968	0.9991	1.0000	2.0000	2.0000
Brats18_CBICA_BHN_1	0.8829	0.9068	0.9421	0.8801	0.8586	0.9356	0.9983	0.9981	0.9987	1.4142	2.8284	2.2361
Brats18_CBICA_BKY_1	0.7888	0.9185	0.9058	0.8086	0.8877	0.8738	0.9969	0.9963	0.9976	2.2361	3.0000	4.2426
Brats18_CBICA_BLI_1	0.9196	0.9325	0.9443	0.9262	0.9072	0.9437	0.9969	0.9968	0.9970	1.0000	2.2361	2.2361
Brats18_CBICA_BLK_1	0.8873	0.8534	0.8759	0.8553	0.9212	0.8717	0.9992	0.9908	0.9986	2.2361	5.0990	3.6056
Brats18_MDA_1012_1	0.8461	0.8934	0.9141	0.7988	0.9201	0.8842	0.9986	0.9939	0.9989	2.0000	3.1623	2.8284
Brats18_MDA_1015_1	0.8392	0.9328	0.9124	0.8636	0.9744	0.9221	0.9974	0.9921	0.9982	3.0000	5.3001	4.4721
Brats18_MDA_1081_1	0.8039	0.9027	0.8504	0.8655	0.9445	0.9363	0.9921	0.9932	0.9909	3.1623	5.8310	6.3246
Brats18_MDA_907_1	0.7557	0.8877	0.8790	0.7005	0.8199	0.8445	0.9980	0.9979	0.9983	4.0000	6.0000	3.3166
Brats18_MDA_922_1	0.8923	0.9300	0.9442	0.9272	0.9060	0.9508	0.9979	0.9956	0.9988	2.0000	3.3166	2.2361
Brats18_TCIA02_230_1	0.9278	0.9511	0.9212	0.8928	0.9173	0.8778	0.9986	0.9985	0.9983	1.4142	2.2361	7.6812
Brats18_TCIA02_400_1	0.9403	0.9040	0.9454	0.9588	0.9936	0.9601	0.9992	0.9921	0.9993	1.4142	5.0990	2.0000
Brats18_TCIA03_288_1	0.8857	0.9482	0.9405	0.8356	0.9105	0.9123	0.9990	0.9992	0.9993	1.7321	1.7321	2.0000
Brats18_TCIA03_313_1	0.8776	0.9567	0.9401	0.8174	0.9480	0.9355	0.9974	0.9973	0.9965	3.6056	3.4641	7.8740
Brats18_TCIA03_604_1	0.9100	0.9490	0.9434	0.8640	0.9094	0.9367	0.9993	0.9997	0.9988	1.4142	2.0000	2.0000
Brats18_TCIA04_212_1	0.9252	0.9550	0.9474	0.9197	0.9975	0.9350	0.9994	0.9970	0.9996	1.4142	2.2361	1.0000
Brats18_TCIA04_253_1	0.8410	0.9009	0.8401	0.7835	0.9031	0.7472	0.9985	0.9948	0.9991	3.1623	3.1623	4.5826
Brats18_TCIA07_600_1	0.9173	0.9459	0.9338	0.8741	0.9529	0.9090	0.9988	0.9951	0.9982	2.0000	5.0000	3.3166
Brats18_TCIA07_601_1	0.9146	0.9276	0.7860	0.9010	0.9210	0.6706	0.9986	0.9953	0.9987	1.7321	3.0000	46.5725
Brats18_TCIA07_602_1	0.8764	0.9060	0.9090	0.8238	0.8834	0.8529	0.9966	0.9934	0.9979	3.0000	9.4340	7.2801
Brats18_TCIA13_610_1	0.8623	0.9388	0.7848	0.8113	0.9232	0.6802	0.9993	0.9968	0.9988	4.8990	2.0000	6.0828
Brats18_TCIA13_611_1	0.8132	0.8463	0.7692	0.7655	0.8264	0.7273	0.9985	0.9939	0.9977	8.4319	4.1231	15.6205
Brats18_TCIA13_636_1	0.7595	0.9519	0.7553	0.9408	0.9492	0.7321	0.9918	0.9924	0.9771	2.2361	3.0000	9.4340
Brats18_UAB_3446_1	0.8749	0.9538	0.5233	0.9134	0.9474	0.6186	0.9987	0.9966	0.9881	8.8711	2.0000	13.1530
Brats18_UAB_3448_1	0.9041	0.9176	0.8436	0.9789	0.9605	0.9495	0.9985	0.9967	0.9970	1.7321	2.2361	6.0828
Brats18_UAB_3454_1	0.8375	0.9424	0.7220	0.9126	0.9226	0.7053	0.9973	0.9958	0.9943	20.2731	3.3166	17.4929
Brats18_UAB_3455_1	0.8965	0.9060	0.7368	0.9732	0.9485	0.8449	0.9990	0.9918	0.9950	1.0000	4.0000	18.1384
Brats18_UAB_3456_1	0.9189	0.9019	0.8421	0.9447	0.9804	0.7672	0.9992	0.9925	0.9994	1.4142	3.0000	5.3852
Brats18_UAB_3490_1	0.8147	0.9122	0.7157	0.9901	0.9532	0.9643	0.9946	0.9849	0.9857	2.2361	3.1623	19.2873

Brats18_UAB_3498_1	0.8485	0.8845	0.6766	0.9615	0.9027	0.5241	0.9976	0.9900	0.9994	2.2361	3.6056	20.6700
Brats18_WashU_S036_1	0.8021	0.8774	0.8327	0.9573	0.8614	0.9593	0.9887	0.9917	0.9863	4.0000	7.8740	10.7238
Brats18_WashU_S037_1	0.8080	0.9360	0.8079	0.7546	0.9666	0.8341	0.9991	0.9932	0.9974	8.4845	2.2361	23.3666
Brats18_WashU_S041_1	0.9110	0.9495	0.9539	0.9208	0.9625	0.9827	0.9988	0.9953	0.9987	1.0000	1.7321	1.4142
Brats18_WashU_W033_1	0.8850	0.9198	0.9331	0.8622	0.9687	0.9610	0.9985	0.9860	0.9970	1.4142	2.2361	4.0000
Brats18_WashU_W047_1	0.8564	0.7448	0.6362	0.9509	0.8443	0.5530	0.9981	0.9762	0.9968	1.4142	22.2261	18.4269
Brats18_WashU_W053_1	0.9082	0.9076	0.8906	0.8957	0.9509	0.8674	0.9995	0.9971	0.9992	1.4142	2.8284	4.2426
Average	0.8579	0.9126	0.8577	0.8803	0.9198	0.8589	0.9971	0.9937	0.9964	2.9829	4.1000	8.3871
Stdev	0.0800	0.0376	0.0993	0.0766	0.0492	0.1168	0.0039	0.0045	0.0045	3.2682	3.3357	8.7187

Table 4.5 Test results of 22 Small enhancing tumor (SET) cases in Experiment 5

Label	*Dice*			*Sensitivity*			*Specificity*			*Hausdorff95*		
	ET	WT	TC	ET	WT	TC	ET	WT	TC	ET	WT	TC
Brats18_CBICA_ABT_1	0.8835	0.9388	0.9449	0.8678	0.9692	0.9157	0.9996	0.9985	0.9998	1.0000	1.7321	1.0000
Brats18_CBICA_ALV_1	0.7313	0.9194	0.6829	0.6899	0.9409	0.6803	0.9998	0.9962	0.9996	5.1039	2.2361	11.1870
Brats18_CBICA_ANK_1	0.8064	0.9173	0.8638	0.8525	0.9048	0.8970	0.9994	0.9971	0.9994	4.3589	2.4495	5.0990
Brats18_CBICA_AQE_1	0.7010	0.6036	0.6787	0.5634	0.4560	0.5751	0.9999	0.9995	0.9996	2.0000	6.0000	3.7417
Brats18_CBICA_ARR_1	0.7975	0.9164	0.8690	0.7516	0.9339	0.9024	0.9995	0.9968	0.9988	2.0000	3.1623	4.8990
Brats18_CBICA_ATW_1	0.7799	0.8161	0.3429	0.9811	0.7573	0.8302	0.9988	0.9929	0.9878	6.7082	5.0990	27.7849
Brats18_CBICA_AUC_1	0.3757	0.7211	0.5011	0.3029	0.6230	0.3355	0.9967	0.9967	0.9999	4.8990	8.6948	7.3485
Brats18_CBICA_AZA_1	0.6850	0.5912	0.6876	0.6627	0.4280	0.6022	0.9988	0.9991	0.9983	5.7446	13.9642	5.0000
Brats18_TCIA03_216_1	0.7993	0.9670	0.8704	0.7905	0.9562	0.9547	0.9988	0.9989	0.9953	1.4142	1.4142	13.4536
Brats18_TCIA09_248_1	1.0000	0.9366	0.7248	1.0000	0.8900	0.5929	1.0000	0.9992	0.9983	N.A.	3.1623	7.8740
Brats18_TCIA10_195_1	0.0000	0.9425	0.7288	N.A.	0.9187	0.6058	1.0000	0.9946	0.9952	N.A.	3.6056	13.0000
Brats18_TCIA10_311_1	0.2820	0.7512	0.2669	0.1671	0.6798	0.1542	1.0000	0.9992	1.0000	7.9370	4.1231	7.8740
Brats18_TCIA10_609_1	0.0000	0.9565	0.8798	0.0000	0.9427	0.8310	0.9995	0.9976	0.9974	61.1065	2.0000	6.3246
Brats18_TCIA11_612_1	0.0000	0.9011	0.7461	0.0000	0.8395	0.7176	1.0000	0.9996	0.9983	N.A.	3.1623	5.0000
Brats18_TCIA12_613_1	1.0000	0.5139	0.0000	1.0000	0.3776	0.0000	1.0000	0.9998	1.0000	N.A.	5.3852	N.A.
Brats18_TCIA13_617_1	0.0780	0.7602	0.5928	0.0406	0.6307	0.4251	1.0000	0.9972	0.9995	21.7647	11.5758	11.1803
Brats18_TCIA13_638_1	0.4052	0.8390	0.3671	0.2739	0.7269	0.2253	0.9991	0.9984	0.9995	13.3417	13.6748	20.8806
Brats18_TCIA13_646_1	0.0000	0.9525	0.0037	N.A.	0.9540	0.0019	0.9999	0.9992	0.9999	N.A.	1.4142	21.7371
Brats18_TCIA13_652_1	0.0000	0.9089	0.0165	0.0000	0.8423	0.0088	1.0000	0.9996	1.0000	N.A.	3.4641	30.6176
Brats18_UAB_3449_1	0.7967	0.8784	0.3236	0.7642	0.9767	0.7665	0.9992	0.9855	0.9787	7.2801	3.7417	19.5192
Brats18_UAB_3499_1	0.7185	0.8520	0.4963	0.6442	0.9312	0.3396	0.9993	0.9904	0.9996	3.0000	21.9727	21.4709
Brats18_WashU_W038_1	0.8566	0.9510	0.9098	0.8076	0.9547	0.9786	0.9995	0.9985	0.9986	1.4142	1.4142	3.3166
Average	0.5317	0.8425	0.5681	0.5580	0.8015	0.5609	0.9994	0.9970	0.9974	9.3171	5.6113	11.8242
Stdev	0.3695	0.1319	0.3043	0.3616	0.1903	0.3269	0.0007	0.0035	0.0050	14.7915	5.2676	8.5675

The consistency of the performance is also related to the reproducibility of the system in processing individual patient cases. For a given patient case, a functional system should be able to reproduce similar results, if not the same, each time after the system is re-trained from the initial state. To test the reproducibility of the proposed system, six patient cases of the 66 cases in the test set are chosen to present the scores and the images. The information of the six patient cases is shown in Table 4.6.

The six patient cases are sorted according to their *Dice* scores of ET areas. The *Dice* scores of Case 1 and Case 2 in Table 4.7 and Table 4.8 are excellent. It should, however, be noted that such scores are the most frequently appearing in testing the 66 cases, as indicated by the values of *Dice* - mode, shown in Table 4.2. For these patient cases, the ten experiments have generated almost the same scores, with standard deviations less than 2.6% for *Dice* scores of ET, WT and TC. The *Dice* score of Case 3 and Case 4 are very close to the mean scores in Table 4.2, and they are among the minority of the 66 cases. Moreover, those of Case 5 and Case 6 are rare and very low. The results of the last 4 cases obtained in the 10 experiments are also quite consistent. A similar level of consistency has also been observed in the test results of the other individual cases, regardless the levels of their *Dice* scores. One can conclude that the proposed CNN system has a very good reproducibility of the results to the same inputs.

Table 4.6 Information of six patient cases

Cases	Patient ID	Scores	Number of voxels of ET areas*
1	Brats18_MDA_922_1	Table 4.7	22839
2	Brats18_CBICA_ABT_1	Table 4.8	8057
3	Brats18_MDA_907_1	Table 4.9	19296
4	Brats18_CBICA_AZA_1	Table 4.10	8859
5	Brats18_CBICA_ALT_1	Table 4.11	51683
6	Brats18_TCIA13_638_1	Table 4.12	4356

* The number of voxels is the average of those counted in the ten experiments.

Table 4.7 Test results of Case 1 obtained in the ten experiments

Exp.	*Dice*			*Sensitivity*		
	ET	WT	TC	ET	WT	TC
No. 1	0.8822	0.9302	0.9232	0.9428	0.9030	0.9575
No. 2	0.8854	0.9288	0.9249	0.9270	0.9087	0.9483
No. 3	0.8862	0.9291	0.9294	0.9269	0.8986	0.9548
No. 4	0.8861	0.9336	0.8828	0.9405	0.9153	0.9580
No. 5	0.8923	0.9300	0.9442	0.9272	0.9060	0.9508
No. 6	0.8899	0.9322	0.8725	0.9306	0.9095	0.9563
No. 7	0.8905	0.9293	0.9040	0.9255	0.9007	0.9515
No. 8	0.8809	0.9312	0.9079	0.9327	0.9083	0.9538
No. 9	0.8933	0.9289	0.8819	0.9407	0.9018	0.9632
No. 10	0.8867	0.9305	0.9128	0.9333	0.9024	0.9591
AVERAGE	0.8873	0.9304	0.9084	0.9327	0.9054	0.9553
STDEV	0.0041	0.0016	0.0233	0.0065	0.0050	0.0044

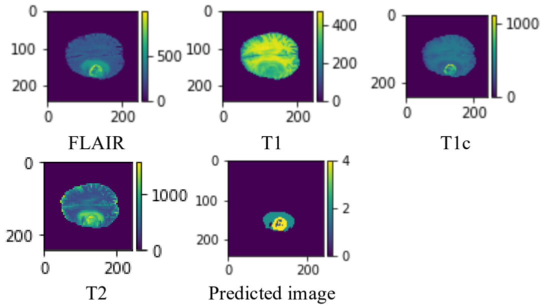


Figure 4.7 Four input slices (the 96th) of Case 1 and the predicted tumor locations.

Table 4.8 Test results of Case 2 obtained in the ten experiments

Exp.	*Dice*			*Sensitivity*		
	ET	WT	TC	ET	WT	TC
No. 1	0.8698	0.9184	0.9337	0.8501	0.9622	0.9002
No. 2	0.8781	0.9225	0.9409	0.8956	0.9670	0.9206
No. 3	0.8892	0.9276	0.9348	0.9086	0.9580	0.9313
No. 4	0.8880	0.9273	0.9331	0.9062	0.9737	0.9168
No. 5	0.8835	0.9388	0.9449	0.8678	0.9692	0.9157
No. 6	0.8824	0.9163	0.9441	0.9063	0.9739	0.9087
No. 7	0.8845	0.9340	0.9228	0.8693	0.9643	0.9294
No. 8	0.8887	0.9249	0.9421	0.9136	0.9657	0.9141
No. 9	0.8895	0.9333	0.9412	0.8974	0.9700	0.9151
No. 10	0.8709	0.9292	0.9545	0.9336	0.9703	0.9490
AVERAGE	0.8824	0.9272	0.9392	0.8949	0.9674	0.9201
STDEV	0.0073	0.0070	0.0086	0.0252	0.0050	0.0136

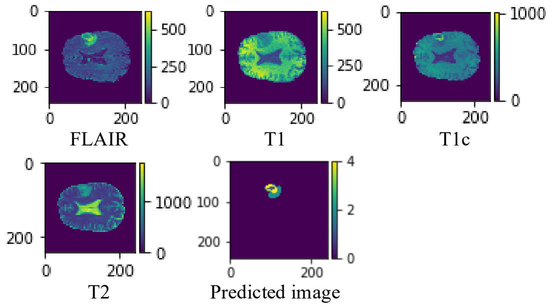


Figure 4.8 Four input slices (the 83rd) of Case 2 and the predicted tumor locations.

Table 4.9 Test results of Case 3 obtained in the ten experiments

Exp.	*Dice*			*Sensitivity*		
	ET	WT	TC	ET	WT	TC
No. 1	0.7268	0.9016	0.8291	0.6766	0.8499	0.7757
No. 2	0.7356	0.8827	0.8533	0.6775	0.8123	0.8274
No. 3	0.7746	0.8820	0.8668	0.7305	0.8099	0.8285
No. 4	0.7470	0.8917	0.8632	0.7173	0.8331	0.8494
No. 5	0.7557	0.8877	0.8790	0.7005	0.8199	0.8445
No. 6	0.7309	0.8836	0.8573	0.6738	0.8183	0.8114
No. 7	0.7592	0.8787	0.8550	0.7340	0.8072	0.8200
No. 8	0.7530	0.9029	0.8701	0.7292	0.8538	0.8749
No. 9	0.7641	0.8977	0.8585	0.7130	0.8409	0.8102
No. 10	0.7224	0.8860	0.8582	0.6794	0.8222	0.8405
AVERAGE	0.7469	0.8895	0.8590	0.7032	0.8268	0.8282
STDEV	0.0174	0.0086	0.0131	0.0247	0.0168	0.0268

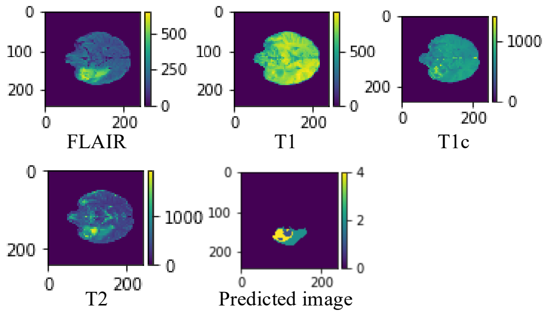


Figure 4.9 Four input slices (the 53rd) of Case 3 and the predicted tumor locations.

Table 4.10 Test results of Case 4 obtained in the ten experiments

Exp.	*Dice*			*Sensitivity*		
	ET	WT	TC	ET	WT	TC
No. 1	0.6374	0.6004	0.6855	0.6576	0.4391	0.7022
No. 2	0.6401	0.5968	0.7414	0.6387	0.4379	0.7344
No. 3	0.6575	0.5959	0.7575	0.6949	0.4355	0.7679
No. 4	0.6422	0.5839	0.7476	0.6508	0.4208	0.6955
No. 5	0.6850	0.5912	0.6876	0.6627	0.4280	0.6022
No. 6	0.6661	0.6014	0.7525	0.7074	0.4472	0.7440
No. 7	0.6906	0.6024	0.7610	0.7155	0.4427	0.7165
No. 8	0.6499	0.6015	0.7456	0.6846	0.4415	0.7667
No. 9	0.6448	0.6299	0.6868	0.7343	0.4813	0.8207
No. 10	0.6596	0.5651	0.7340	0.6469	0.3999	0.6675
AVERAGE	0.6573	0.5968	0.7300	0.6793	0.4374	0.7218
STDEV	0.0185	0.0163	0.0308	0.0327	0.0207	0.0605

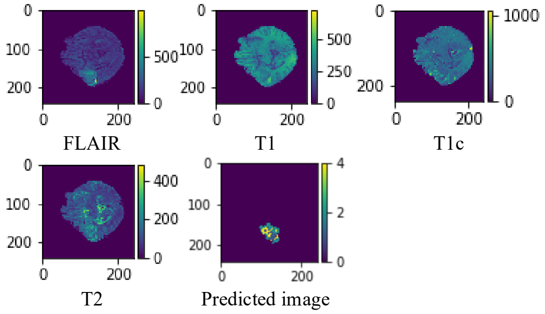


Figure 4.10 Four input slices (the 57th) of Case 4 and the predicted tumor locations.

Table 4.11 Test results of Case 5 obtained in the ten experiments

Exp.	*Dice*			*Sensitivity*		
	ET	WT	TC	ET	WT	TC
No. 1	0.4617	0.8881	0.8794	0.8505	0.8398	0.8455
No. 2	0.4763	0.8934	0.8895	0.8254	0.8568	0.8683
No. 3	0.4655	0.8936	0.8865	0.8173	0.8480	0.8555
No. 4	0.4645	0.8912	0.8842	0.8244	0.8560	0.8683
No. 5	0.4452	0.8859	0.8624	0.7948	0.8352	0.8178
No. 6	0.4310	0.9072	0.9073	0.8681	0.8863	0.9199
No. 7	0.4609	0.8855	0.8664	0.8102	0.8397	0.8376
No. 8	0.4348	0.8983	0.8827	0.8381	0.8649	0.8594
No. 9	0.4031	0.8948	0.8925	0.8577	0.8686	0.8958
No. 10	0.4202	0.8913	0.8693	0.8473	0.8493	0.8302
AVERAGE	0.4463	0.8929	0.8820	0.8334	0.8544	0.8598
STDEV	0.0235	0.0064	0.0135	0.0229	0.0157	0.0305

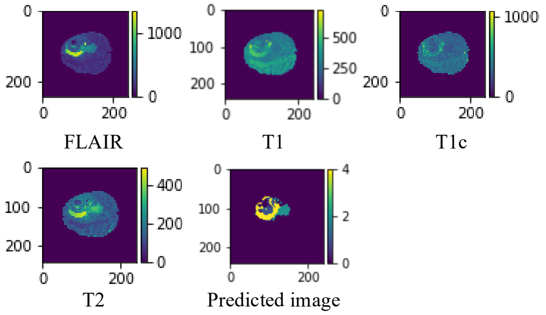


Figure 4.11 Four input slices (the 110th) of Case 5 and the predicted tumor locations.

Table 4.12 Test results of Case 6 obtained in the ten experiments

Exp.	*Dice*			*Sensitivity*		
	ET	WT	TC	ET	WT	TC
No. 1	0.2723	0.8379	0.3016	0.1661	0.7246	0.1780
No. 2	0.2590	0.8543	0.4401	0.1558	0.7500	0.2830
No. 3	0.2273	0.8564	0.3373	0.1329	0.7540	0.2037
No. 4	0.3020	0.8397	0.2801	0.1858	0.7274	0.1633
No. 5	0.4052	0.8390	0.3671	0.2739	0.7269	0.2253
No. 6	0.2133	0.8413	0.3231	0.1224	0.7311	0.1934
No. 7	0.3680	0.8581	0.3167	0.2451	0.7565	0.1888
No. 8	0.2754	0.8446	0.3112	0.1703	0.7356	0.1850
No. 9	0.3312	0.8365	0.3941	0.2048	0.7223	0.2459
No. 10	0.3145	0.8481	0.3908	0.1993	0.7407	0.2437
AVERAGE	0.2968	0.8456	0.3462	0.1856	0.7369	0.2110
STDEV	0.0602	0.0081	0.0501	0.0473	0.0127	0.0374

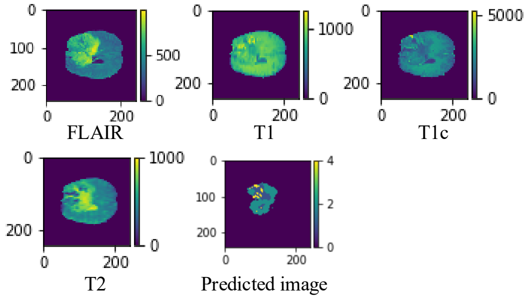


Figure 4.12 Four input slices (the 86th) of Case 6 and the predicted tumor locations.

4.3.3 Compare with State-of-the-Art Methods

The assessment results of the proposed system are compared with those produced by eight other CNN systems having moderate complexity and reported in recent years. The mean scores of Dice, Sensitivity, Specificity and Hausdorff95 of the proposed system, together with those of the eight systems, are presented in Table 4.13. The results of False Discovery Rate (FDR), False

Negative Rate (FNR), and the measures of computation complexity are found in Table 4.14. As training/testing samples play very important role in the performance evaluation, for each of the systems listed for comparison, the information about which datasets were used and whether the results were generated by CBICA Image Processing Portal is also found in the two tables.

Table 4.13 Comparison of the results – *Dice*, *Sensitivity*, *Specificity* and *Hausdorff95*

Systems	Dataset (BRATS)	Assessed by CBICA	*Dice*			*Sensitivity*			*Specificity*			*Hausdorff95*		
			ET	WT	TC	ET	WT	TC	ET	WT	TC	ET	WT	TC
Ding et al. [24]	2015	No	0.592	0.831	0.671	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
Ding et al. [27]	2015	No	0.63	0.86	0.71	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
Li et al. [23]	2017	No	0.642	0.876	0.763	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
Pereira et al. [26]	2017	**Yes**	0.733	0.895	0.798	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	5.074	5.920	8.947
Chen et al. [30]	2017	**Yes**	0.7346	0.8930	0.7388	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
Chen et al. [25]	2017	No	0.6498	0.8347	0.7321	0.8035	0.8453	0.7493	0.9994	0.9986	0.9992	30.3100	36.4000	25.5900
Zhou et al. [31]	**2018**	**Yes**	0.7525	0.8642	0.7738	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
Hu et al. [28]	**2018**	**Yes**	0.7178	0.8824	0.7481	0.8684	0.9074	0.7621	0.9947	0.9918	0.9969	5.6864	12.6069	9.6223
Proposed	**2018**	**Yes**	**0.7486**	**0.8849**	**0.7583**	**0.7797**	**0.8757**	**0.7580**	**0.9978**	**0.9949**	**0.9968**	**5.0232**	**6.4887**	**10.7027**

Table 4.14 Comparison of the results – *FDR*, *FNR*, computation complexity/volume

System	Dataset (BRATS)	False Discovery Rate			False Negative Rate			Max number of filters in a layer	Number of layers	Number of Parameters	Number of FLOPs per patient case
		ET	WT	TC	ET	WT	TC				
Ding et al. [24]	2015	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	256	≥ 29	6.66M	N.A.
Ding et al. [27]	2015	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	512	N.A.	N.A.	N.A.
Li et al. [23]	2017	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	≥ 9	N.A.	N.A.
Pereira et al. [26]	2017	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	160	≥ 12	N.A.	N.A.
Chen et al. [30]	2017	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	150	11	100K	N.A.
Chen et al. [25]	2017	0.4545	0.1756	0.2843	0.1965	0.1547	0.2507	N.A.	N.A.	10.03M	N.A.
Zhou et al. [31]	**2018**	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	50	≥ 11	Millions	N.A.
Hu et al. [28]	**2018**	0.3883	0.1413	0.2654	0.1316	0.0926	0.2379	160	≥ 13	N.A.	N.A.
Proposed	**2018**	**0.2801**	**0.1057**	**0.2414**	**0.2203**	**0.1243**	**0.2420**	**16**	**7**	**11.716K**	**21.14G**

It should be underlined that, the CNN block in the proposed system requires only 11.716K parameters for its 7 convolution layers, as shown in Table 4.14. Its computation cost is significantly lower than those reported so far, and the system yields, nevertheless, a high processing quality. One can easily see that, compared with other brain tumor segmentation systems of modest computation, the proposed system has

(i) very good *Dice* scores,

(ii) the lowest false discovery rates (*FDR*) in the detection of ET, WT and TC,

(iii) false negative rates or miss rates comparable to others, and

(iv) the results among the best reported in the detection of ET voxels in the aspects of *Dice* and *FDR*. In particular, the *FDR* of ET is 10% lower than the second best found in the list.

The excellent processing quality is mainly owing to the specifically designed CNN for brain tumor segmentation. Though it has only 7 convolution layers, the operations in each layer are made to extract critical feature information, first for the localization of the tumor areas and then for the precise classification. Furthermore, the refinement block provides another improvement, after the CNN, in the ET detection: *Dice* is increased from 72.2% to 74.9%, the false discovery rate is reduced from 32.4% to 28.0%, and Hausdorff95 is reduced from 7.3 to 5.0.

Another important item in the performance metrics is the number of floating-point operations (FLOPs) required to complete the segmentation of each patient case. With a large amount of input data, i.e., 35.712M (240×240×155×4) voxels in 3D brain MRI images in each patient case, the proposed CNN requires only 21.14G FLOPs to complete the task. This extremely small number of FLOPs results mainly from the simplicity of the proposed CNN with 11.716K parameters. Also, the pre-processing block helps to reduce more than 50% of the input data volume applied to the CNN. Because of the extremely small number of FLOPs, the proposed system requires only 2.6 seconds for testing a patient case on a NVIDIA P100 Pascal GPU with 12GB HBM2 memory.

4.4 Summary

The proposed system is evaluated by CBICA Image Processing Portal with dataset BRATS2018. It requires only 50 epochs for training and 21.14G FLOPs for testing a patient case, thanks to the very simple CNN, which has only 11716 parameters.

To test the consistency of the proposed system, ten experiments have been done, i.e., training and testing the proposed system from initial state for ten times. The histograms and boxplots of the 10 sets of test results obtained from the 10 experiments are very similar. The median *Dice* scores for enhancing tumor, whole tumor and tumor core are 0.85, 0.92 and 0.86, respectively, and their standard deviations are no more than 0.38%. In addition, the mean Dice scores are 0.75, 0.88 and 0.76, and their standard deviations are no more than 1.28%. These data demonstrate that the scores of the proposed system are very consistent and reliable.

To test the reproducibility of the proposed system, six patient cases of the 66 cases in the test set are chosen to present the scores and the images. The standard deviations of the *Dice* and *Sensitivity* obtained in the ten experiments are very low, which demonstrate that the proposed system is able to generate very close output to the sample input after retraining.

The results of the proposed system are compared with those of the seven CNN systems reported in recent years. The proposed system yields good results under the lowest computation complexity. It has: very good *Dice* scores, i.e., 0.7486, 0.8849 and 0.7583 for ET, WT and TC; the lowest false discovery rates (*FDR*) in the detection of ET, WT and TC; and the best results in the detection of ET voxels in the aspects of *Dice*, *Hausdorff95* and *FDR*.

The high performance of the proposed system owes to the efficient filtering operations in its convolution layers. The test results also confirm that the pre- and post-processing blocks efficiently contribute to the high performance.

5. Conclusion

Brain tumors cause serious problems in health. Brain tumor segmentation is to localize the tumors and to identify the tumor areas among the three types of intra-tumoral structures, namely edema, non-enhancing (solid) core/necrotic (or fluid-filled) core and enhancing core. The quality of the segmentation is critical for the diagnosis. As manual segmentation is time-consuming, it is necessary to develop automated segmentation systems to enable timely diagnosis for treatments and recovery.

The objective of the work presented in this thesis is to develop CNN systems for high-quality brain tumor segmentation at the lowest computation cost, with a view to facilitating their implementation and applications. To this end, instead of adopting a general-purpose CNN structure, one needs to design the systems specifically to extract signal features in brain tumor areas for a fine classification of brain image voxels.

To achieve the objective, a very simple and custom-designed CNN system has been proposed. It is composed of three parts, a pre-processing block, an application-specific CNN and a refinement block.

The pre-processing block is done in 2 steps to reduce the data volume applied to the CNN. The first step is to remove excessive margins in each slice of brain images. The second step is to remove the tumor-free slices. Such slices are identified by counting the percentage of background pixels in the slices and by applying simple analysis on the symmetry of the brain areas, in the aspects of pixel values inside the areas and the outlines of the areas. The pre-processing reduces more than 50% of the data volume. It helps to decrease not only the computation in the succeeding CNN, but also the risk of false positive classification of the system.

The CNN block has a very simple structure, involving 7 convolution layers. Each layer produces 16 output channels, except the last one, and the total number of parameters is 11716, which makes this block the simplest U-net-based CNN so far reported. The input data acquired under different conditions are first channel-wisely normalized to uniform the data range before the convolutions. The convolution layers are grouped into 2 parts. In the first part, the convolutions combined with max-pooling are performed to localize brain tumor areas. Two convolution modes, namely depthwise convolution and standard convolution, are performed in parallel in the first 2 layers to extract elementary features efficiently. In the second part, the convolutions combined with upsampling are to segment different tumor areas. For a fine classification of pixel-wise precision, the feature maps are modulated by the weighted local feature maps generated in the first part of the CNN.

The refinement block is to remove isolated false-positive voxels in the output maps of the CNN block. As tumors are 3D objects, true tumor voxels must appear in certain number of consecutive slices, which is used to detect false-positive voxels in the refinement. The segmentation quality is improved by the detection and removal of such false-positive voxels.

The proposed system has been trained and tested by using dataset BRATS2018, and the test results have been assessed by CBICA Image Processing Portal. Since the CNN is very simple in structure and requires only 11716 parameters, it needs only 50 epochs to complete a training process and to be ready for testing patient cases.

To evaluate the consistency of the performance, ten experiments are conducted. In each of the ten experiments, the proposed system is trained from initial state, and then all the 66 patient cases in the testing pool of BRATS2018 have been tested. The test results of the proposed system are very good. The median *Dice* scores for enhancing tumor, whole tumor and tumor core are 0.85, 0.92 and 0.86, respectively, and the mean *Dice* scores are 0.75, 0.88 and 0.76. It is important to mention that the 10 sets of test results obtained from the 10 experiments are very similar, and their statistical feature data have very small deviation. Also, for a particular patient case, the 10 experiments have given almost the same results. Hence, it has been proven that the proposed system is able to operate in a consistent manner, its results are reproducible, and thus reliable.

The performance of the proposed system has been compared with those of the CNNs of moderate complexity reported in recent years. In terms of segmentation quality, the overall performance of the propose system is comparable to the best ones so far reported. In particular, in the aspect of detecting enhancing tumor, it gives the best Dice score, the shortest Hausdorff95 distance and the lowest false discovery rate. It should, however, be underlined that the high processing quality of the proposed system has been achieved at an extremely low computation cost, as its convolution layers require only 11716 parameters, while most of segmentation systems need millions, or tens of millions, of parameters to do the same tasks.

It should be noted that the proposed system is not for general purpose. It has been designed to meet the specific needs to segment brain tumors or other kinds of tumors in medical images. In this way, the redundancy in computation can be minimized, the information density in data flow increased, and the computation efficiency/quality improved. The work presented in this thesis demonstrates that a CNN system can be made to perform a high-quality processing, at a very low computation cost, for a specific application. Hence, application-specific CNN (ASCNN) is an effective approach to lower the barrier of computation resource requirement of CNN systems to make them more implementable and applicable for general public.

References

[1] https://www.britannica.com/science/brain

[2] B. H. Menze *et al.*, "The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)," *IEEE Trans. Med. Imag.*, vol. 34, no. 10, pp. 1993-2024, Oct. 2015.

[3] P. Gibbs, D. L. Buckley, S. J. Blackband, and A. Horsman, “Tumour volume determination from MR images by morphological segmentation,” *Phys. Med. Biol.*, vol. 41, no. 11, pp. 2437-2446, 1996.

[4] T. Imtiaz, S. Rifat, S. A. Fattah and K. A. Wahid, "Automated Brain Tumor Segmentation Based on Multi-Planar Superpixel Level Features Extracted From 3D MR Images," *IEEE Access*, vol. 8, pp. 25335-25349, 2020.

[5] N. Geschwind and W. Levitsky, “Human Brain: Left-right asymmetry in temporal speech region,” *Science*, vol. 161, pp. 186–187, 1968.

[6] A. Kermi, K. Andjouh and F. Zidane, "Fully automated brain tumour segmentation system in 3D-MRI using symmetry analysis of brain and level sets," *IET Image Process.*, vol. 12, no. 11, pp. 1964-1971, 11 2018.

[7] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.

[8] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” *Proc. of International Conference on Machine Learning*, pp. 807–814, 2010.

[9] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” *Proc. of International Conference on Machine Learning*, vol. 30, p. 3, 2013.

[10] M. N. Gibbs and D. J. MacKay, “Variational gaussian process classifiers,” *IEEE Trans. Neural Networks*, vol. 11, no. 6, pp. 1458–1464, 2000.

[11] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv:1409.1556*, 2014.

[12] V. Badrinarayanan, A. Kendall and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 39, no. 12, pp. 2481-2495, 1 Dec. 2017.

[13] C. Szegedy et al., "Going deeper with convolutions," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2015, pp. 1-9.

[14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2016, pp. 2818-2826.

[15] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2016, pp. 770-778.

[16] G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, July 2017, pp. 2261-2269.

[17] Y. LeCun, Y. Bengio, and G. Hinton, "*Deep learning*," *Nature*, vol. 521, no. 7553, p. 436-444, 2015.

[18] D. Ciregan, U. Meier and J. Schmidhuber, "Multi-column deep neural networks for image classification," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3642-3649

[19] Robbins, Herbert; Monro, Sutton. "A Stochastic Approximation Method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400-407, 1951.

[20] Diederik P. Kingma, Jimmy Ba, "Adam: A method for stochastic optimization", *arXiv:1412.6980*, 2014.

[21] J. Long, E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2015, pp. 3431-3440.

[22] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Proc. MICCAI*, Nov. 2015, pp. 234–241.

[23] Haichun Li, Ao Li and Minghui Wang, "A novel end-to-end brain tumor segmentation method using improved fully convolutional networks," *Comput. Biol. Med.*, vol. 108, p. 150-160, May 2019.

[24] Y. Ding, F. Chen, Y. Zhao, Z. Wu, C. Zhang and D. Wu, "A Stacked Multi-Connection Simple Reducing Net for Brain Tumor Segmentation," *IEEE Access*, vol. 7, pp. 104011-104024, 2019.

[25] L. Chen, P. Bentley, K. Mori, K. Misawa, M. Fujiwara and D. Rueckert, "DRINet for Medical Image Segmentation," *IEEE Trans. Med. Imag.*, vol. 37, no. 11, pp. 2453-2462, Nov. 2018.

[26] S. Pereira, A. Pinto, J. Amorim, A. Ribeiro, V. Alves and C. A. Silva, "Adaptive feature recombination and recalibration for semantic segmentation with Fully Convolutional Networks," *IEEE Trans. Med. Imag.*, May 2019.

[27] Y. Ding, C. Li, Q. Yang, Z. Qin and Z. Qin, "How to Improve the Deep Residual Network to Segment Multi-Modal Brain Tumor Images," *IEEE Access*, vol. 7, pp. 152821-152831, 2019.

[28] K. Hu *et al.*, "Brain Tumor Segmentation Using Multi-Cascaded Convolutional Neural Networks and Conditional Random Field," *IEEE Access*, vol. 7, pp. 92615-92629, Jul. 2019.

[29] K. Kamnitsas, C. Ledig, V.F. Newcombe, J.P. Simpson, A.D. Kane, D.K. Menon, D. Rueckert, B. Glocker, “Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation,” *Med. Image Anal.* vol. 36, pp. 61–78, 2017.

[30] Chen, Shengcong, Ding, Changxing and Liu, Minfeng, “Dual-force convolutional neural networks for accurate brain tumor segmentation,” *Pattern Recognit.*, vol. 88, pp. 90-100, Apr. 2019.

[31] Zexun Zhou, Zhongshi He, Meifeng Shi, Jinglong Du and Dingding Chen, “3D dense connectivity network with atrous convolutional feature pyramid for brain tumor segmentation in magnetic resonance imaging of human heads,” *Comput. Biol. Med.*, vol. 121, June 2020.

[32] Tuzikov, A., Colliot, O., Bloch, I. “Evaluation of the symmetry plane in 3D MR brain images,” *Pattern Recognit. Lett.*, 2003, 24, pp. 2219–2233.

[33] S. Prima, S. Ourselin and N. Ayache, "Computation of the mid-sagittal plane in 3-D brain images," *IEEE Trans. Med. Imag.*, vol. 21, no. 2, pp. 122-138, Feb. 2002.

[34] Zhou Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600-612, April 2004.

[35] L. Nyúl and J. Udupa, “On standardizing the MR image intensity scale,” *Magn. Reson. Med.*, vol. 42, no. 6, pp. 1072–1081, 1999.

[36] L. G. Nyúl, J. K. Udupa, and X. Zhang, “New variants of a method of MRI scale standardization,” *IEEE Trans. Med. Imag.*, vol. 19, no. 2, pp. 143–150, Feb. 2000.

[37] https://www.med.upenn.edu/sbia/brats2018/data.html

[38] S. Bakas *et al.*, "Advancing The Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features", *Nature Scientific Data*, 4:170117 (2017) DOI: 10.1038/sdata.2017.117

[39] S. Bakas *et al.*, "Identifying the Best Machine Learning Algorithms for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BRATS Challenge", *arXiv preprint* arXiv:1811.02629 (2018)

[40] I. Loshchilov and F. Hutter, “Sgdr: stochastic gradient descent with restarts,” *International Conference on Learning Representations 2016*, 2016.

[41] Yoav Benjamini and Yosef Hochberg. “Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing,” *J. R. Stat. Soc. Series B Stat. Methodol.*, vol. 57, no. 1, pp. 289–300, 1995.

[42] Ralph M. Richart M. D. “Evaluation of the true false negative rate in cytology,” *Am. J. Obstet. Gynecol.*, vol. 89, no. 6, pp. 723-726, July 1964.