A CNN Based Method for Brain Tumor Detection

by

Heng Wang

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs in partial fulfillment of the requirements for the degree of

Master of Applied of Science

in

Systems and Computer Engineering

Data Science Program for Electrical and Computer Engineering
Carleton University
Ottawa, Ontario

Abstract

The objective of this thesis is to detect the tumor in the brain images. It presents a new method of brain tumor detection and localization by using image segmentation and convolution neural network. Compared with the artificial neural network, this approach reduces the complexity of the learning model and has robustness to the noise within the image.

In order to ensure the quality of the medical images, there are several image preprocessing techniques applied before tumor recognition, which include the procedure of removing the noise and non-brain tissue from the image and enhancing the contrast. By using active contour for image segmentation, the tumor area is separated from the image as its energy appears different in pixels and the feature extraction reveals the mathematical properties of the tumor.

After the tumor localization, the target regions are imported into to the CNN as inputs and CNN classifies them into different categories based on the training results from the learning procedure. This thesis uses the 4-fold cross validation for result testing. With over 80% accuracy, the CNN shows great potential in tumor detection. In addition, this thesis covers the section of how parameter settings influencing the CNN performance.

Another improvement in the thesis is to replace the ReLU function with ELU function within the non-linear layer. With the introduction of two hyperparameters, which controls the saturation for the negative value and the exponential decay, the vanishing gradient problem is alleviated and the learning speed is accelerated.

Acknowledgements

First of all, I would like to give my truly gratitude to my supervisor, Professor Peter X.Liu, for his support and instruction while I was conducting this research. He provided me serious advice and encouragement when I got stuck during the period.

After four years' working and struggling in the society, it wouldn't be an easy task when I decided to quit my previous job in Ericsson and went all the way from China to Canada to open up a new chapter of my life, which took courage and energy to reach this goal. Witnessing the unprecedented power of data era, I made a choice of perusing a master degree in data science in Carleton University. Lack of knowledge of computation and math, I found it more difficult to do a wonderful job in this field than I ever thought. Thanks to my parents, who had always been there for me when I made this decision, which in other people's view was a little crazy. Their unconditional love and sacrifice become a magic power when I am fighting for my way to success.

Also, there are special thanks to my friends: Senior Shichao Liu, Mahla and Afsoon from my research group, Professor Elodie Roullot from Systems and Computer Engineering, Professor Li from Nanchang University, have all supplemented my understanding and my work.

Finally, I express particular appreciation to my friend Wentao from Ciena during my thesis period, who spent dozens of nights working and sharing with me about the algorithm and academic details of machine learning.

Table of Contents

List of Tables

List of Figures

List of Acronyms

ANN	Artificial Neural Network
BRATS	Brain Tumor Image Segmentation
CNN	Convolution Neural Network
DM	Directional Moment
ELU	Exponential Linear Unit
GLCM	Grey Level Co-occurrence Matrix
IDM	Inverse Difference Moment
MRI	Magnetic Resonance Imaging
RBM	Restricted Boltzmann Machine
ReLU	Rectified Linear Unit
SAE	Stacked Auto Encoder
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine

1 Introduction

This chapter introduces the background and objective of the thesis and presents the literature review on the history of machine learning, which provides the details of the development of machine learning. In addition, the contribution and structure of the master thesis is also shown in this chapter.

1.1 Overview

Making the machine think like or beyond human is an eternal theme of the development of computer. In 1950, Alan Turing came up with the idea how to treat computer as intelligence, which required the computer to comprehend language, to learn, to memorize, to deduct and so on, all of which could be regarded as the branches of artificial intelligence[1,2]. Based on this, machine learning becomes one of the hottest topics because of its brilliant performance on analyzing specific tasks with exceptional efficiency, which includes the subjects of statistics, probability, complex computation, convergence theory and so on.

Deep learning is an important topic of machine learning, as it attempts to use complex or multiple dimension non-linear structure to analyze the data in high dimension[2,3]. Deep learning aims to learn the intrinsic patterns from the training dataset. With the target of letting the machine process like human, it has an effective impact on recognition on literature, image and audio etc. There are several deep learning architectures, for example, deep neural networks, convolutional neural networks, deep belief networks and recurrent neural networks, all of which have been applied to computer visualization, speech recognition, natural language processing, audio recognition and bioinformatics processing, where they have been proved excellent [4,5] .

Like other applications, deep learning in medical image processing starts with an observation, and it can be presented in intensity values, or in a more subtle way within the set of edges, regions of particular shape and so on. Research in this field attempts to make perfect representations and create models to learn from large-scale data [9].

1.2 Objective

The objective of this thesis is to detect the tumor in the brain MRI image. It brings in a new method of combining image segmentation and convolution neural network for detecting and localizing the brain tumor. The dataset used in the thesis comes from BRATS 2015 challenge, which has been focusing on the segmentation of brain tumor in MRI scans. With the expectation of effective and precise detection of brain tumor, this method will play a great role in replacing human detection on tumor within the medical images in an efficient way.

1.3 Literature Review

Deep learning is a branch of machine learning field, which in brief means to enable the computer to learn and extract the target rules and features from the training samples and test them on the new samples or make the prediction based on the gigantic amount of data through the algorithm. The origin of deep learning comes from the research and analysis of human brain operation system[2,6]. In 1981, Professor David Hubel and Torsten Wiesel, found out how human brain layer structure worked in human visual information processing. Starting from Retina and passing by low level V1, it extracted the edge features through V2 area and detected the basic structure of the target object or partial image. Towards high level V4, it recognized the whole body [9].

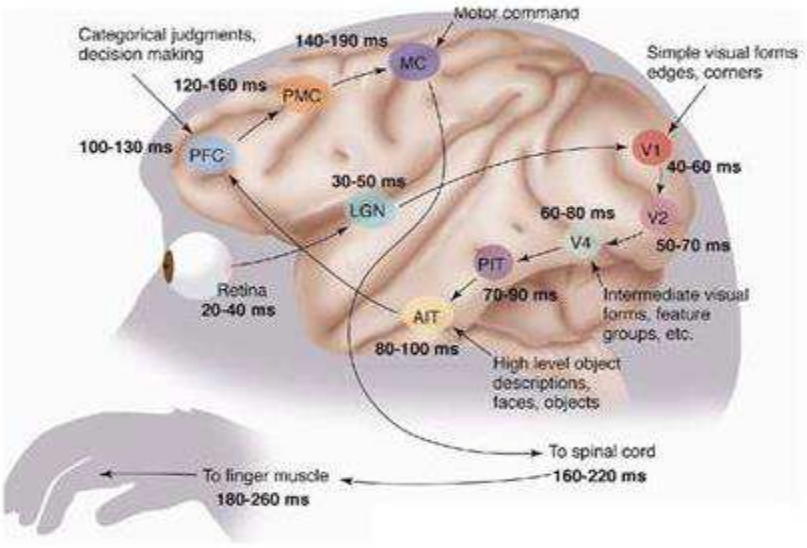


Figure 1 Human Brain Visual System

(Adapted from A. Dynamic vision: from images to face recognition by Gong S, McKenna S J, Psarrou by 2000. Imperial College Press)

This phenomenon triggers the thinking of how neural system working in human brain, whose procedure is a complex iteration of processing the input signal and extracting it. Like the previous working principle, by analyzing the primitive signals from retina, the brain makes the basic processing of the image and extracts the key information again and again, and finally recognizes the whole object [5]. Based on this understanding, the human brain is a deep learning structure and the whole cognitive procedure is also deep. This principal is the foundation of deep learning. By summarizing and organizing the features coming from low level, it extracts the features belonging to high level [9].

The development of machine learning like others experiences the peak and bottom. In 1943, Professor Warren Mcculloch and Professor Walter Pitter published the paper coming up with the idea of artificial neural network and its math model, creating the chapter of the research on artificial neural network[6]. In 1949, Professor Donald Hebb announced the theory of neural psychology, which pointed out that the learning process

of neural network took place between each neuron. In 1956, Professor Frank Rosenblatt was inspired by the thought of this theory and created the concept of perceptron, which was used to simulate the perception of human[6,7].

Perceptron is one-layer neural network, composed by linear and threshold component. The model of the perceptron is presented as:

$$Y = f(\sum_{i=0}^{\infty}(W_i X_i - \theta) \tag{1.3.1}$$

where the jump function is defined as

$$f(u) = 1, u = \sum_{i=0}^{\infty}(W_i X_i - \theta) > 0 \tag{1.3.2}$$

$$f(u) = -1, u = \sum_{i=0}^{\infty}(W_i X_i - \theta) \leq 0 \tag{1.3.3}$$

According to the functions defined, it is clear to see that the main function of perceptron is used to classify the sample data. In the early phase, the perceptron was treated to have a great potential in classification. However, in 1969, Professor Marvin Minsky and Professor Seymour Papery found out the limitation of one layer perceptron, which was not enable to solve the simple XOR problem [6,9]. In order to solve this drawback, the researchers updated this perceptron into multiple layers which introduced the convex domain to correctly classify the training samples. With the increase of hidden layers, the convex domain could be in any shape, so it solved any complex classification problem [4]. Unfortunately, there was one major disadvantage of this solution, which was how to train the weights of hidden layer. As for the nodes within each layer, they didn't have expected exported value. As a result, it was not possible to train the multi-layer

perceptron through the learning rules of classic perceptron, which pushed the research on artificial neural network into the bottom[9].

Until 1982, Professor John J.Hopfiled , together with Professor David E.Rumelhart and Professor James L. McCelland published the paper of parallel processing, renewing the interest on artificial neural network[2]. They used the error back propagation to compensate the imagination of Minsky on multi-layer network. The main idea of this algorithm was that the learning process was consisted by two steps: signal forward propagation and signal backward propagation [9]. During the period of forward propagation, the training sample was imported into input layer. After layer-by-layer processing, it was transferred to the export layer. If there was difference between the expected exported value and actual exported value, the error difference would be switched into backward propagation phase. Within the phase, the error difference was sent backward to the input layer through hidden layer and it was distributed into every unit of all the layers so that the weight was treated as the basis of the collated value of each node[9].

Since 1980, the development of machine learning was divided into two phases: shallow learning and deep learning. The first phase of machine learning came to shallow learning, dating back from 1980s. The back propagation of artificial neural network brought the new light to machine learning, raising the trend of machine learning based on the statistic model to the new level up to present[4]. It was found that by using back propagation in artificial neural network, the artificial neural network could be self-corrected on the parameter setting, which made the network fit on the training data to a large extent[2]. Based on this principal, the learning rules of the training samples could be extracted from

the huge amount of training samples. This kind of method had multiple advantages compared to the previous systems, which were based on the artificial rules.

In 1990s, there were a variety of shallow machine learning models, like Logistic Regression, Boosting, Support Vector Machine etc, which could all be treated as the models without hidden nodes, like logistic regression, or with only one hidden layer like Boosting and Support Vector Machine[19]. These kinds of machine learning models had effective success in both research theory and real application implementation.

As mentioned above, the neural network uses the backward propagation to collate the weight of the hidden layer. However, based on the gradient search to correct the weights, the issue of gradient diffusion is easy to happen, due to the fact that the solution to non-convex function is local optimization instead of global optimization [45]. To make it worse, this phenomenon may be severe when the number of network layer is increasing.

Witnessing the rapid evolution of inter-networks, the society met the data tsunami, swallowing itself among the unprecedented amount of data, therefore, how to make good use of the data and extract the valuable information became an urgent task, which pushed the trend of machine learning to another phase: deep learning[5].

In 2006, the master of machine learning field, professor Geoffrey Hinton from Toronto University and his student Ruslan published an paper on Science, bringing the deep learning in both research and application field to a new level. Within the paper, it came up with two main key ideas, one of which pointed out that the multiple artificial neural network had a strong learning ability and this would make the deep learning model extract the intrinsic rules of the training dataset and make it more convenient in classification and visualization. Another one was how to use the layer-by-layer training

method to tackle the problem of deep neural network failing to reach the optimum [2,5]. This training method took use of the results coming from the upper layer and made them as the initial settings for the succeeding layer, where the deep model could extend itself to the unsupervised learning[2,3].

Therefore, starting from 2006, the deep learning has gained consistent heat among academia, especially for the research groups in Toronto University, Montreal University, New York University, and Stanford University. Reviewing the current research on deep learning, as long as the volume of training data is huge enough plus the hidden layers are deep enough, even without the pre-training process, the deep learning can achieve good results, which reflects the intrinsic association between data and deep learning[1,3,4]. Since 2011, Google and the institution of Microsoft have successfully implemented deep neural network in audio processing, making the error rate reduced by 20% to 30%, which could be regarded as the cornerstone in the research on audio recognition[46]. In 2013, deep neural network also had a breakthrough in image recognition, reducing the previous error rate by 9% in the ImageNet evaluation. Within the same year, the pharmacy company used the deep neural network in predicting the biology activity, reaching the best result within the world[7].

Besides, in 2012, professor Andrew Ng from Stanford University, who had been focusing the research attention on deep learning algorithm in Google Brain, created a large-scale neural network with 16000 processors containing billions of nodes within his team in Google X Laboratory[7]. This neural network could train the massive random selected videos. By abundant training, this system recognized the image of cat by self-training,

which was one of the classic cases in deep learning field and gained extreme attention by every field[6].

Nowadays, the companies possessing the massive data resource like Google, Facebook, Baidu have devoted consistent resources in reaching the highest level of deep learning, based on the understanding that by using the more powerful and sophisticated deep learning model, deep learning can reveal the intrinsic rules of the training data and make the even more precise prediction[5,7].

There are three major research fields of deep learning: audio recognition, natural language processing and image recognition processing [8]. For audio recognition, in the previous long term, Gaussian Mixture Model was used to figure out the probability of each model unit. Due to the simplicity, it had a good ability in discrimination and it was convenient to implement. However, Gaussian Mixture Model is a shallow learning model in fact, so the spatial distribution of target features are not good enough. So since 2009, the experts in Microsoft Asian Institution, together with Professor Hinton, developed the deep neural network in audio recognition system, which upgraded the current technology in audio recognition into a brand new level. By implementing deep neural network, the association information between various types of features among the data was fully represented and mapped the continuous features into the high dimension features [5]. Through this dimensional mapping method, the data was fully trained in the deep neural network.

Natural Language Processing is another important application field of deep learning. In the last century, the main stream of processing natural language was based on statistic model, which was also the foundation of artificial neural network. However, in the field

of natural language, it didn't gain enough attention. In the modeling phase, neural network was implemented in dealing with the natural language. Until 2008, American NEC Research Institute migrated deep learning in natural language processing by mapping the words into one-dimensional vector space and multi-layer kernel structure with one dimension to solve the word segmentation, speech tagging, participle, word semantic meaning and structure[12,46]. They created a network model to solve these four classic problems in natural language processing field and obtained rather precise results. For image processing, it is the earliest field of implementing deep learning. Compared with the swallow learning like support vector machine, whose training method is to obtain the best linear hyper-plane to minimize the error, deep learning can discover the higher level features, like texture, shape derived from the lower level features by building the networks with multiple layers.

There are several deep learning models for image processing such as stacked auto-encoder, restricted Boltzmann machine, and convolution neural network. For stacked auto-encoder, it is a special type of two-layer neural network that learns a compressed representation of the inputs by minimizing the reconstruction error between the input and the output. Compared with a single-layer auto-encoder, when stacking auto-encoders into multiple layers, its representational power is improved greatly [20]. The main drawback of auto-encoder is that it can easily memorize the training data so that it has the tendency of overfitting.

A restricted Boltzmann machine is a network made up by symmetrically-coupled binary random units, which means the RBM is a fully connected and undirected network. It can be regarded as a generative stochastic artificial neural network that can learn a probability

distribution over the inputs[10]. The training purpose of restricted Boltzmann machines is to maximize the product of probabilities assigned to the training set. The asset of this model is that it can encode any distribution from the training samples. Moreover, it can reveal the higher-order correlations among the data by the usage of the hidden unit. The primary disadvantage of RBM is that the training procedure is very tricky to learn well. For the algorithm used in RBM, contrastive divergence, it requires the sampling from a Monte Carlo Markov Chain and it needs more attention during the training phase[10]. For the models of SAE and RBM, the inputs are always in vector form. However, concerning medical images, the structural information among neighboring pixels is also vital for image analysis. So if the image is transformed into vector, it will inevitably destroy the valuable information. As a result, the convolutional neural network is designed to better utilize spatial and configuration information within the image[10,20]. The architecture of convolution neural network is based on the inspiration of Professor Hube and Professor Wiesel on modeling the animal visual system, especially simulating the simple and complex cells of animal visual skin: V1 layer and V2 layer[20]. In 1989, Professor Yann LeCuan from Toronto University and his colleagues came up with the idea of convolution neural network, which was a kind of deep neural network containing convolution layer.

In the early trials of convolution neural network, it accomplished good results on the small scale problems, however, it didn't experience any breakthrough for a longtime, because convolution neural network didn't have a good effect on large image with huge pixels[46]. Until 2012, Professor Hinton and his students applied deep convolution neural network to solve the well-known ImageNet Challenge with wonderful results, pushing

forward the deep learning in image processing field. This breakthrough mainly focused on improving the algorithm of convolution neural network by introducing the concept of weight attenuation. It could effectively reduce the weights to avoid the overfitting issue. What is more important, with the rapid development of GPU speed, it is easier to generate more training data during the training phase and make it better for the architecture to fit the training data.

Generally, a classic convolution neural network contains as least two non-linear convolution layers for training and two permanent sub-sampling layers and one fully connection layer and the number of hidden layers should be 5 or above[3]. Unlike artificial neural networks, CNN exploits three mechanisms of local receptive field, weights sharing, and subsampling that help greatly reduce the complexity of the model. Currently, deep learning is able to apprehend and recognize the simple natural images, not only improves the precision of the recognition, but also avoids the issue of costly time and effort in image processing by human. Unlike simple images, medical images have their unique property and difficulty, as well as their needs for high precision of the target pathology area. As a result, deep learning in medical image processing is filled with challenges and bright future.

1.4 Thesis Contribution

This thesis focuses on detecting the brain tumor within the MRI images, compared with the previous application in tumor detection, there are a few points of improvement within this field.

- This thesis brings in a new method of detecting and localizing the tumor by image segmentation and convolution neural network. Compared with artificial neural

network, this approach has its advantage of reducing the complexity of the learning model. Moreover, it has the robustness to the noise within the MRI images.

- Another improvement within the thesis is to replace the ReLU activation function, which is a general function within the non-linear layer. This thesis uses the ELU function, which alleviates the vanishing gradient problem by saturating smoothly to -1 for negative values. Besides, it introduces another parameter to control the exponential decay. By this approach, the learning speed is accelerated.

1.5 Thesis Structure

In order to better understand how the convolution neural network works in tumor detection, the structure of the thesis is organized as followed: Chapter 2 introduces the basics of convolution neural network. Chapter 3 presents the procedure and the methods of medical image processing in tumor detection. Chapter 4 shows the process of designing and training the convolution neural network. Chapter 5 presents the results of using CNN in brain tumor detection. Chapter 6 gives out the conclusions and discusses the future work within this field.

2 Convolution Neural Network Introduction

In this chapter, a general introduction to convolution neural network is given. The main ideas of convolution neural network are presented, the motivation for using CNN is also addressed. In order to understand the key points of convolution neural network in medical image processing, the functionality of each layer within the CNN architecture is explained.

2.1 Artificial Neural Network Introduction

Convolution neural network can be regarded as an enhancement of artificial neural network, therefore, the basics of the artificial neural network is introduced. Artificial neural network is made up of neurons, which are connected with each other and form a neural network [9]. A neuron is the unit of a neural network and the basic structure of a single neuron is shown as followed.

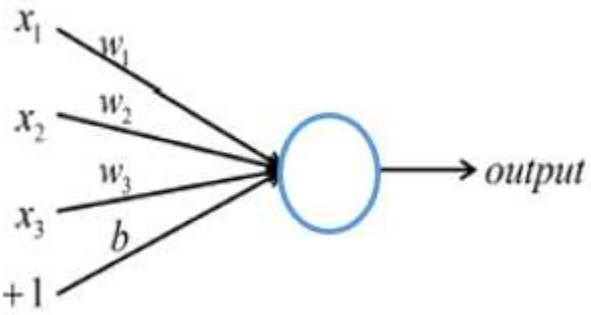


Figure 2 Basic Structure of a single neuron

According to the figure given, suppose the neuron takes x_i as the input and get the outputs based on the following computation:

$$output = f\left(\sum_{i=1}^{3}(w_i\, x_i\, + b)\right) \qquad (2.1.1)$$

where w_i are defined as weights, the b is defined as bias and $f(.)$ is a non-linear activation function .

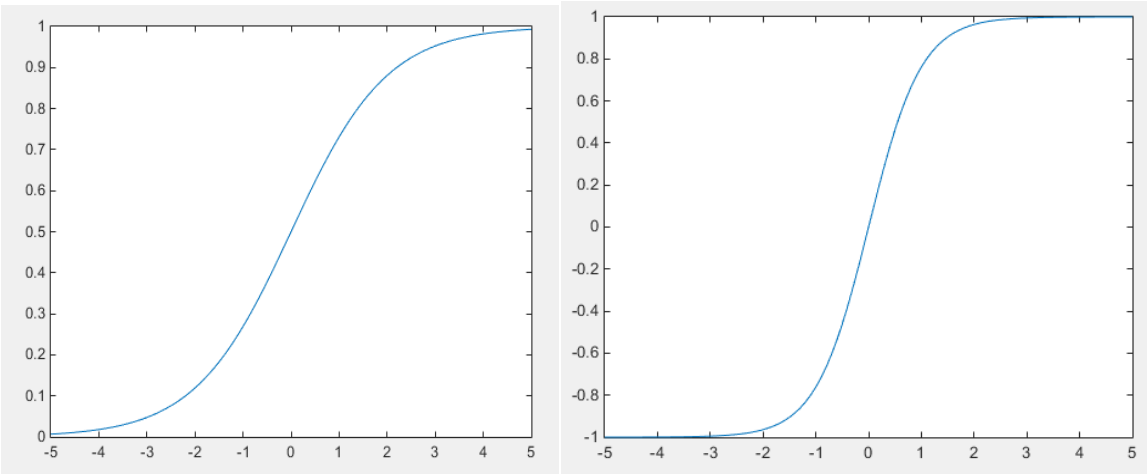
During the computation phase, every input value x_i is weighted and multiplied by an weight w_i ,then the weighted input values plus the bias b are devoted into the activation function, where this linear combination is transformed into a non-linear one. There exist several classic non-linear activation functions, among which logistic sigmoid function and hyperbolic tangent function are general choices.

Sigmoid function:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.1.2}$$

Hyperbolic tangent function:

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.1.3}$$



(a) (b)

Figure 3 Sigmoid function (a) and Hyperbolic tangent function (b)

According to the function definition, for one single neuron, the mapping relations between input and output is in fact a logistic regression [52].

A classic neural network are made up of multiple neurons and the outputs of the previous layer are the inputs of the next layer. This figure followed is an example for a feed-forward neural network, which is also known as multi-layer perceptron[34].

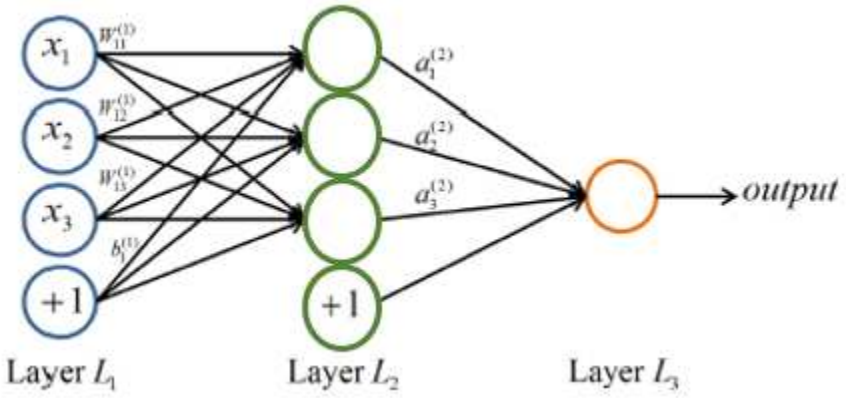


Figure 4 Basic Structure Multi-layer Perceptron

As it is shown, within the neural network, the neurons are grouped into layers. Each layer is fully connected to the subsequent one and the connections do not form cycles. It is clear to see that within each layer, there is a bias parameterb_i , which is used to compute the output of the corresponding neuron. The leftmost layer is the input layer and the rightmost layer is the output layer. The layer in the middle is defined as hidden layer, as its values can't be observed in the training set[52].

Suppose there are three layers within the neural network, $n_l = 3$, and the i –th layer is labeled as L_i . Therefore, the first layer, which is also defined as the input layer, is presented as L_1 , the second layer as L_2 and the third (output) layer as L_3 . The parameters of the model are defined as $(\mathrm{W}, \mathrm{b}) = (W^{(1)}, b^{(1)}, W^{(2)}, b^{(1)})$, where $b_i^{(l)}$is the bias to unit i in layer $l + 1$ and $W_{ij}^{(l)}$ denotes the weight value to the connection between

the unit j in layer l and the unit i in layer $l+1$. Enhance, $W^{(1)} \in R^{3\times3}$, $W^{(2)} \in R^{1\times3}$, $b^{(1)} \in R^3$ and $b^{(2)} \in R^1$ are defined in this case.

In the initial step, the activations of the hidden nodes are computed as:

$$a_1^{(2)} = f(W_{11}^{(1)} x_1 + W_{12}^{(1)} x_2 + W_{13}^{(1)} x_3 + b_1^{(1)}) \quad (2.1.4)$$

$$a_2^{(2)} = f\left(W_{21}^{(1)} x_1 + W_{22}^{(1)} x_2 + W_{23}^{(1)} x_3 + b_2^{(1)}\right) \quad (2.1.5)$$

$$a_3^{(2)} = f\left(W_{31}^{(1)} x_1 + W_{32}^{(1)} x_2 + W_{33}^{(1)} x_3 + b_3^{(1)}\right) \quad (2.1.6)$$

where $a_i^{(l)}$ is defined as the activation of unit i in layer l, and $f(.)$ is a non-linear activation function. The activations of the hidden units are then devoted into the next layer, which takes all these activations as the inputs. Therefore, the output of the last layer is computed as:

$$output = a^{(3)} = \sigma\left(W^{(2)} a^{(2)} + b^{(2)} W_{12}^{(2)} a_2^{(2)}\right)$$

$$= \sigma\left(W^{(2)} + f\left(W^{(1)} x + b^{(1)}\right) + b^{(2)}\right) \quad (2.1.7)$$

where σ means that the activation function for the output layer could be different than the activation function in the hidden units.

The whole process is called forward propagation, based on the fact that the inputs are forwarded through the network [14]. There is one thing to stress that the non-linear activation function is a must within the network, which transforms the original linear combination into a non-linear one.

2.2 Convolution Neural Network Introduction

2.2.1 CNN Basic Structure

Convolutional neural network is similar to artificial neural network, as both of them are made up of self-optimized neurons, which are imported by inputs and perform a non-

linear transformation [15]. Compared with the artificial neural network, convolution neural network is widely used in pattern recognition on images, for it encodes image-specific features into the network architecture, making the network more suitable for image-based feature learning[17].

There are five basic elements within the convolution neural network: input layer, convolutional layer, non-linear layer, pooling layer and fully connected layer. The basic functionality of a convolution neural network can be put into five key points [47].

Point 1: The input layer holds the pixel values of the image.

Point 2: The convolutional layer carries the convolution calculation of the inputs.

Point 3: The non-linear layer is used to apply non-linear transformation of inputs produced by the previous layer.

Point 4. The pooling layer performs sampling of the given inputs, reducing the number of parameters involved in the activation.

Point 5. The fully connected layer shows the class scores from the activations and they are used for classification.

It is easy to notice that the creation and optimization of this model is a little difficult to understand. In order to tackle this issue, the principals of convolution neural network will be explained in the succeeding part.

2.2.2 Convolution Layer

The convolution layer is the foremost part of convolutional neural network, which does the heavy calculation of convolution neural network operation[20]. The convolution neural network extracts different features of the inputs and the convolution layer extracts low-level features of the image, such as edges, lines, and corners. The key points of the

convolution layers is the usage of learnable kernels. The kernels are generally small in spatial dimensionality, but they can spread along the entire input[13,16]. When an input comes into a convolutional layer, the convolution layer convolves each filter across the input and then it produces a 2 dimensional activation map. Every kernel has its own activation map, which will be stacked along the depth later. As a result, it is necessary to stress that the depth of the filter should be the same as the depth of the input [22,23]. The figure below visualizes the classic operation of the kernel within the convolution layer. After extraction of the target part from the input, the kernel passes through the entire vector. After gliding through the input, the output of convolution calculation is the scalar product of each value in the kernel.

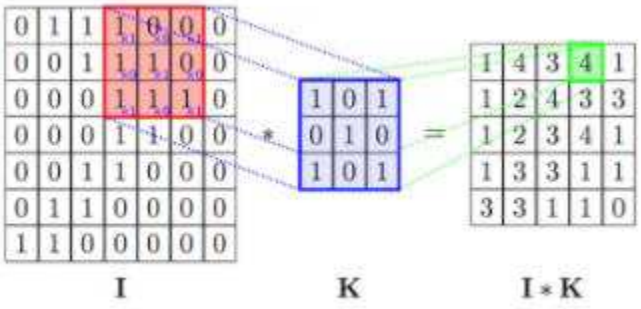


Figure 5 An example of Kernel Calculation within Convolution Layer

(Adapted from Neural network design by Demuth H B, Beale M H, De Jess O, et al. by 2014 Martin Hagan)

Generally, the kernel starts from the top left corner of the image. Hypothetically, there is an input with 32 × 32 array of pixel values and the kernel covers the area of 5 × 5. As the filter is sliding around the input image, it multiplies the values in the kernel with the original pixel values of the image and then the multiplication is summed up and every part of the input volume produces its own number. After the kernel passing through all

the parts of the image, the output of the convolution operation is an array with a 28 × 28. This array is known as the feature map. Compared with artificial neural network, the convolutional layer shows great ability in reducing the complexity of the model by using kernel.

When designing the convolution layer, there are three hyper-parameters needing consideration: the depth, the stride and setting zero-padding[21].

The depth of the output volume corresponds to the number of the kernels used for convolution and each of these kernels makes up its own feature map of the image input. By reducing this hyper parameter, it can minimize the total number of neurons within the network. However, it may decrease the performance of the convolution neural network on the pattern recognition [42].

The stride illustrates the steps of kernel sliding through the input volume. If the stride is set as 1, then the filters will move one pixel at a time. If the stride is set as 2, then the filters will jump 2 pixels at a time during the sliding period, which will produce smaller output volumes. It is clear to see that if the stride is set to a greater number, it will reduce the amount of overlapping area and produce lower spatial dimension outputs [42].

In general cases, it is convenient to fill zeros around the border of the input, which is called zeros-padding. The introduction of zero padding gives further control to the spatial size of the output volumes.

When using these parameters, it changes the spatial dimensionality of the convolutional layers output [42]. The formula followed gives the details of this change:

$$\frac{(V - R) + 2Z}{S + 1} \qquad (2.2.1)$$

where V represents the input volume size, R represents the receptive field size , Z is the amount of zero padding set and S represents the stride.

It is necessary to notice that if the calculation from this equation is not equal to an integer, then the stride needs to be altered to meet this expectation, or the neuron is not able to fit for the given input.

Despite the effort of these methods, in some cases, the model is still enormous as some images may have multiple dimension. For this consideration, parameter sharing is used to reduce the overall number of parameters within the convolutional layer. The idea of parameter sharing is based on an assumption that if the feature of one region is useful to compute at a set spatial region, then it is likely to be useful in another region. Within the convolution layer, each activation map within the output volume is set as the same weights and bias, so there is a huge decrease on the number of the parameters produced by the convolutional layer. Based on this theory, during the phase of backpropagation, each neuron in the output represents the overall gradient so that only a small group of weights need to be updated instead of every single one [42].

2.2.3 Non-linear Layer

As mentioned, the convolution neural network applies a non-linear transformation on the input, whose purpose is to identify the features within each hidden layer. In artificial neural network, the non-linear transformation function is sigmoid or hyperbolic tangent. However, for image processing, if the sparsity of data is more, the result will be better. Based on this understanding, rectified linear units is often used as the non-linear transformation.

For rectified linear unit, it implements a function of $y = \max(x, 0)$, so the output is in the same size as the input. Rectified linear unit increases the non-linear properties of the decision function and it has no negative effect on the receptive fields of the convolution layer. Compared to other non-linear functions, the training speed of the rectified linear unit is much faster. The figure followed sets an example of the rectified linear units.

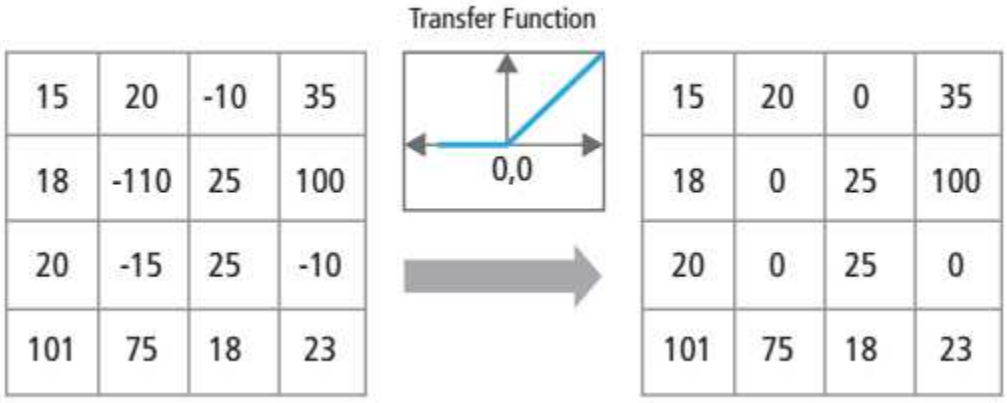


Figure 6 An Example of Rectified Linear Unit Transformation

(Adapted from Neural network design by Demuth H B, Beale M H, De Jess O, et al. by 2014 Martin Hagan)

2.2.4 Pooling Layer

After the operation of convolution layer, the data comes into the pooling layer. The major purpose of the pooling layer is also to reduce the dimensionality, the number of the parameters as well as the computational complexity. Besides, it helps to make the features robust against noise and distortion. The pooling layer operates on each feature map of the input and scales its dimensionality by using the function defined. Generally, there are two classic pooling functions, which are max pooling and average pooling. The figure followed illustrates the operations of both pooling methods.

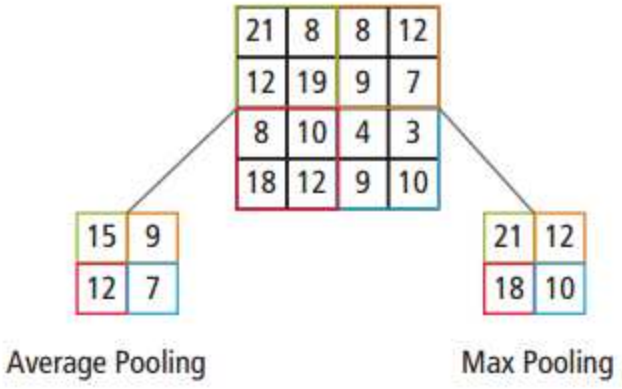


Figure 7 Two Classic Methods for Pooling

(Adapted from Neural network design by Demuth H B, Beale M H, De Jess O, et al. by 2014 Martin Hagan)

According to the reference cases, the pooling layer applies the max pooling function with a 2 × 2 kernel and a stride of 2 along the spatial dimensions of the input. It is based on the concern of the destructive functionality of pooling layer. By this operation, it reduces the feature map down to 25% of the original size, while it still maintains the depth volume to its standard size. In addition, it allows the layer to pass through the entire spatial dimensionality of the input with the overlapping area to be utilized. If the stride is set to 3 with a kernel size set to 3, it will effectively decrease the performance of the model.

2.2.5 Fully Connected Layer

After several repeats of the previous layers, the data comes to the final layer of the convolution neural network, which is the fully connected layer. Within the fully connected layer, the neurons are directly connected to the neurons in the two adjacent layers. The aim of this layer is to sum up the weights of the features coming from the previous layers and indicates the probability of each class. For example, if there is a convolution neural network for gender classification, and the output vector is a

probability of [0.7, 0.3], it means there is 70% probability of male gender and 30% for female gender.

Until this, the functionality of each layer within the convolution neural network has been explained. A classic convolution neural network basically contains two parts. One part is several repeats of convolution layer, non-linear layer, pooling layer. The purpose of this part is to reduce the dimensionality of the input volume. Another part is the fully connected layer, following the previous repeated layers, and the softmax layer, where the cross-entropy loss is optimized [20]. A softmax layer is used to convert any vector of real numbers into a vector of probabilities [20], which will be introduced in the next two chapter.

2.3 Summary

Compared with the artificial neural network, the convolution neural network takes a lead in image analysis. First, it has robustness to noise and distortions such as change in shape, complex lighting conditions, horizontal and vertical shifts of the object, and so on, which are the common issues of medical images. Second, it requires fewer memory for operation by convolution operation and local parameter sharing. Based on this theory, convolution neural network can be seen as an effective method in medical image processing, which will be shown in the next chapters.

3 Brain Tumor Image Processing

This chapter gives the methods of medical image processing involved in brain tumor detection, which include the activities of image pre-processing, image segmentation and the feature extraction from medical image. The methods of image pre-processing help to improve the quality of the images. Image segmentation and feature extraction are regarded as the important steps for tumor detection and localization.

3.1 Proposed Solution Overview

As it is known, tumor is regarded as abnormal growing cells in the body. There are two types of brain tumor: benign and malignant. The benign brain tumor has a uniformity in structure and does not contain cancer cells, while the malignant tumor is the opposite, with non-uniformity structure as well as containing cancer cells[24].

In clinical field, low-grade I and II glioma are considered as the benign tumors, and the high-grade III and IV glioma are treated as the malignant brain tumors[25,26]. For each type of the tumors, there are different medical solutions, as a result, in this thesis, the target goal not only focuses on finding out this abnormal area within the brain images, but also needs to recognize whether it is benign or malignant for future treatment. An example of different types of the tissue is shown as followed, where on the left side is the healthy brain MRI image, the right side is the image with malignant tumor and in the middle is the image with benign tumor.

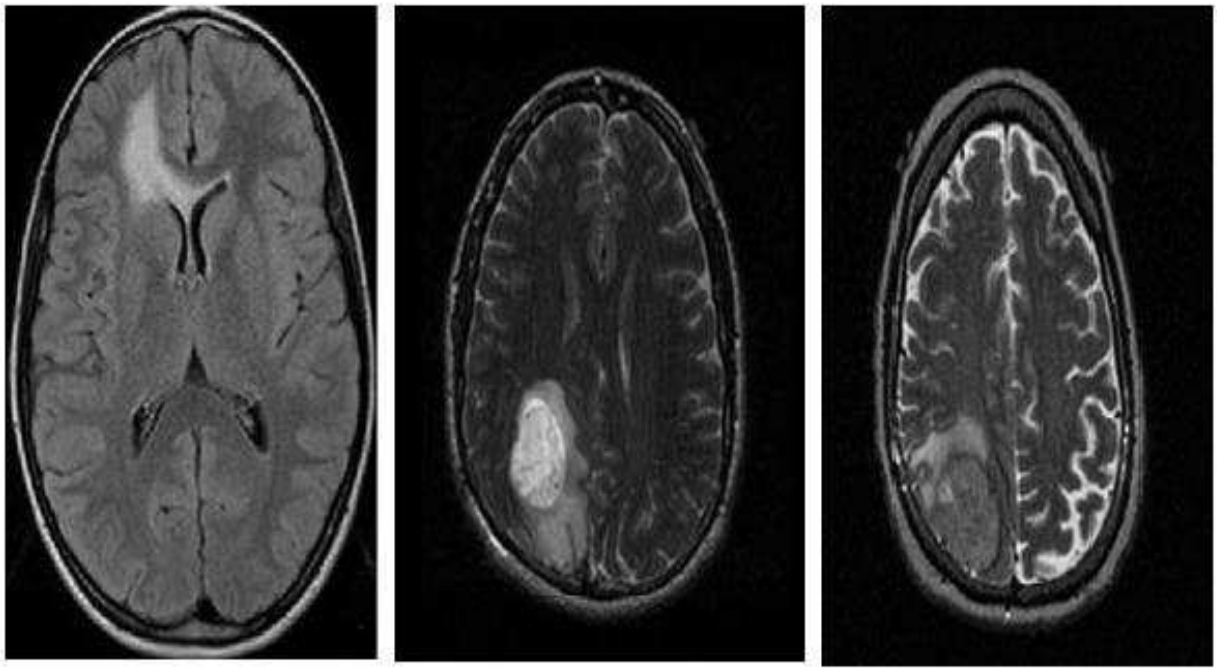


Figure 8 An Example of Different Types of Brain Tissue

In order to detect the tumor from medical images, before importing the images into convolution neural network, several methods need to be implemented to achieve better results, as the original medical image has the problems like noise, low contrast, bad resolution and so on [26]. For this consideration, the image preprocessing methods like smoothing, skull stripping, and segmentation methods like edge detection and morphological operation should be employed. These approaches not only minimize the impact of the drawbacks of the modality, but also extract the valuable area of the medical image, which is called region of interest, as it contains the most important information of the image for future use [27].

After these phases, the extraction of the features of the target region is conducted, as it reveals the mathematical properties of different types of the tissue within the images. The texture based features and shaped based features are the major features of brain tumor, so the calculation of these features on different type of brain tissues is shown to see if there

is any difference between them in order to classify the tissue into the categories: health tissue, benign tumor and malignant tumor. Then next step is to design and train the convolution neural network and test the CNN performance. In the end, the conclusion is drawn to see whether CNN acts well on tumor detection. The figure followed illustrates the main steps of the experiment.

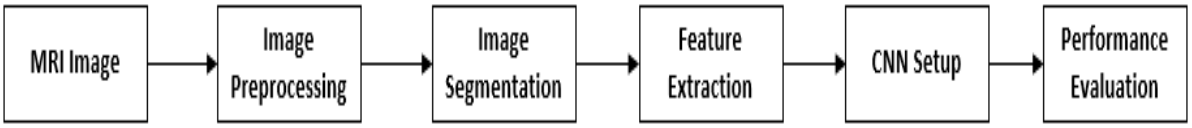


Figure 9 Flowchart of brain tumor detection

3.2 Medical Image Processing Overview

To summarize the main areas of medical image processing, generally there are four parts as the figure depicted [37]. Please notice this thesis only focuses on the part of medical image analysis.

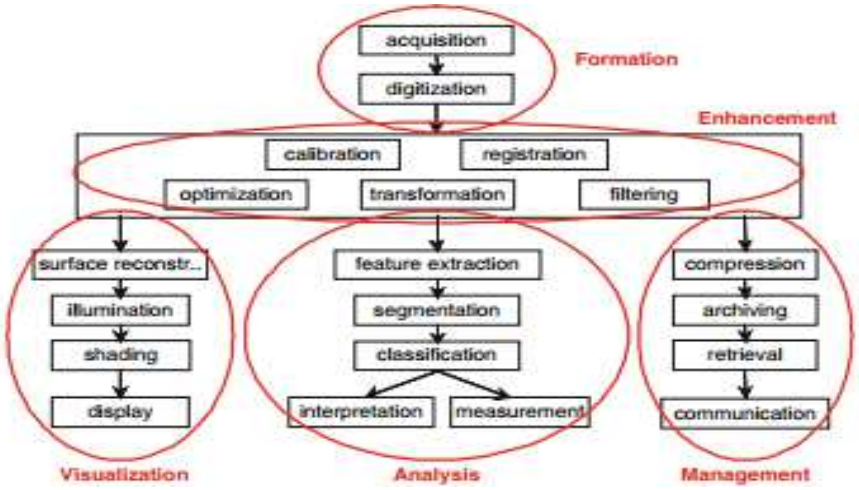


Figure 10 Flowchart of Medical Image Processing

(Adapted from The image processing handbook by Russ J C, Matey J R, Mallinckrodt A J, et al. by 1994 Computers in Physics)

There are two major methods within the medical image processing: high-level processing and low-level processing. Low-level image processing includes the operation on the raw data, pixel, edge, while high-image processing refers to the processing activities on the texture, region, object, and scene levels [38]. The definitions of these different levels are summarized in the table as followed. There is one thing to stress here that the scene level of the image is not presented in this thesis.

Table 1 Definition of Different Levels of Image

Image Level	**Functionality**
Raw data	The original image record
Pixel	Discrete pixels of the image
Edge	One dimension image structure
Texture	Two or three dimension image structure
Region	Two or three dimension image with boundary well-defined
Object	Texture or regions with certain meaning, like semantics
Scene	Spatial or temporal representation of image object

According to the definition above, there is a problem not be avoided: when it comes to the high-level image processing, due to the complex property of the medical image, it is hard to acquire the prior knowledge, which is called semantic gap [39]. Besides, there are several other issues within the medical images, which are low resolution, noise interference, low contrast, geometric deformation and so on. Such challenges make it difficult for computer interpretation of medical images. Fortunately, there are several solutions to minimize these effect with the development of medical image processing.

Image preprocessing refers to the operation of simplifying the image but retaining important information. By this means, the noise impact on the medical images is minimized. Image segmentation means extracting the anatomical structures of the images for shape analysis and visualization [41]. These methods will be introduced in details in the next subchapter.

3.3 Medical Image Preprocessing

3.3.1 Contrast Stretching

As the first step, image preprocessing plays an important role in minimizing the impact of the flaws the medical image brings in, like noise, low contrast, bad resolution. There are several aspects within the image preprocessing, which include image enhancement and image smoothing. Image enhancement refers to the image modification by changing the pixel values to improve its visual effect, while image smoothing refers to removing the useless information from the image and reserving the important one.

Both of these activities improve the visualization of the image, which can be summarized into two aspects: contrast stretching, noise filtering. Contrast stretching refers to the modification of the image pixel in order to enhance the contrast of the image, as the homogeneity of the image makes it hard to distinguish different function areas of the image. Image histogram reflects the characteristics of image so that modifying the histogram of the image would result in the changes of the image contrast. Based on this understanding, the contrast stretching is designed to stretch the narrow range to the whole of the available dynamic range [31].

There are several stretching methods, one of which is widely used for image processing, as it is easy to understand and implement: linear contrast stretching. The process of linear

contrast stretching can be put as the normalization phase, where the original pixel value is changed into the new one as the formula given:

$$DN_{new} = 255 \times \frac{DN_{ori} - DN_{min}}{DN_{max} - DN_{min}} \quad (3.3.1)$$

There is one thing to stress that it is better to make the new pixel value as the integer, as for the histogram of the grey-level, the pixel values are all integer, ranging from 0 to 255. The result of the image before and after the stretching as presented below.

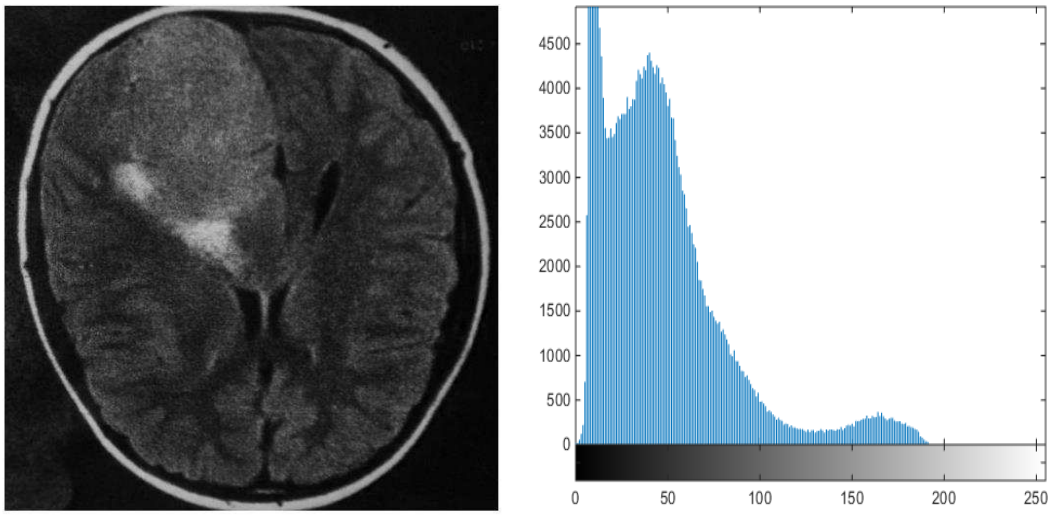


Figure 11 The Original MRI Image and Its Histogram

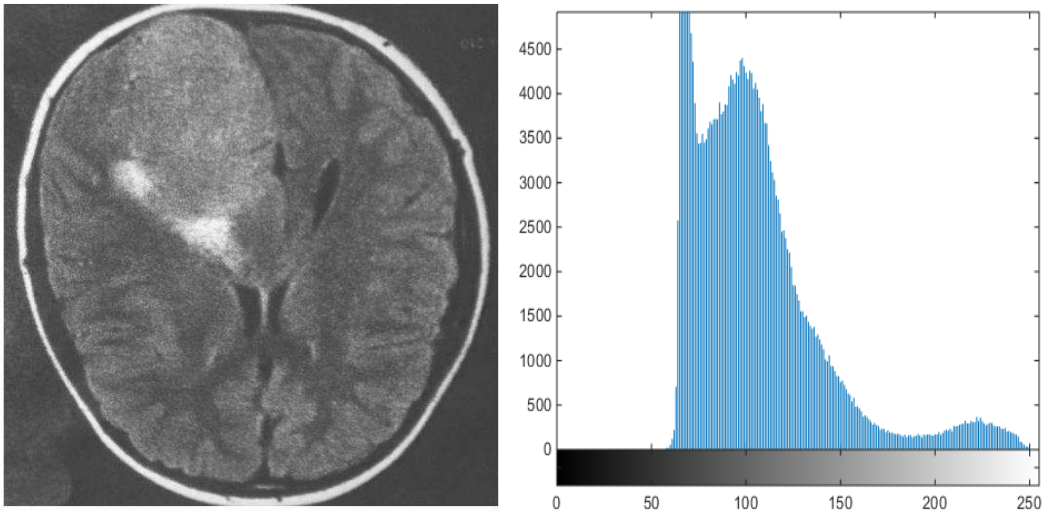


Figure 12 MRI Images and Its Histogram after Linear Stretching

3.3.2 Noise Filtering

Another common method for medical image preprocessing is noise filtering, which is applied for removing the unrelated information within the image [59]. Noise is brought into the image at the period of image acquisition and transmission. For MRI image, generally it originates from the interaction between the clinical device and the patients. The major types of the noise within the MRI image is Gaussian Noise and Impulsive Noise[69]. Gaussian Noise is statistical one with the density function of normal distribution, while Impulsive Noise is a summary of acoustic noise, containing the instantaneous sharp sounds, like impulse. The solutions to these types of noise are usually Gaussian filter and Median filter.

Gaussian filter can be interpreted as a 2 dimensional operator to blur images and remove detail and noise from the image. As it is known, it uses the kernel whose shape is similar to Gaussian distribution to reorganize the image. As image is represented as a combination of pixels, the Gaussian filter performs the convolution operation as the formula given:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3.3.2}$$

In real implementation, the operator acts on the three standard deviations from the mean, as it covers over 97% of the image, so the result is acceptable [69]. The figure followed shows the result of the Gaussian filtering.

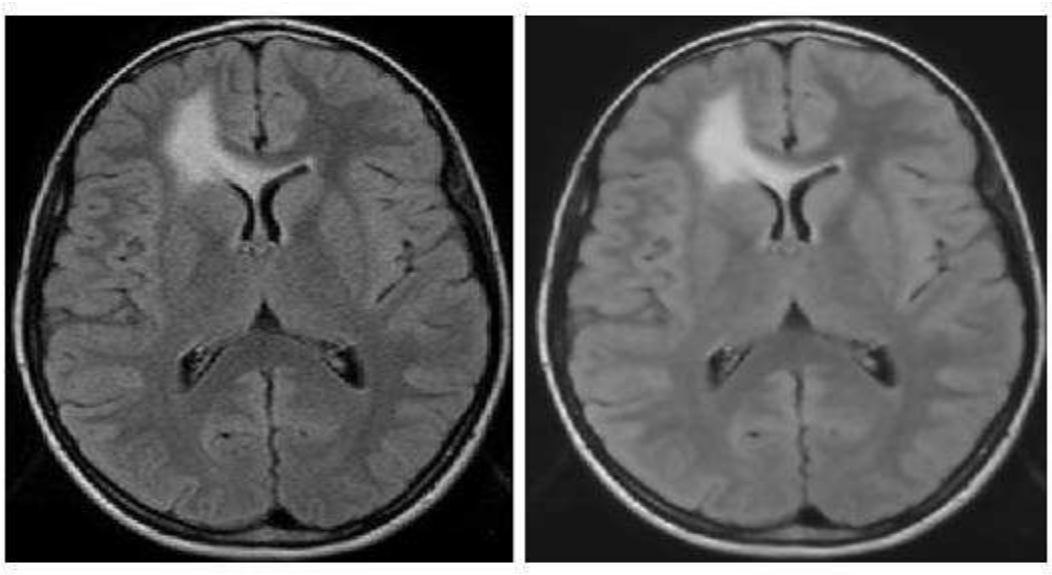


Figure 13 Gaussian Filtering on MRI Images

The idea and the operation of median filter is similar to Gaussian filter, while there is one major difference that the kernel used in this method is the median filter instead of the Gaussian one. The calculation of mean filtering of the image is to replace each pixel value in an image with the average of its neighbors including itself. In this way, impulsive noise is eliminated. The figure below illustrates the effect of the median filtering on the MRI image.

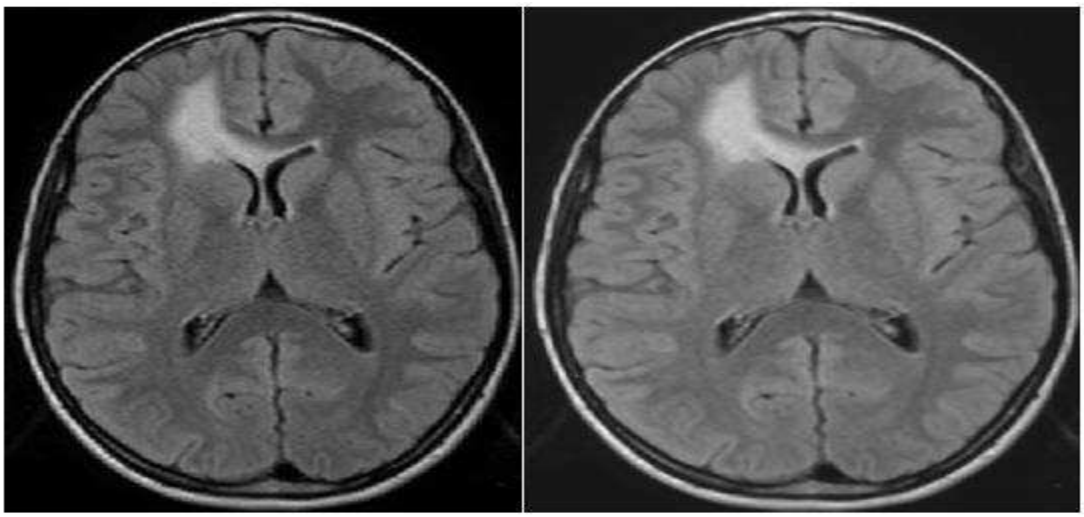


Figure 14 Median Filtering on MRI Images

3.3.3 Skull Stripping for Brain Image

Another important image preprocessing method is skull stripping, which could be regarded as an exclusive procedure in brain image processing, as for brain tumor detection, elimination all the non-brain tissue makes it easy and effective for further analysis. By skull stripping, it is possible to get rid of unrelated tissues like fat, skin, and skull from the brain images [41]. Reviewing the techniques of skull stripping, there are several approaches applied for this task, one of which uses contour, threshold segmentation, morphological operation, and histogram analysis to remove the skull from the brain images.

The skull stripping method in this thesis can be divided into two phase, first of which uses the morphological operation to reconstruct of the image; and the second phase is to apply the threshold to the segmentation from the first phase to finally obtain the skull stripped image[60]. The flowchart below summarizes the steps of the skull stripping method and each step will be introduced later.

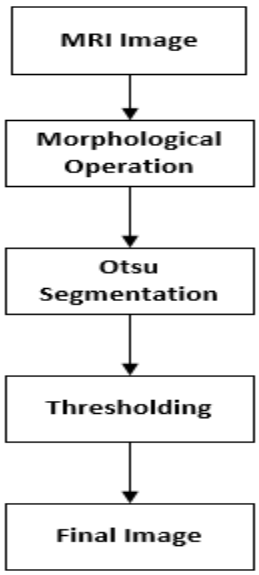


Figure 15 Flowchart of Skull Stripping Algorithm

The first step is morphological operation, referring as the non-linear transformation of the image. It includes four basic operations: Erode, Dilate, Open and Close. The basis of the morphological operation is to probe the image with a kernel, which is called a structuring element. The structuring element is a relatively small binary image filled with one or zero. It passes through all the pixels of the image and compares itself with the neighboring of pixels [61]. There are several structuring elements for morphological operation, and the 3×3 structuring element is used as the figure illustrated.

0	1	0
1	1	1
0	1	0

Figure 16 Structuring Element for Morphological Operation

The erosion of an image refers to the process of the structuring element passing through the image from the left to right and from the top to bottom. During the process, the structuring element examines the image to see whether there is an overlapping area with the structuring element. If there is, the pixel will remain as it is. If not, the pixel will be set to 0. Like erosion, dilation operation is very similar, just one major difference that the pixels of the overlapping area under the center position of will be turned to 1[62,63]. And the open operation is a combination of erosion and dilation. For opening, it starts with the erosion and ends with the dilation, while closing is quite the opposite. The purpose of opening is to open up the gap between objects connected by a thin range of pixels so that any area that left from erosion could be reconstructed to their original ones by the dilation, while the purpose of closing is to fill up the holes in the regions, with

keeping their original size at the same time. The figure followed shows the results of these four operations.

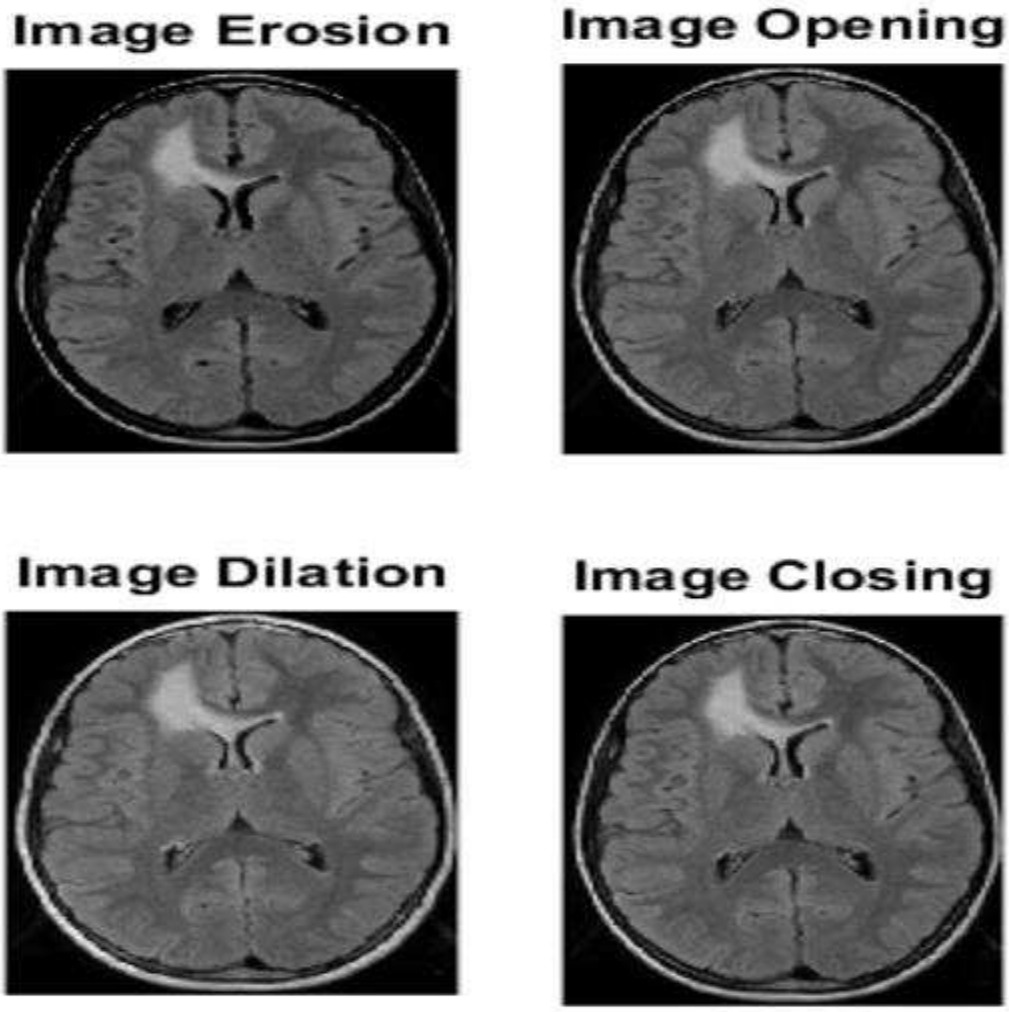


Figure 17 Morphological Operation on MRI Image

After the morphological operation, the next step is to convert the gray scale image into binary image by threshold value. In this way, the output is a binary image with 1 (white) for all pixels greater than the threshold and 0 (black) for all other pixels and the threshold is determined by Otsu's method.

$$b(x,y) = \begin{cases} 1, & f(x,y) \geq Th \\ 0, & otherwise \end{cases} \quad (3.3.3)$$

The idea of Otsu's method is to find out the threshold that minimizes the variance within the class and it also maximizes the variance between the classes.

$$\sigma_w{}^2(t) = q_1(t)\sigma_1{}^2(t) + q_2(t)\sigma_2{}^2(t) \quad (3.3.4)$$

where $q_1(t), q_2(t)$ stands for the probability of the classes. By using this method, the threshold of the binary map is optimized. The figure below shows the result of the thresholding.

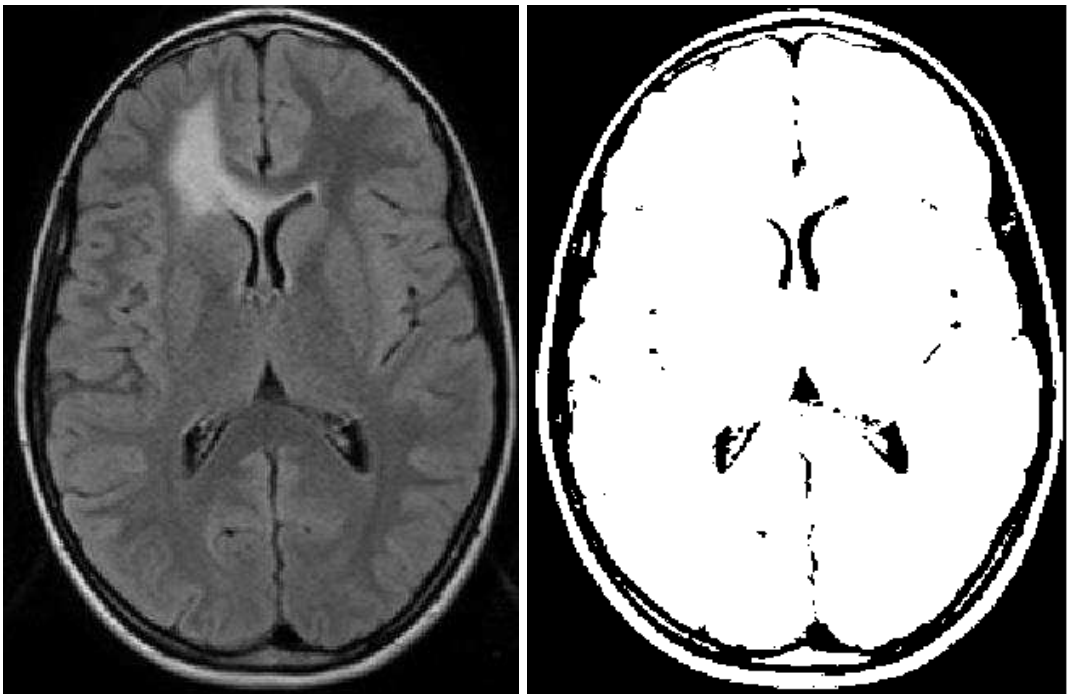


Figure 18 Otsu Method on MRI Image

After these operations, the skull within the image is removed. As a result, the preprocessed brain MRI image will be extended for future use. The figure followed shows the final result of the skull stripping method for the images.

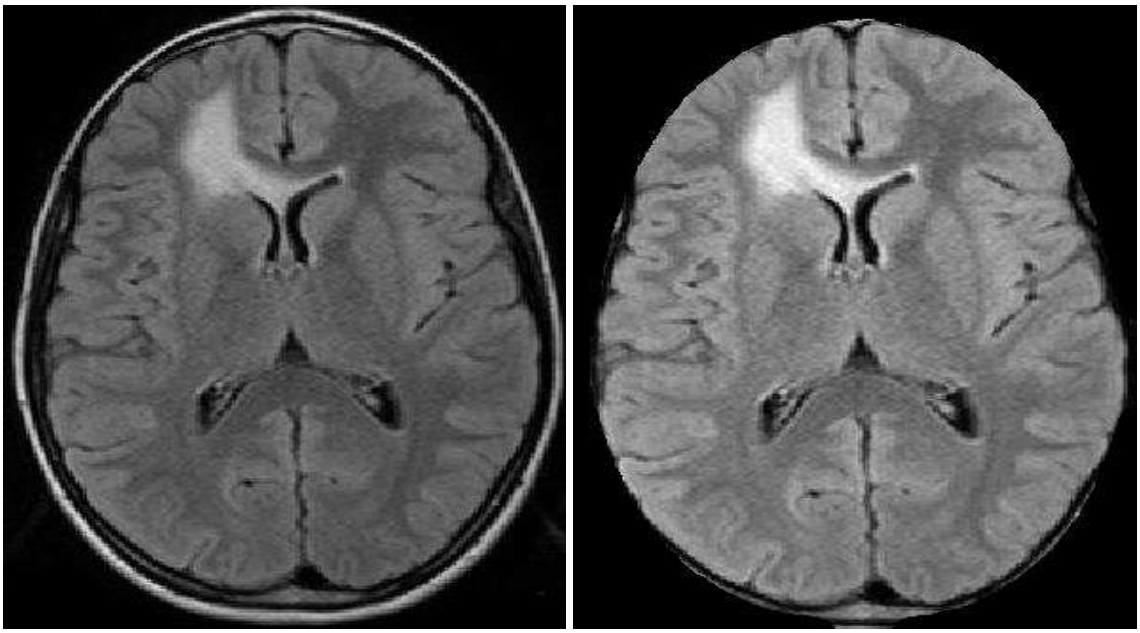


Figure 19 Skull Stripping Comparison on MRI Images

3.4 Image Segmentation

3.4.1 Edge Detection

After image preprocessing, now it comes to the step of image segmentation. Image segmentation is an important phase in medical image analysis. It separates the image into different regions, which have the common characteristics, such as color, texture, contrast, shape, edge and gray level. For brain tumor segmentation, it includes the activities of separating the tumor tissues from normal tissues within the images. There are two major steps for image segmentation, one of which is the edge detection[69].

Edge is a significant property of an image, as it illustrates the local changes of the image. Edge generally takes place at the boundary between two different regions so that edge detection offers the help of recovering the information from the image and detecting the boundaries between the regions [35,69].

An edge of the image represents the intensity changes and it is usually accompanied with a discontinuity in image intensity or the first derivative of the image intensity[56,57,69].

Based on this understanding, the gradient is applied to measure the changes. There are a variety of edge detectors for the last decades, in this thesis, Sobel detector is introduced and used after several trials to see the performance of the four classic detectors: Sobel, Prewitt, Canny and Laplacian[69].

The idea of Sobel dector is to apply a 3×3 kernel for gradient calculation. As mentioned, in order to detect the edge, gradient is regarded as a useful method for finding out the changes of the intensity within the image. For Sobel detector, the magnitude of the gradient is computed by

$$M = \sqrt{{s_x}^2 + {s_y}^2} \tag{3.4.1}$$

and the partial derivatives are calculated by

$$s_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6) \tag{3.4.2}$$

$$s_y = (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4) \tag{3.4.3}$$

where the matrix for each coefficient is defined as

a_0	a_1	a_2
a_7	$[i,j]$	a_3
a_6	a_5	a_4

Figure 20 Coefficient Matrix for Sobel Operation

c is a constant equal to 2.

In addition, the convolution masks for s_x and s_y is defined as

$$s_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad s_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Figure 21 Convolution Mask for Sobel Operation

According to the calculation method of Sobel detector, it is shown that this operator puts an emphasis on the pixels closer to the center of the mask. The figure followed shows the result of Sobel detector for the brain image.

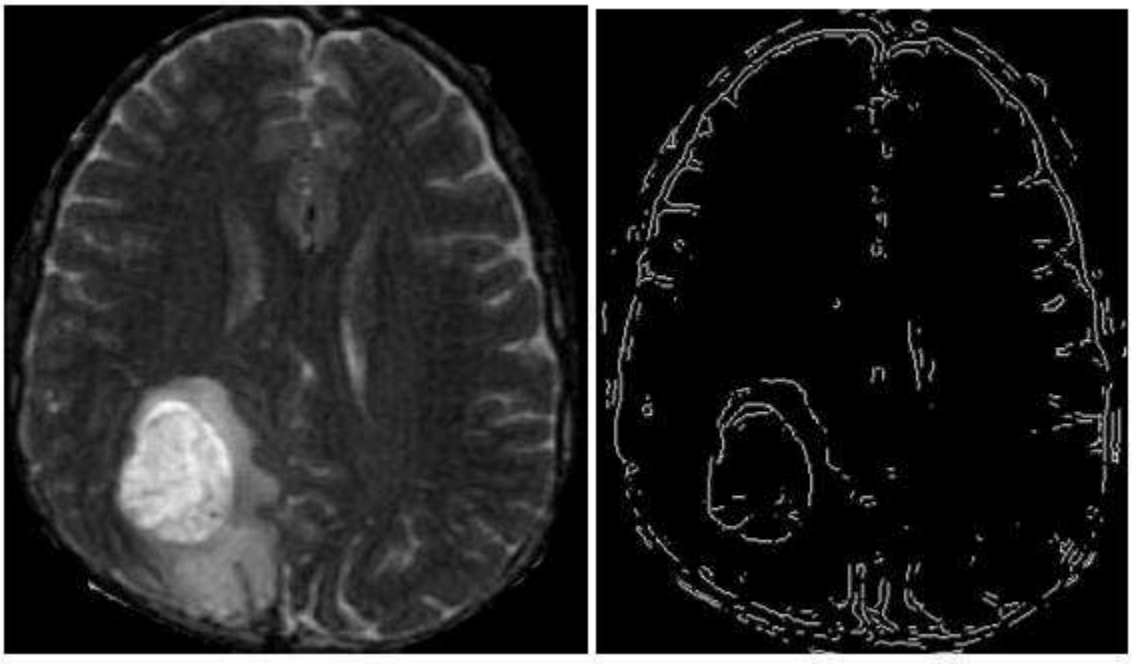


Figure 22 Sobel Operator on MRI Images

Based on the result of the edge detection, it primarily offers the regions of different areas so that in the segmentation phase, the area of the tumor region will be paid attention to.

3.4.2 Active Contour Based Segmentation

There are a variety of segmentation techniques for medical images, in this thesis, an edge based segmentation method is used, which is active contour model.The idea of the active contour model is inspired by the snake movement on designing the energy functional .The designing of the snake is regarded as an energy minimization procedure, where the total energy is minimized by three terms: internal force, image force and external force[64]. The energy function for this model is defined as:

$$E^*_{snake} = \int_0^1 E_{snake}\big(v(s)\big)ds = \int_0^1 E_{int}\big(v(s)\big) + E_{image}\big(v(s)\big) + E_{ext}\big(v(s)\big)ds \quad (3.4.4)$$

where E_{int} refers to the internal engery of the bending spilne, E_{image} raises the image force and E_{ext} represents the external constrain forces.

Internal forces can be found within the curve itself, serving to force a piecewise smoothness constraints[65]. The image forces push the snake toward salient image features like lines, edges and subjective contours. The external forces are computed from image data and responsible for moving to the snake near the local minimum [66].

The apparent advantage of active contour model is its robustness to anomalies and noise within the images because of the contribution of applying geometric constraints on photometric ones, the integration of the energy, as well as the entire length of the curve[36,64]. The flexibility of active contour handles a variety of features by placing adequate constraints[64]. However, this formulation may lead to an unstable behavior, which is the drawback of the algorithm.

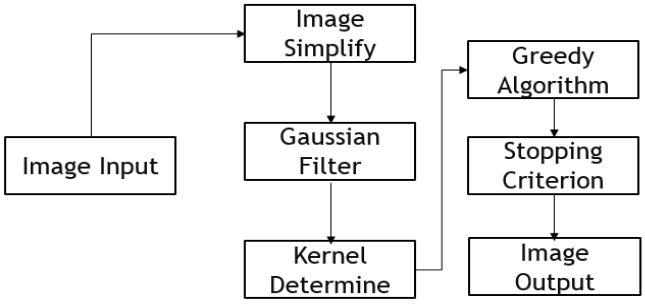


Figure 23 Process of Active Contour Based Image Segmentation

The figure above shows the process of the active contour model. Like other segmentation method, before the actual operation, it is recommended to smooth the image by using Gaussian filtering. After determining the kernel, the greedy method is established. By gradient search, the edge of the image is transformed and the boundaries of the image are delineated. In this thesis, only a small neighborhood of surrounding pixels are computed of energy. This is because that if the computed energy of a certain point is lower than the current energy, the contour will point to this new location[66,69].

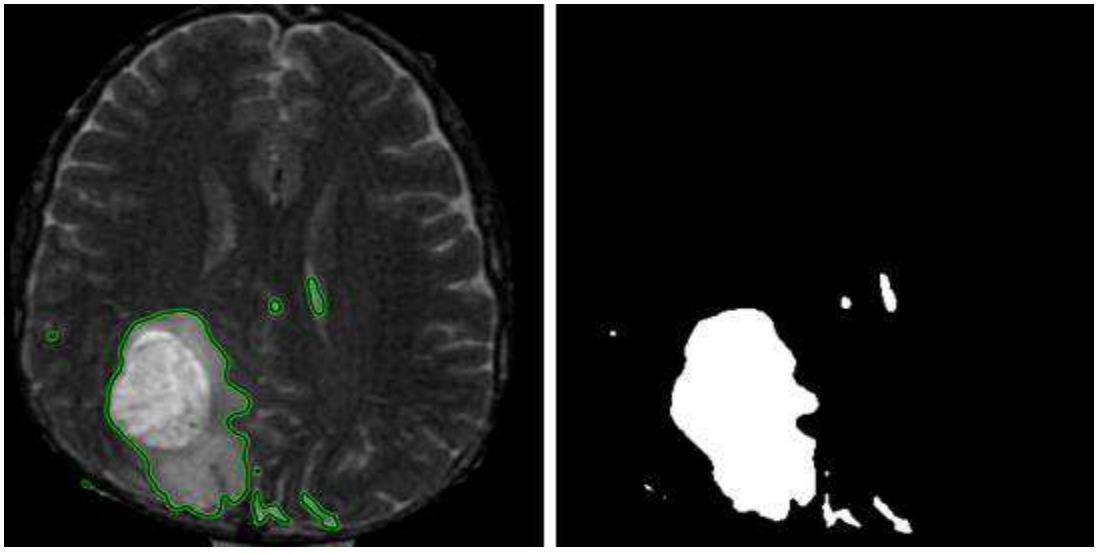


Figure 24 Active Contour Segmentation on MRI Images

The figure above shows the result of the segmentation. Concerning the active contour model, there is no indication about stopping criterion on the snake evolution, within this thesis, the number of iterations stops at 3000 after the test trial.

3.5 Medical Image Feature Extraction

A key part of the medical image processing is feature extraction, as it reveals the mathematical properties of different regions. The major purpose of the feature is to reduce the data involved in the calculation, as it already contains the relevant information of the image, which is extended for future use like classification, segmentation[39]. In order to get the better results, the first major step is the selection of the features extracted from the image.

Feature extraction includes the activity of extracting specific characteristics from the pre-processed images belonging to different categories. As mentioned, good feature selection can improve the efficiency of the model and get better results, as a result, it is recommended to pick up those that can distinguish themselves with others. There are some criteria of good feature selection. The first comes to the uniqueness of the feature, as each object should have its own unique representation, like ID for human, to distinguish itself from other. Another thing is the integrity, for the algorithm can use it to describe the object as a whole part and it will not change with the changes of external environment like position, size or rotation[52,53]. Besides, it would be better that the feature selected is easy to interpret and integrated into the algorithm, based on the consideration on the implementation efficiency.

For this understanding, there are a variety of classic features extracted from medical image for extended use. As the figure depicted, there are three major types of features

within the images, the mathematical definition and its functionality will be introduced in details in the succeeding part.

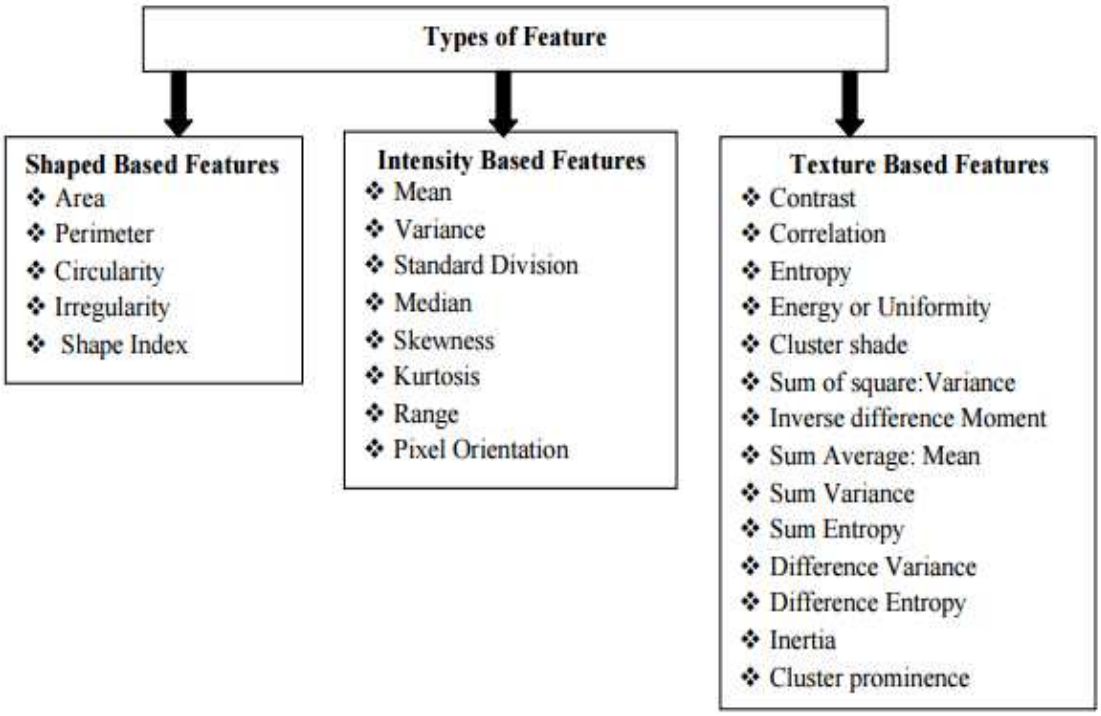


Figure 25 Summary of Different Types of Medical Image

(Adapted from. Feature extraction & image processing for computer vision by Nixon M S, Aguado A S by 2012, Academic Press)

With the help of these features, the convolution neural network will operate to distinguish different parts of the brain image, like white matter, gray matter, tumor and so on, as different tissue have different values of these features based on their nature property.

3.5.1 Intensity Based Feature

Intensity based features are frequently used for image analysis. Mean, Variance, Standard Deviation, Median, Skewness and Kurtosis belong to this category, which are extracted from segmented regions or region of interest. An example of using these feature is Linear Discriminant Analysis for classification. The mathematic formula of these feature are shown as followed:

Mean(μ):the average level of intensity of the image

$$\mu = \frac{1}{M \times N} \sum_{0}^{M-1} \sum_{0}^{N-1} f(x, y) \tag{3.5.1}$$

where $f(x, y)$ refers to the 2-dimensional function of the image with size of $M \times N$

Variance(σ^2):the variation of the intensity around mean

$$\sigma^2 = \frac{1}{M \times N} \sum_{0}^{M-1} \sum_{0}^{N-1} (f(x, y) - \mu)^2 \tag{3.5.2}$$

Skewness(μ^3): the symmetry property of the image

$$\mu^3 = \frac{1}{M \times N} \times \frac{1}{\sigma^3} \sum_{0}^{N-1} (f(x, y) - \mu)^3 \tag{3.5.3}$$

Kurtosis(μ^4):the flatness of the histogram

$$\mu^4 = \frac{1}{M \times N} \times \frac{1}{\sigma^4} \sum_{0}^{N-1} (f(x, y) - \mu)^4 \tag{3.5.4}$$

Range (R): The range has two elements: one is the minimum pixel intensity value and the other is the maximum pixel value.

Pixel Orientation:

$$PO = tan^{-1}\left(\frac{y - m}{x - m}\right) \tag{3.5.5}$$

where m refers to the point which is required to measure, $x\ and\ y$ represents the point in the axis.

3.5.2 Texture Based Feature

Texture provides higher order description of images and includes the spatial distribution of tone variations. Compared with the intensity feature, the texture feature illustrates the similarity within the image. For MRI image, the texture features distinguish the areas of

gray matter, white matter, cerebrospinal fluid and tumor region, which is the foundation of object detection. The classic texture features are shown as followed:

Energy(En): the quantifiable amount of the extent of pixel pair repetition, this parameter is used for measuring the similarity of the image.

$$En = \sqrt{\sum_{0}^{M-1}\sum_{0}^{N-1} f^2(x,y)} \tag{3.5.6}$$

Entropy(E):measures the randomness of the textural in the image

$$E = -\sum_{0}^{M-1}\sum_{0}^{N-1} f(x,y)\log_2 f(x,y) \tag{3.5.7}$$

Contrast(Con):measures the intensity variation of a pixel and its neighbor

$$Con = \sum_{0}^{M-1}\sum_{0}^{N-1} f(x,y)(x-y)^2 \tag{3.5.8}$$

Correlation (Cor):measures the relations between the target pixel and its neighbor

$$Cor = \frac{\sum_{0}^{M-1}\sum_{0}^{N-1} f(x,y)(x,y) - M_x M_y}{\sigma_x \sigma_y} \tag{3.5.9}$$

Inverse Difference Moment (IDM): the measurement of local homogeneity of an image and it could be used to detect whether the image is textured or not.

$$IDM = \sum_{0}^{M-1}\sum_{0}^{N-1} \frac{1}{1+(x-y)^2} f(x,y) \tag{3.5.10}$$

Coarseness($Cness$):the measurement of the roughness of the texture within the image. Generally, good textures of the image have small values of coarseness.

$$Cness = \frac{1}{2^{M+N}} \sum_{0}^{M-1}\sum_{0}^{N-1} f(x,y) \tag{3.5.11}$$

Directional Moment (DM): the measurement the texture alignment of the image from angle view.

$$DM = \sum_{0}^{M-1}\sum_{0}^{N-1}|x-y|f(x,y) \tag{3.5.12}$$

Sum Average ($Mean$):

$$Mean = \frac{1}{2N}\sum_{0}^{2(N-1)} f(x,y) \tag{3.5.13}$$

Sum Variance (SV):

$$SV = \frac{1}{2N}\sum_{0}^{2(N-1)} (f(x,y)-Mean)^2 \tag{3.5.14}$$

Sum Entropy (SE):

$$SE = -\sum_{0}^{N-1} f(x,y)\log_2 f(x,y) \tag{3.5.15}$$

Cluster shade (CS): the measurement of the skewness of the image matrix

$$CS = \sum_{0}^{M-1}\sum_{0}^{N-1}\big(f(x)+f(y)-\mu_x-\mu_y\big)^3 p(x,y) \tag{3.5.16}$$

where $p(x,y)$ represents the probability density of the image.

Cluster prominence (CP): the measurement of the asymmetry of the image

$$CP = \sum_{0}^{M-1}\sum_{0}^{N-1}\big(f(x)+f(y)-\mu_x-\mu_y\big)^4 p(x,y) \tag{3.5.17}$$

Generally, the features of shape and texture are regarded as the classic features for identification of the medical image. In this thesis, an approach called Grey Level Co-occurrence Matrix (GLCM) is implemented for feature extraction[54].The original design

of GLCM matrix is to extract the second order statistical texture features from the image, while nowadays it has been extended into higher order textures features, based on the relationships among pixels[54,55]. Within this thesis, only second order statistics are calculated and used. This is mainly because that even though the theory of using GLCM for third or higher order is possible, the time consuming for calculation plus the difficulty of interpretation is a major setback of GLCM in the implementation case.

GLCM reveals the texture profile of the image, as mentioned, it usually applies on the second order texture measurement, which defines the relationship between the neighboring pixels, while the first order measurement is declared within the original image like mean , variance and so on[54,55].

The GLCM matrix is derived from how often a pixel of gray-level i occurs horizontally, vertically, or diagonally to its adjacent pixels j . And the direction analysis of the GLCM matrix is presented as followed. As the figure illustrated, there are four angles within the matrix: zero degree, 45 degree, 90 degree and 135 degree[54]. For GLCM matrix, in general case, it starts at the distance of 1 and zero degrees.

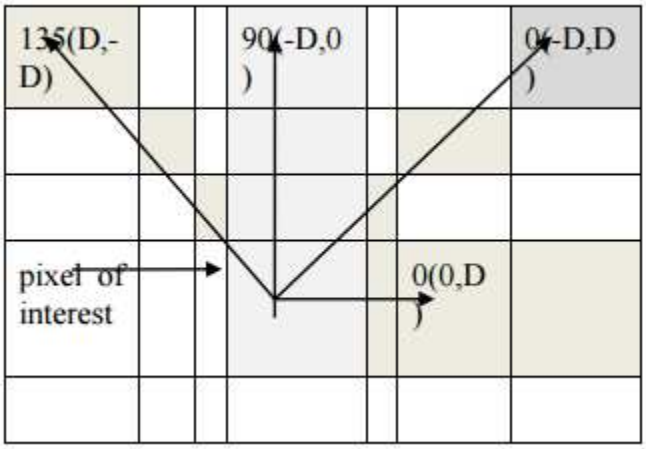


Figure 26 Different Angles of the GLCM Matrix

(Adapted from Machine vision by Jain R, Kasturi R, Schunck B G. by 1995 McGraw-Hill)

The GLCM matrix reveals how often different combinations of pixel brightness values occur in an image. The figure and the tables below provide an example of what GLCM matrix looks like for a certain area of the images. With the help of the GLCM matrix as well as the formula above, the texture based features are calculated for future use.

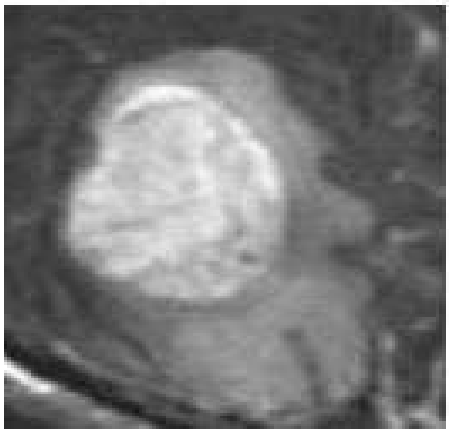


Figure 27 An Example of Brain Tumor Image for GLCM

Table 2 Pixel Value of Certain Region of the MRI Image

30	33	35	35	35	35	34	33
30	33	35	37	37	36	36	35
30	33	36	38	39	37	37	37
30	33	37	40	41	38	38	39
30	34	38	41	43	41	41	42
30	34	39	43	44	46	46	46

Table 3 GLCM Matrix for Corresponding Image

598	330	29	9	0	0	0	0
393	7789	558	26	6	4	0	0
14	540	2376	391	6	7	1	0
0	6	401	2724	184	5	7	0
0	1	13	184	1030	264	18	1
0	0	2	14	272	1959	124	3
0	0	0	5	13	134	251	1
0	0	0	0	1	1	3	0

The GLCM matrix gives the calculation of the texture based features, as mentioned, there are several shape based features, which are also suitable for the brain tumor image analysis[56]. In this thesis, three features under this category are used for image analysis: area, perimeter and circularity, which can be regarded as the supplement for GLCM and for brain tumor detection.

3.5.3 Shape Based Feature

The image shape is a binary representation of the object and shape features are regarded as the geometric properties of the object, which can reveal the valuable information for

image identification [69]. Concerning the analysis of shape features, the external boundary is usually used, as it depicts the outline and region area of the shape. Shape based features are calculated by using connected regions within the image and the boundary pixels are used to calculate the perimeter and area. There are several major indicators under this category, which in the real case are extracted from the segmented region of the image, as they contain the foremost information.

Area (A) is the number of pixels inside the target region, which includes the boundary.

Perimeter (P) is a path surrounding the target region and it can be interpreted as the number of pixels along the boundary line.

Circularity (C) is the measurement of how closely the shape of an object approaches to a circle. This parameter is applied in two dimension and it is dependent on the shape large-scale features rather than the sharpness of its edges and corners.

Shape Index (SI) is a statistic method to quantify the shape of any unit of area.

The formula of these indicators are shown as followed and $\mathrm{E}_d(x, y)$ is defined as the boundary pixel of the region,

$$\mathrm{E}_d(x,y) = \begin{cases} 1, & if\ connectivity\ of\ b(x,y) = 2 \\ 0, & otherwise \end{cases} \tag{3.5.18}$$

where

$$b(x,y) = \begin{cases} 1, & f(x,y) \geq Th \\ 0, & otherwise \end{cases} \tag{3.5.19}$$

$$Th = 0.5(\max(\ f(x,y)) - \min(\ f(x,y))) \tag{3.5.20}$$

$$P = \sum_{0}^{M}\sum_{0}^{N} \mathrm{E}_d(x,y) \tag{3.5.21}$$

$$C = \frac{4\pi A}{P^2} \tag{3.5.23}$$

$$SI = 1.27AL \tag{3.5.24}$$

where L is the length of the longest axis of the region.

The table below gives the feature evaluation on different types of the brain tissue. According to the results, there are differences among the parameters within the example images. With the help of the images from the training set, the rules for classification of the different types of the tissue will be learnt. There is one thing to mention here that the terms in the table have no units.

Table 4 Feature Evaluation on Different Types of Brain Tissue

Feature	Malignant Tumor	Benign Tumor	Normal Brain
Contrast	0.07	0.096	0.208
Correlation	0.96	0.97	0.942
Cluster Prominence	124.147	236.453	108.169
Cluster Shade	15.47	27.246	0.603
Dissimilarity	0.069	0.093	0.16
Energy	0.422	0.364	0.221
Entropy	1.275	1.561	1.908
Homogeneity	0.965	0.954	0.927
Sum of squares	3.303	4.921	9.143
Sum average	3.145	3.685	5.454
Sum variance	7.127	11.063	20.795
Sum entropy	1.226	1.489	1.734
Coarseness	0.766	0.569	0.22
DM	1.675	1.898	1.55
IDM	1.566	0.894	0.454
Area	136.5	492	-
Perimeter	132	488	-
Circularity	0.0984	0.026	-

3.6 Summary

In this section, the basic process of the medical image processing as well as the methods for brain tumor image processing are shown. In this thesis, its purpose is to detect the tumor within the MRI images. In order to achieve this goal, there is a necessity of pre-process the images to help to improve the quality of the images and obtain the important information. For tumor detection, the usage of the CNN is to find out whether there is a tumor within the images and classify it if there exists. The image pre-processing serves as a step of getting rid of the noise and image segmentation helps to localize the tumor. In the next chapter, the details of implementing the CNN will be presented.

4 CNN Architecture Design and Training

In this chapter, the content of designing and training the convolution neural network is introduced, which includes the CNN architecture setup, the parameter tuning as well as the training steps and consideration for CNN. This chapter provides the designing and training details of the CNN applied on medical images and serves as the theory foundation for the next chapter, which shows the results of the CNN performance on brain tumor detection.

4.1 CNN Architecture Design

Reviewing the architecture of CNN, as mentioned, there are several layers within the CNN architecture, so designing the CNN starts with the design of each layer.

4.1.1 Input Layer

The first layer of the CNN architecture comes to the input layer, where the medical images are imported as the inputs. The images from the BRATS Challenges have been extracted into 32×32 pixels, which can be regarded as the region of interest, containing the valuable information of the tissue. By using this method, it avoids the meaningless calculation and analysis on the region which doesn't stand for the property of the target tissue.

4.1.2 Convolution Layer

The convolution layer carries on the foremost calculation and the most sophisticated tasks within the architecture. As mentioned in the previous chapter, to start the convolution layer, first it is recommended to decide the volume size, which requires the settings of the four hyperparameters: the number of the filters K, the spatial extent F, the stride S and

the amount of zero padding P. Putting aside the setting of the number of the filters K, according to the reference cases, the settings of the rest three parameters are applied as $F = 2, S = 1, P = 1$.Besides, within the convolution neural network, the neurons are organized into three dimensions: height, width and depth. In this case, the input image volume is $32 \times 32 \times 1$, as a result, the output volume should be $16 \times 16 \times$ K, according to the formula followed,

$$W2 = \frac{(W1 - F) + 2P}{S + 1} \tag{4.1.1}$$

$$H2 = \frac{(H1 - F) + 2P}{S + 1} \tag{4.1.2}$$

$$D2 = K \tag{4.1.3}$$

where $W1$, $H1$, $D1$ is the width, height and depth of the input and $W2$, $H2$, $D2$ is the width, height and depth of the output.

The next step is to decide the size of the convolution kernel and the number of the filters used for convolution layer. There is no rules and restriction on the settings of these parameters. In common cases, they are dependent on the complexity of the problem and network architecture. The common size settings for the convolution kernel are 3×3, 5×5 and 7×7 , depending on the size of the image inputs. In this case, the image inputs of the brain tumor have been extracted into 32×32 and the stride is set as 1. As a result, in order to improve the learning speed of the convolution neural network and guarantee the accuracy at the same time, 5×5 is chosen as the kernel size of the convolution calculation.

For the decision on how many filters are used for image convolution, according to the previous reference, the number of the filter is usually set as the power of 2. So in this

thesis, the number of the filters is set as 64. In the next chapter, there will be a section illustrating the influence of different parameter settings on the performance of the convolution neural network, which will include the number of the filters as well as the convolution kernel size.

4.1.3 Non-linear Layer

The next layer is the non-linear layer, which operates the non-linear transformation of the output from the convolution layer. In the previous chapter, the ReLU function is commonly applied for non-linear transformation.

Rectified linear unit has been proved to speed up the training procedure of the convolution neural network. For rectifier function, it does not saturate for positive input. Moreover, ReLU is easy to implement and promote sparse activations of the inputs[72].

$$f(x) = \max(x, 0) \tag{4.1.4}$$

$$f'(x) = \begin{cases} 1, & x > 0 \\ 0, & otherwise \end{cases} \tag{4.1.5}$$

In this thesis, there is an improvement of the transformation function in order to accelerate the speed of the learning. This improvement is based on the traditional rectified linear units and it is introduced by Clevert et al. in 2016[72]. It is called the exponential-rectifying function, and the definition is shown as

$$f(x) = \begin{cases} x, & x > 0 \\ a(e^x - 1), & otherwise \end{cases} \tag{4.1.6}$$

and the derivative of the function is presented as

$$f'(x) = \begin{cases} 1, & x > 0 \\ ae^x, & otherwise \end{cases} \tag{4.1.7}$$

The idea of this improvement is to address the vanishing gradient problem. Compared with the ReLU function, the ELU has a hyperparameter, α, which controls the value to

which an ELU saturates for negative inputs. Therefore, the vanishing gradient problem is alleviated, as the positive part of these functions is an identity, whose derivative is one and not contractive[72].

The vanishing gradient problem is an issue taking place during the training phase of convolution neural network, where updating the parameters of convolution layers by backpropagation becomes difficult due to vanishing of gradient[72]. Take Sigmoid function as an example, which has the gradient in the range (0, 0.25]. During the backpropagation phase, it computes the gradient by the chain rule, which means the gradient decreases greatly after several multiplication within the hidden layers. In the end, the front layers experience slow training on such small number.

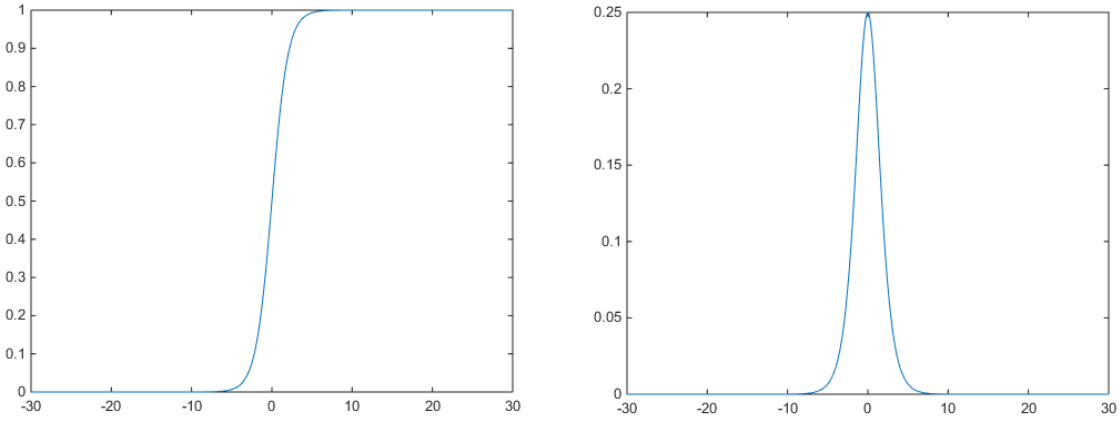


Figure 28 Sigmoid Function and Its Derivative

ELU has minimized this issue compared with ReLU. By saturating smoothly to -1 for negative values, it forces the mean unit activation closer to zero. With the mean activation closer to zero, it enhances the learning efficiency, for it pushes the gradient closer to the natural gradient, reducing the bias shift effect, which is similar to batch normalisation[72].

Based on this foundation, there is an improvement of placing a parameter β on the function, which is defined as:

$$f(x) = \begin{cases} x, & x > 0 \\ a(e^{\beta x} - 1), & otherwise \end{cases} \quad (4.1.8)$$

$$f'(x) = \begin{cases} 1, & x > 0 \\ a\beta e^{\beta x}, & otherwise \end{cases} \quad (4.1.9)$$

where β controls the scale of the exponential decay.

Using this parameter, the network can control its non-linear behavior throughout the course of the training phase. If β becomes smaller, the decay will be larger and the saturation is faster. The visualization of this change will be presented in the next chapter.

4.1.4 Pooling Layer

After the non-linear layer is the pooling layer. As mentioned, there are two classic types of pooling: max pooling and average pooling. In this thesis, max pooling is used for its efficiency. The purpose of max pooling is to reduce the complexity of the previous layer. By applying the max pooling, there are settings for the window size and the stride. In this case, the size of the window is set as 2 and the stride is also 2, which are common settings of max pooling. These settings improve the convergence rate of the network and the figure followed shows the movement of the max pooling process.

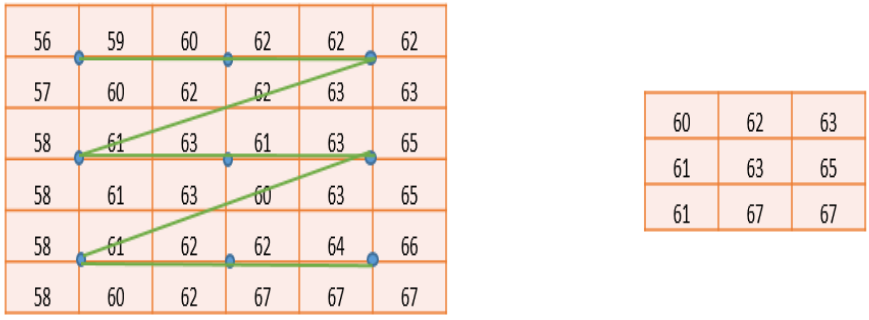


Figure 29 Stride and Kernel Design for Pooling Layer

4.1.5 Fully Connected Layer (Softmax)

After several repeats of the convolution layer, non-linear layer and pooling layer, now the data comes to the fully connected layer, whose task is to analyze the high-level features learnt by the model and classify the inputs based on these features into different categories. Within the fully connected layer, every input is connected with every output unit with the corresponding probability so that the spatial information of the image is missing[73].

The fully connected layer is usually one-dimensional and each neuron stands for the group needing to be classified. The output produced by this layer is a score for each class. In this thesis, its aim is to detect the tumor within the MRI image and classify them. For this understanding, within the final layer, there are ten classes, specifying the different tissue types within the image.

However, the output from the convolution neural network could be hard to put, for this concern, it is recommended to finish the convolution neural network with a softmax function. The idea of the softmax function is to train the network and penalize the deviation between the true labels of the instance. It can be interpreted as the error function within the fully connected layer [46]. The main job of the softmax function is to transform the labels into probabilities and then apply a cross-entropy function to compute the loss which determines the penalty. The softmax function is defined as:

$$S(Y) = \frac{e^{z_j}}{\sum_{j=1}^{k} e^{z_j}} \tag{4.1.8}$$

where z_j is defined as the total weighted sum of inputs of the output unit j

The loss is then computed by using cross-entropy function, measuring the difference between two probability vectors. In this case, there are ten classes needing to be

specified in the fully connected layer, so the the error function for this multi-class cross-entropy is

$$MCE(\theta) = \sum_{i=1}^{n}\sum_{j=1}^{10} \delta y_i j \ln \sigma_j \tag{4.1.9}$$

where $\delta y_i j$ is the Kronecker delta:

$$\delta y_i j = \begin{cases} 0, & if\ y_i \neq j \\ 1, & if\ y_i = j \end{cases} \tag{4.1.10}$$

The table below summarizes the setting within each layer of the convolution neural network. The details of training the network will be shown in the next subchapter.

Table 5 Convolution Neural Network Architecture

Layer	Layer Content	Tensor size
0	Input Image	1×32×32
1	Conv(5×5,64 maps)	64×28×28
2	Batch Normalization	-
3	Non-linearity	64×28×28
4	Maxpooling(2×2,stride 2)	64×14×14
5	Conv(5×5,64 maps)	64×10×10
6	Batch Normalization	-
7	Non-linearity	64×10×10
8	Maxpooling(2×2,stride 2)	64×5×5
9	Fully Connection (10)	10×1×1
10	Non-linearity	10×1×1
11	Fully Connection + Soft max	20×1×1

4.2 CNN Training

After settling down the configuration of each layer, the next procedure is to train the network with the image training set. The purpose of training the convolution neural network is to adjust its internal parameters to increase its performance on the input data. For this understanding, training the network can be interpreted as an optimization problem, given a suitable objective function and constraints.

Concerning the network training, it can be divided into several steps: weight initialization, parameter updating, stochastic gradient descent, backpropagation and regularization. The details of each phase will be illustrated as followed. The figure below summarizes the basic computation principal of training convolution neural network.

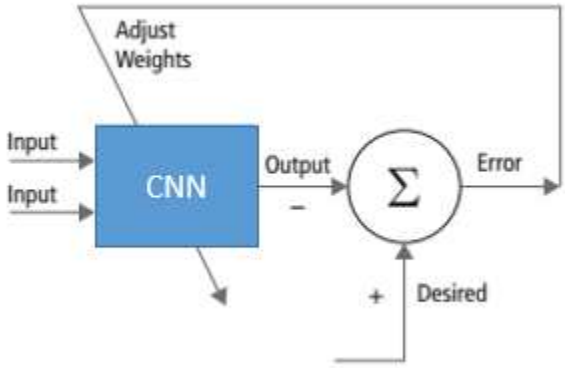


Figure 30 Training Process of Convolution Neural Network

4.2.1 Weight Initialization

Initializing the weights in convolution neural network is an active research area. As the first step of network training, the initialization of the parameters needs some serious consideration. If the initial weights of the network are set too small or too large, it will result in the problems with backpropagation, as the network needs more time and cost to adjust the parameters through the backpropagation, therefore, it is of great importance

that the initial values are appropriately configured[73]. This thesis won't cover the topic of comparing the method of how to choose the initial weights with another.

When training the convolution neural network, it is recommended that all the weights are not initialized to zero, as it makes the output of any input remain unchanged. For this consideration, choosing random weight initializations can lick this issue. Learning from the reference cases of image analysis, a common practice is to start with small weights. The input weights to each unit in the network are initialized by sampling values from a Gaussian distribution with mean μ= 0 and variance $\delta^2 = 2/Nin$, where Nin is the number of inward connections of node and all the biases are set to zero. The settings are based on the central limit theorem, as the volume of the inputs is huge. By this means, it helps to ensure that the network isn't initially highly biased towards any particular subset of inputs, which leads to the problem of slow speed of convergence. In addition, it is also recommended to normalize the variance for the output of each node to make sure that all the neurons have an equal impact on determining the output. It is mainly because that normalization speeds up the process of convergence[46].

Like mentioned, in order to avoid delays in convergence process because of improper weight vector initializations, the batch normalization layer should be applied before the non-linear activations. The main advantages of batch normalization are faster learning and higher overall accuracy. This approach allows the network to get a higher learning rate, potentially providing another boost in speed.

Normalization is regarded as the pre-processing step to make the data comparable among features by shifting inputs to zero-mean and unit variance. As the data flows through the convolution neural network, the weights and parameters adjust those values; however,

they sometimes make the data too big or too small again, which is referred as internal covariate shift. It will covariate shift occurring within a neural network going from layer 2 to layer 3, for example. This is because that the distribution of outputs of a certain specific layer in the network may change through the phase of the network learning and weights updating, which leads to the issue of slowing down the learning procedure.
By normalizing the data in each mini-batch, this problem is largely avoided, as the batch normalization helps to make the data flowing between intermediate layers of the network look like whitened data. Besides, since batch normalization has a regularizing effect, it can also remove the influence of dropout. The implementation of batch normalization will be introduced in details together with the SGD training.

4.2.2 Parameter Updating

After initializing the weights of the network, the next step is to update the weights so as to reduce the penalty incurred from the cost function. First, the cost function needs to be chosen. In this thesis, the cross-entropy error function is applied:

$$MCE(\theta) = \sum_{i=1}^{n}\sum_{j=1}^{k} \delta y_i j \ln \sigma_j \qquad (4.2.1)$$

When it comes to training phase, for convolution neural network, it mainly refers to the process of tuning the internal parameters to enhance the classification performance of the input data. In order to obtain the good training results, the learning procedure can be divided into two phases[73]. The first phase comes to feeding the network with the entire training set to complete one epoch. Within each time, the convolution neural network updates the coefficients of the weights on a single iteration. The second phase focuses on the prediction performance of the previous trained network and making the evaluation of

the network performance on the validation set. Both the validation dataset and training dataset need to be disjoint in order to make generalization evaluation.

The parameters have a huge effect on the performance of the convolution neural network, so there is a need of choosing the appropriate settings for the parameters. The parameters chosen in advance of the learning process are called hyperparameters, which include the number of layers, the number of the neurons within the layer, the chosen kernel size in the convolution layer as well as parameters that affects the gradient descent algorithm[73]. The previous three settings have been introduced in the previous subchapter within the design of each layer within the CNN architecture, and there are three additional hyperparameters needing confirmation before the learning procedure, which are learning rate, momentum and weight decay[73].

Learning rate η, affects the impact on each weight update, which directly influences the speed of the network convergence. A large η can lead to the problems of missing the optimal solution while a small η could result in a really long time for training the model.

Momentum γ, whose range belong to (0,1],determines the inertia of the gradient update, which affects how much weight is assigned to the last gradient. A large γ could result in the oscillation reduction of stochastic gradient descent.

Weight decay λ,whose range belong to [0,1),determines the weight regularization. Multiplied by this parameter, the weights will be less than 1 after each update. The purpose of this is to avoid the weights of growing too much during the learning period.

The equations followed update the weights of the hyperparameters involved:

$$\omega \leftarrow \omega' = \omega - v \tag{4.2.2}$$

$$v \leftarrow v' = \gamma v + \eta \frac{\partial C}{\partial \omega} + \frac{\lambda}{2n} \sum_i {\omega_i}^2 \tag{4.2.3}$$

Furthermore, there is another important thing during the learning process, which is to update the learning rate over time. This can be realized by decreasing it with exponential decay every 5 epochs. Concerning the exponential decay, the involved parameter v determines the decaying rate based on the equation given

$$\eta \leftarrow \eta' = \frac{\eta}{1 + kv} \tag{4.2.4}$$

where k represents the number of the iteration, in this case, it equal to 5.

The table below shows in the information of the hyperparameters within the convolution neural network.

Table 6 Hyperparameters Details of the Convolution Neural Network

Parameter	Description	Value
K	Number of the kernels form convolution layer	64,64
CLF	Filter Size of convolution layer	(5,5)(5,5)
CLS	Stride size of convolution layer	(2,2)(2,2)
CLS	Pool size of convolution layer	(2,2)(2,2)
L	Loss function	Cross-entropy
L'	Regularization Method	L2
σ	Activation function	ELU
h	Number of neurons	2048
η	Learning Rate	0.0015
λ	Decay strength	10^{-5}
γ	Momentum	0.9
b	batch size	16

4.2.3 Gradient Descent

Until now, the architecture of convolution neural network has been described. However, the major problem in neural network is how to choose the appropriate values for a given problem[73]. The main target of training the network is to find out the suitable values for the weights and biases, which minimize the error function $E(W, b)$[73]. In this thesis, an optimization algorithm called stochastic gradient descent with momentum is applied. Even though there are a variety of ways to train a network, this method focuses attention on training a network by using error backpropagation and gradient descent, as it can handle various output-unit activations and error functions [68].

The error function $E(W, b) = E(\theta)$is a smooth continuous function of all weights and biases and it is apparent in the parameter-space. The error function used in this thesis is introduced in the previous subchapter, where the cross-entropy error function is applied.The purpose of training is to search a solution to minimize the error function, however, in actual cases, it is not easy to find the perfect solution to satisfy the equation$\nabla E(\theta) = 0$, which is due to the fact that the error function is not a convex problem. It has many local optima besides the global optimum. For this concern, iteration numerical method is introduced to find a sufficiently good solution.

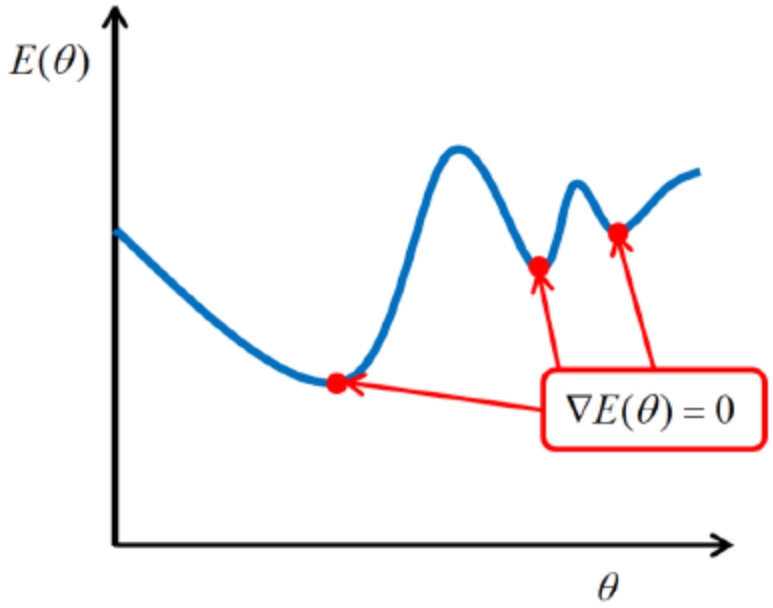


Figure 31 Error Function at Global and Local Optima

(Adapted from Neural networks for optimization and signal processing by Cochocki A, Unbehauen R by 1993 John Wiley & Sons, Inc.)

In the initial steps, a parameter vector is chosen for a starting point $\theta^{(0)}$, then the next steps are conducted in the form of

$$\theta^{(k+1)} = \theta^{(k)} + \Delta\theta^{(k)} \tag{4.2.5}$$

where k represents the number of the iterations. While there are different attempts to find the updated value of $\Delta\theta^{(k)}$,the gradient descent algorithm presented in this section uses gradient information. In particular, a single iteration step involves a small step in the direction of the negative gradient of the form

$$\theta^{(k+1)} = \theta^{(k)} - \eta\nabla E\left(\theta^{(k)}\right) \tag{4.2.6}$$

where $\nabla E(\theta)$ stands as the direction of the greatest rate of increasing the error function and η is defined as learning rate, which is always positive in general. These iteration steps of updating the parameters of the convolution neural network are continued until an expected optimum is reached.

The basic idea of the gradient descent algorithm is illustrated in figure below, where one updated step of two different initial parameter values is iterated. In either cases, the parameter θ is moved in the direction of the greatest decrease of the error function[73] . After each iteration, it is necessary to calculate $E(\theta)$ for the new parameter vector $\theta^{(k+1)}$. In the gradient descent method, this is achieved within the whole training set, which means every parameter of the whole training set needs updating in order to revaluate the $E(\theta)$.

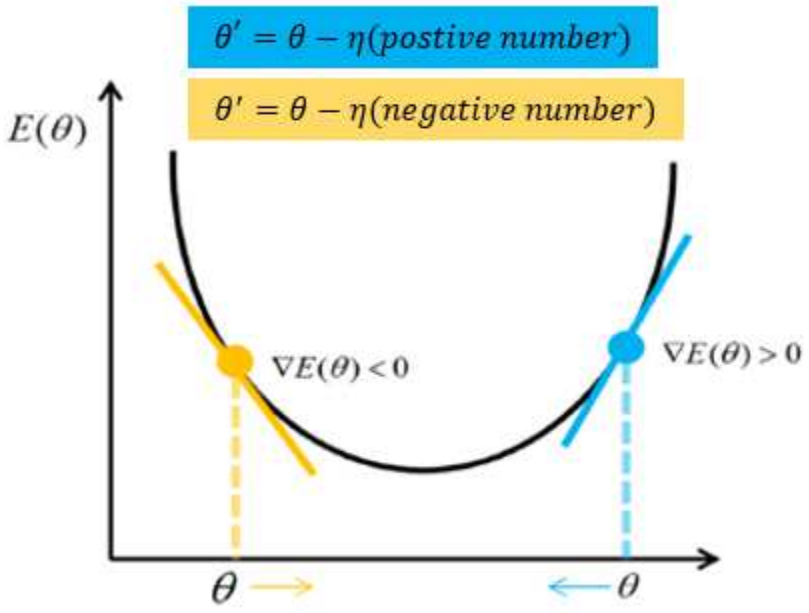


Figure 32 Gradient Descent Algorithm

(Adapted from Neural networks for optimization and signal processing by Cochocki A, Unbehauen R by 1993 John Wiley & Sons, Inc.)

Until this point, the focus has been mostly on analyzing convex functions with one minimum for the entire function. However, the cost functions may have multiple local minima along with the global minimum that provides the optimal solution. In such case, based on the initialization of weights, the network might be stuck at one of the local

minima during the learning and can't converge to the optimal solution. This is due to the fact that gradient descent is a greedy algorithm. The issue of being stuck in a local minimum is more probable when the learning rate is small. One way to avoid converging to a local minimum is by applying momentum on the gradient descent. The aim of adding the momentum term is to speed up the training, leading to a faster convergence without leading to oscillations, which happens by increasing the learning rate.

$$\theta^{(k+1)} = \theta^{(k)} - \eta \nabla E\left(\theta^{(k)}\right) + \alpha \ \Delta E\left(\theta^{(k)}\right) \qquad (4.2.7)$$

where α is a hyperparameter called momentum.

Like mentioned, in order to alleviate the vanishing gradient, during the training, batch normalization is applied. Each activation of the mini-batch is centered to zero-mean and unit variance. The mean and variance are measured over the whole mini-batch, independent from each activation.

Once the learning has stopped, a post-training step is applied where the mean and variance for each activation are computed on the whole training dataset rather than on mini-batches. This new mean and variance replace the ones computed on mini-batches.

Value of x over a mini-batch: $B = \{x_{1.....m}\}$

$$\mu_B \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$$

$${\sigma_B}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)$$

$$\widehat{x_i} \leftarrow \frac{x_i - \mu_B}{\sqrt{{\sigma_B}^2}}$$

Return $\{y_i = BN(x_i)\}$

4.2.4 Backpropagation

Based on the probability vector, the convolution neural network classifies the inputs into the categories. After the feeding forward, there is a softmax activation, which is used for the backpropagation method, which is applied for updating the weights within the architecture. The process starts from the end layer to the first layer. For the non-linear layer, there is no need to update as this layer only provides the non-linear transformation of the inputs [68].Backpropagation also doesn't happen within the pooling layer, for it only reduces the complexity of the input plus there is no derivatives calculation involved within the layer.

The backpropagation operates the convolution layer. If the error occurs in the layer before the convolutional layers and it is defined as E, and the kernel size is defined as $k \times$ k, then the chain rule is performed as

$$\frac{\partial E^l}{\partial {W_{ij}}^l} = \sum_{i=1}^{m-k} \sum_{j=1}^{m-k} \frac{\partial E^l}{\partial Y^l} \frac{\partial Y^l}{\partial {W_{ij}}^l} \tag{4.2.8}$$

For elements within the convolution layer, the equation is

$$\frac{\partial Y^l}{\partial {W_{ij}}^l} = {x_{ij}}^{l-1} \tag{4.2.9}$$

so the equation is changed into

$$\frac{\partial E^{l+1}}{\partial Y^l} = \sum_{i=1}^{m-k} \sum_{j=1}^{m-k} \frac{\partial E^{l+1}}{\partial {x_{ij}}^l} \frac{\partial {x_{ij}}^l}{\partial {y_{ij}}^l} \tag{4.2.10}$$

Combined the derivative of the activation function, the final one is

$$\frac{\partial E^l}{\partial {W_{ij}}^l} = \sum_{i=1}^{m-k} \sum_{j=1}^{m-k} \frac{\partial E^l}{\partial {x_{ij}}^l} {x_{ij}}^{l-1} f'\left(\partial {y_{ij}}^l\right) \tag{4.2.11}$$

During the period of evaluation, error backpropagation is involved, which provides an efficient way for evaluating derivatives[73]. The idea is to compute an error term $\delta_i^{(l)}$ for each unit i in layer, which is used to see how much this unit devotes into any errors in the output. For the output layer, $\delta_i^{(nL)}$ can be computed directly by measuring the difference between the output and the expected values. However, the error terms for the hidden units cannot be directly seen so that this is solved by propagating the errors δ backwards through the network. This whole procedure of backpropagation is illustrated in figure below.

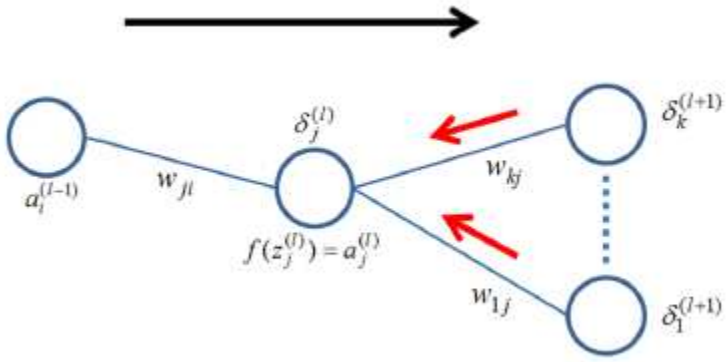


Figure 33 CNN Backpropagation

(Adapted from Artificial neural networks by Schalkoff R J. by 1997 McGraw-Hill)

The detailed steps for error background is presented as follow:

Step1: Apply an input vector x_i to the network and do forward propagation

Step2: Evaluate the error term $\delta_i^{(nL)}$ for every output unit

Step3: Calculate all the error term δ using the backpropagation formula

$$\delta_j^{(l)} = f'(z_j^{(l)}) \sum_k w_{kj}^{(l+1)} \delta_k^{(l+1)} \qquad (4.2.12)$$

where $f'(.)$ is the inverse of the activation function and $z_j^{(l)}$ is defined as the total weighted sum of input to unit j in layer l.

Step4: Get the gradient of the error function with respect to the weight vector w_{ij} by activation $a_i^{(l-1)}$ and error term $\delta_j^{(l)}$. The derivative formula:

$$\frac{\partial E(W,b)}{\partial w_{ji}^{(l)}} = \delta_j^{(l+1)} a_i^{(l)} \tag{4.2.13}$$

Step5: Return and repeat step 2

The backpropagation algorithm calculates the gradient of the error function and the gradient descent method is used to update the parameters within the network [73].

4.2.5 Regularization

For every machine learning algorithm, there is a consideration of avoiding the overfitting issue, and convolution neural network is no exception, as it has large capacity in dealing with data, which has a potential in overfitting. When this happens, the model will achieve poor generalization.

There are several methods, which are known as regularization, to avoid the overfitting, such as early stop, weight regularization, data augmentation and dropout. In this thesis, weight regularization is applied for avoiding overfitting, which is also called L2 regularization.

L2 regularization introduces a cost on each weight in the network. The further away a weight value is from 0, the greater the cost is. In addition, L2 regularization reduces the ambiguity and the number of possible solutions by imposing penalty on higher weights, forcing them to have weights whose magnitude is closer to zero. The penalizing is implemented by adding the term $\frac{\lambda}{2}\omega^2$ to the cost function, in this thesis, the cost function

is the cross-entropy, which is mentioned in the previous chapter, so the objective function is defined as

$$L = \sum_{i=1}^{n}\sum_{j=1}^{k} \delta y_i j \ln \sigma_j + \frac{\lambda}{2}\omega^2 \quad (4.2.14)$$

where λ is the regularization strength. The reason of term 1/2, is that the gradient of this term to parameter ω is $\lambda\omega$. The regularization term adds the product of the sum of all the squares of the weights in the network and a hyperparameter λ is used to control the amount of regularization.

The purpose of weight penalty is to reduce the size of the weights within the network, as large weights could lead to a high loss[73]. If the convolution neural network has no weight penalties, then weights are easy to be pushed from zero to huge figure. The setting of λ can be tricky, as the weight penalty increases the probability of large weights. A large weight can be treated as a certain feature being pretty important for predicting the outputs, as a result, constraints on such large weights will influence on the cost function greatly, making features with large weights treated as equally important. The larger the value of the λ parameter is, the more chance of large weights is. There is no predefined optimal value for λ, so that the value has to be determined empirically. In this thesis, it is defined as 10^{-5} according to the reference case.

In the practice phase, the number of inputs and outputs is determined by the dataset and the task. As the number of hidden units and the number of hidden layers is adjustable, it is recommended to control the complexity of the model, which avoids both underfitting and overfitting.

4.3 Summary

Until now, the details of training the convolution neural network have been shown. In this thesis, a CNN based method for tumor detection and classification is introduced. In order to identify the tumors within the images, training convolution neural network is a necessity. The essence of network training is to tune its parameter settings to reach the optima. So in the next chapter, the results of the experiment and the performance of the convolution neural network will be shown.

5 Experiment Result and Performance Evaluation

In previous chapters, the details of medical image processing and convolution neural network training haven been explained. The target goal of this thesis is to find out if there is tumor tissue within the brain MRI images. If it exists, the CNN needs to recognize it and classify it. So in this chapter, the results of detecting the brain tumor by using CNN is shown. Besides, for training the network, there are several settings needing to be tuned to optimize the performance of CNN. Like mentioned in the previous chapter, when designing the CNN architecture, how deep the convolution neural network is needs to be settled down and verified. For this consideration, there is a section of the CNN performance on tumor detection with different parameter settings.

5.1 Experiment Implementation

5.1.1 MRI Image Database

The images for this thesis come from the BRATS challenge, whose purpose is to enhance the image segmentation for brain tumor images. In the original dataset, there are more than 800 images, some of which are not good for experiment. Like the figure below, the image on the right side only presents the anatomical structure of human brain, not indicating whether it is a healthy brain image or not. In addition, some figures have the quality issues with bad resolution like the image on the left side. As a result, before the experiment, there is a need of selecting the images from the dataset.

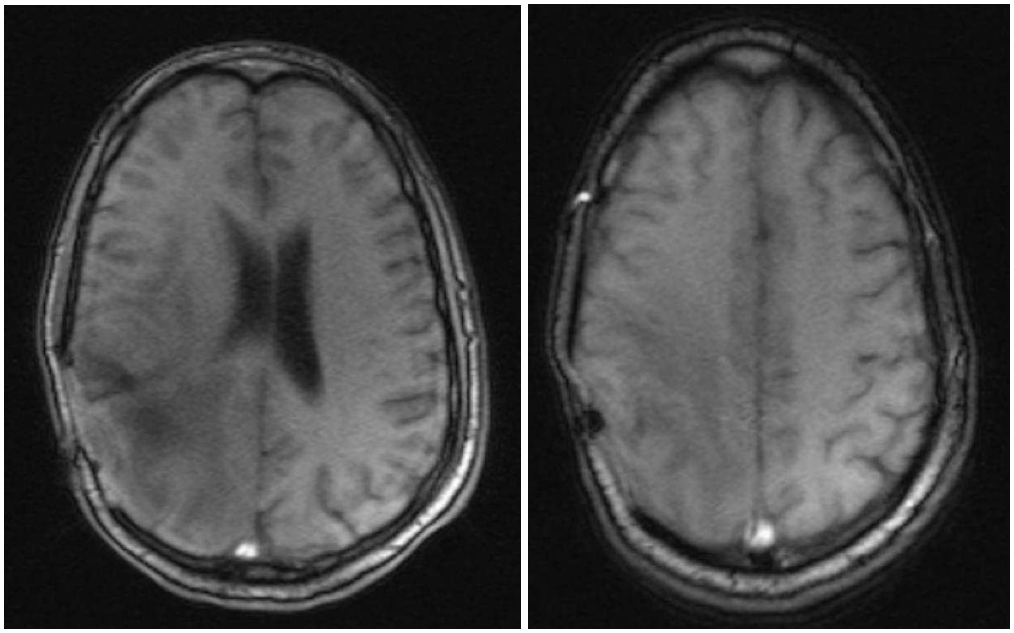


Figure 34 An example of brain MRI image not selected for experiment

After the selection, the number of the images for the experiment has been settled on 480, with 320 images for training and 160 images for testing. After the training phase, the convolution neural network tests its performance on the testing set. Not only should it find out if there is a tumor within the MRI image, but also point out it is benign or malignant if it is unhealthy tissue.

Table 7 MRI Image Dataset Information

Dataset(480)		Training Set(320)		Testing Set(160)	
Normal	Abnormal	Normal	Abnormal	Normal	Abnormal
180	300	120	200	60	100

5.1.2 Implementation Environment

The experiment of the thesis is implemented on Matlab 2014, which is integrated with MatConvNet, a MATLAB toolbox for implementing convolutional neural network. MatConvNet is an open source implementation toolbox designed with an emphasis on

simplicity and flexibility[47]. At the same time, it supports efficient computation on CPU and GPU, allowing to train complex models on large datasets. The computer operates on Intel Core i7 Processing system, with 8 processors, 3.6 GHz CPU and 16 GB RAM.

5.1.3 CNN Tumor Detection

For tumor detection, as mentioned in the previous chapter, it involves in a series of techniques, edge detection, contour detection, object segmentation and skull stripping. This thesis comes up with a new method of combining image segmentation with convolution neural network to detect whether there is a tumor within the brain MRI images and classify it if it exists.

After the image pre-processing, which involves the operation of removing the noise, enhancing the image contrast as well as stripping the non-brain tissue from the image, the object detection starts with distributing 4 kernels on 4 different angles of the picture to segment the image. By this means, it covers 95% of the image area. Through the active contour movement, there are four different segmented regions, whose features may be different from others. The figure followed shows the result of this operation.

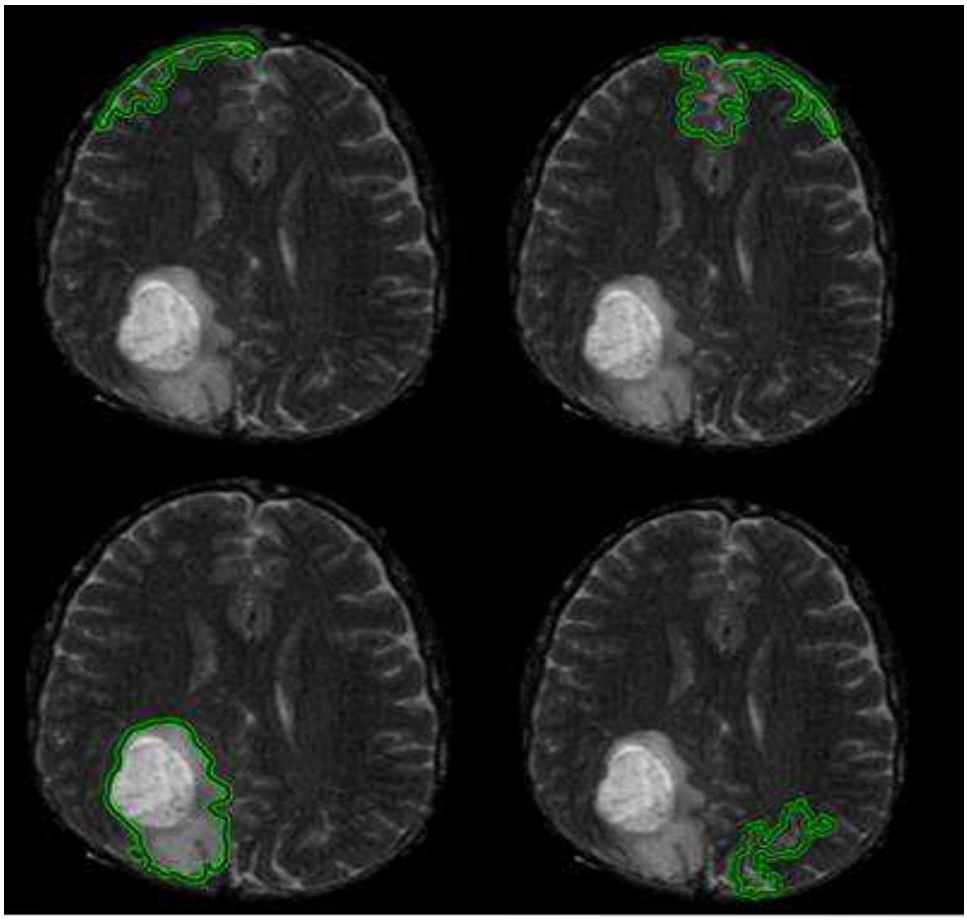


Figure 35 Brain Tumor Image Segmentation

As the features of the tumor are different from other tissue, there are differences between the tumor and other types of the brain tissue by feature calculation. Within the training period, the features from different types of the tumor are calculated and form a certain group, separated from other types of the tissue. Then during the testing period, after the image segmentation, the features of the segmented area are calculated and checked to see if it falls into any tumor categories by the usage of support vector machine. If it does, this area will be presented and outlined, which is referred as the tumor localization. If there is no tumor region, it is regarded as the healthy brain image. The figure followed shows the classification group of a texture based feature to visualize the results. For there are dozens of the features extracted from the MRI images according to the feature

introduction in the previous chapter, the SVM considers the influences of all the indicators to minimize the error in the training phase.

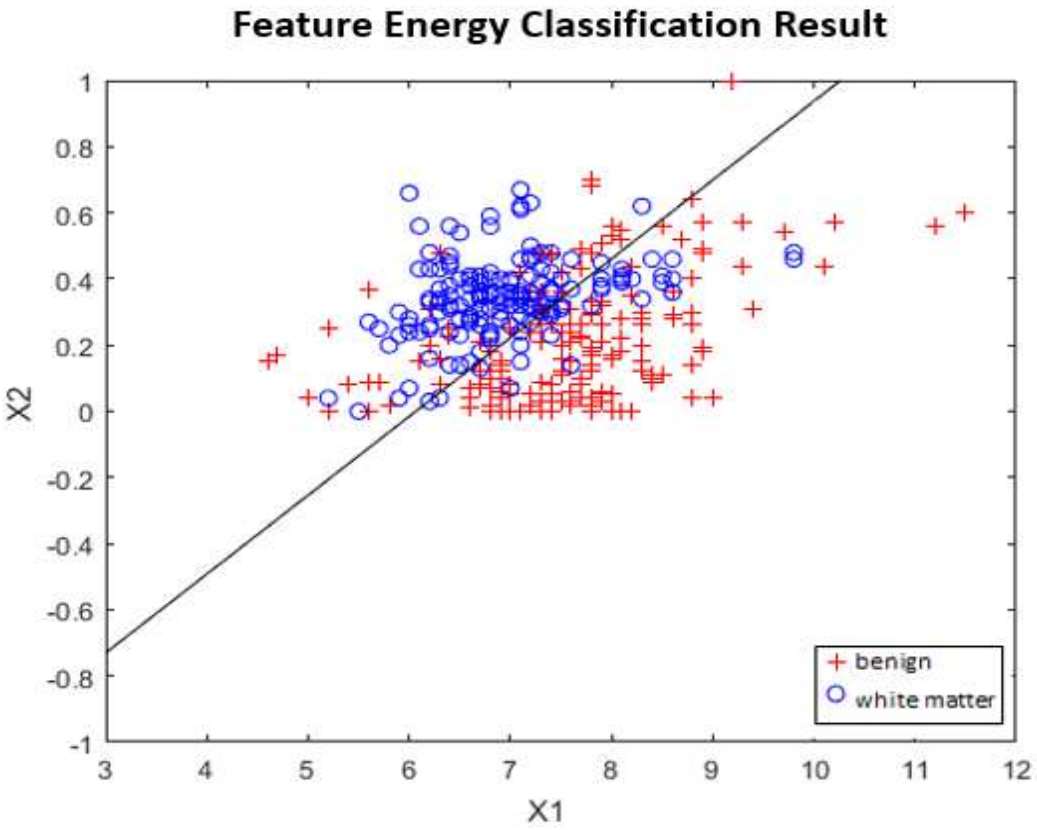


Figure 36 Feature Energy Classification

After that, the localized area is extracted as input to the CNN architecture and CNN training starts. Through dozens of the trainings, the CNN classifies the inputs into different categories based on the training results from the learning procedure. The figures below give the detection results of the convolution neural network on the example data for different types of the brain MRI images.

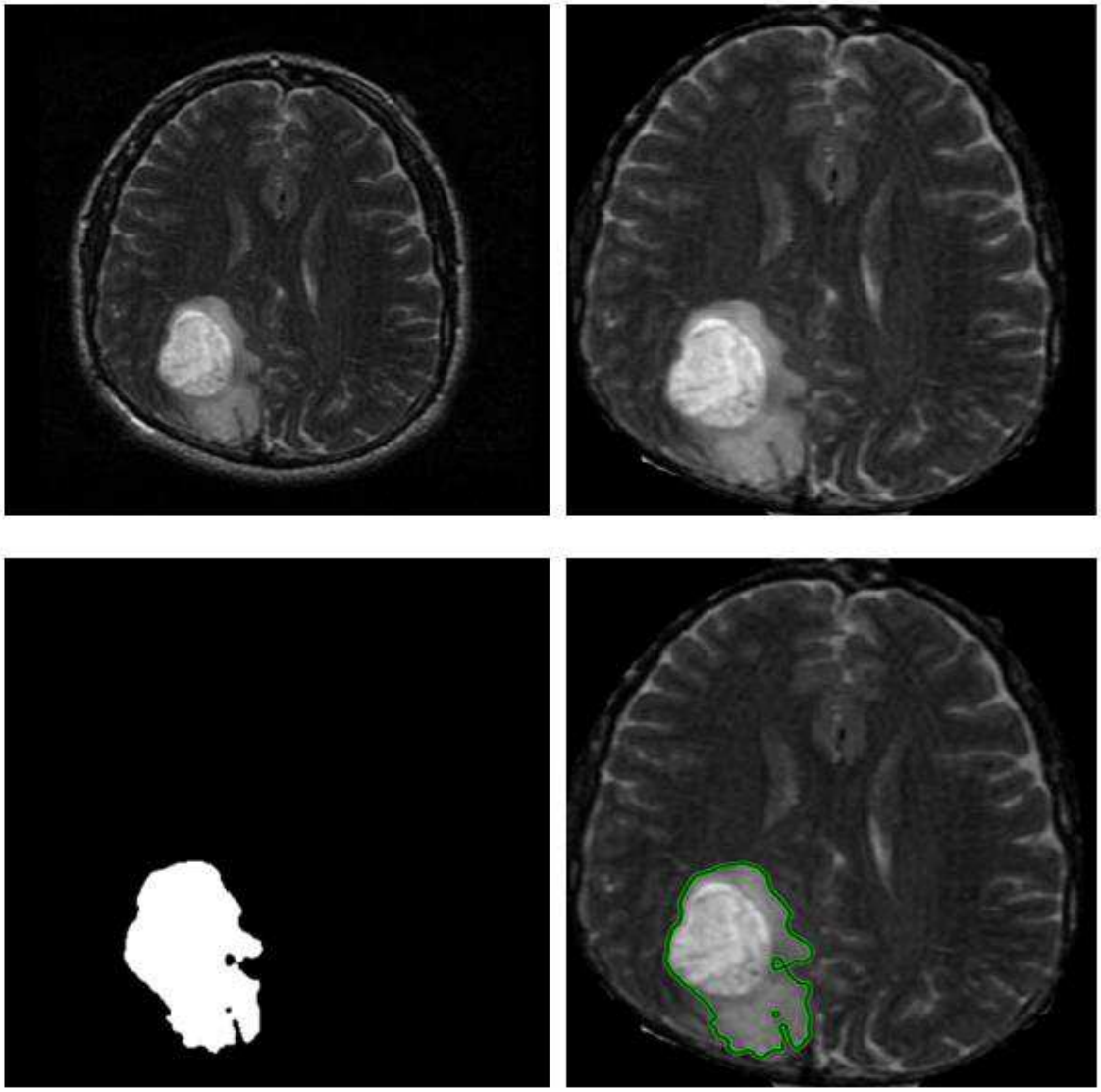


Figure 37 Detection Result on Benign Tumor

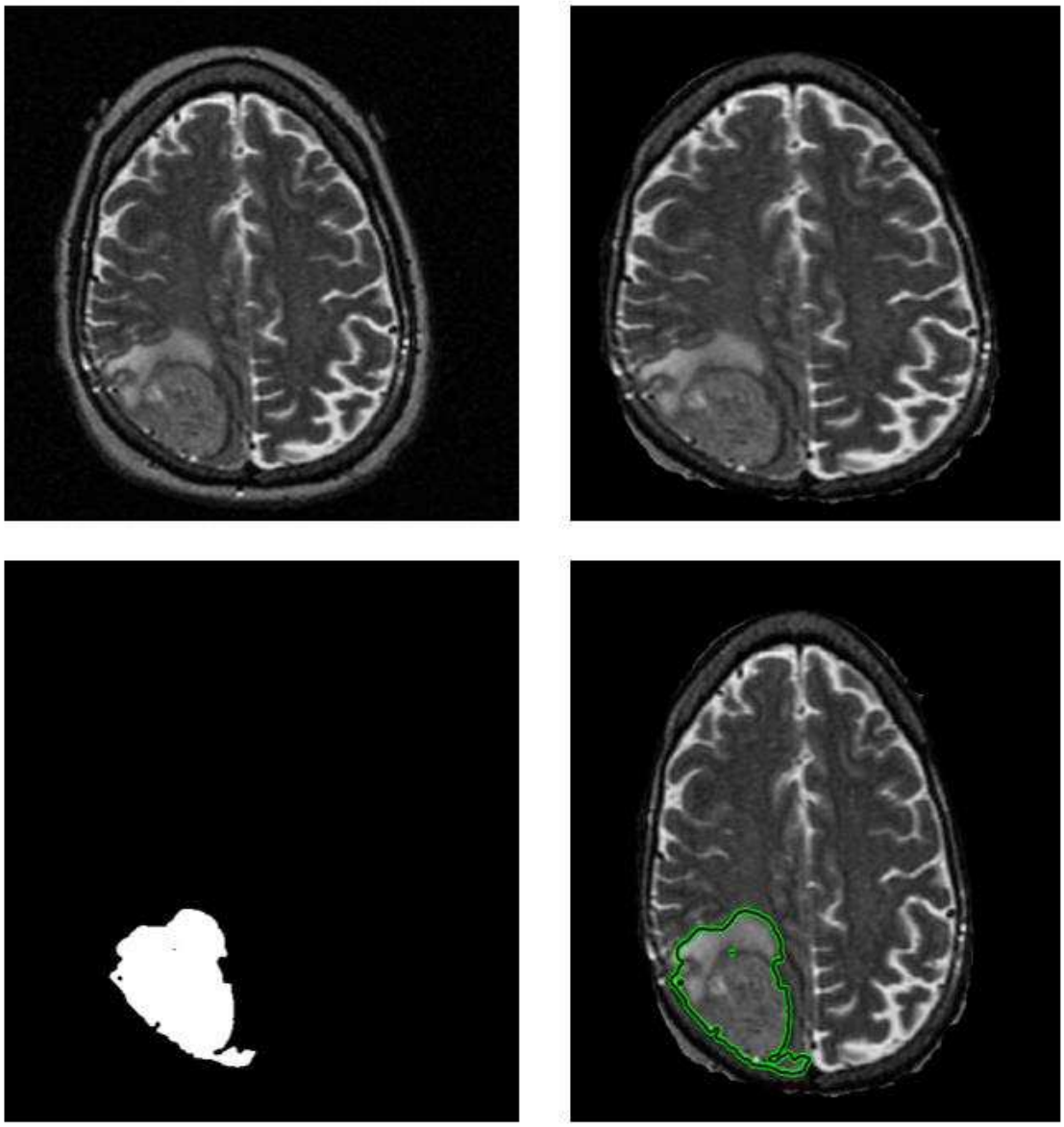


Figure 38 Detection Result on Malignant Tumor

In order to guarantee the integrity of the convolution neural network performance, there is a need of evaluating the results of CNN on detecting the brain tumor within the images. There are three classic indicators for evaluation: sensitivity, specificity and accuracy. Sensitivity is the proportion of true positives that are correctly identified, which shows how good the model is at detecting the abnormality.

$$Sentivity = TP/(TP + FN)$$

Specificity refers to the proportion of the true negatives correctly identified.

$$Specificity = TN/(TN + FP)$$

And accuracy is the proportion of true classification results either on true positive or true negative.

$$Accuaracy = (TN + TP)/(TP + FN + TN + FP)$$

In order to guarantee the quality of the testing results, cross-validation is used to reduce the variance in the validation error instead of using huge volume of data for validation. In this thesis, the dataset is separated into 4 equal partitions and it can be seen in figure below. The final result is then the mean of these results. Before the experiment, in order to avoid any dependence, the images were randomly selected.

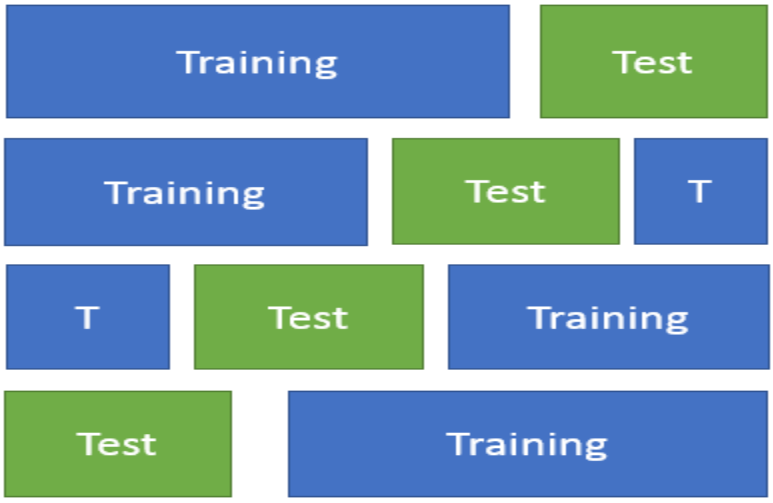


Figure 39 4-fold Cross Validation

The table followed summarizes the results of the classification, which has been divided into two parts: first shows the classification results of the healthy tissue and the pathologic tissue, and the second table presents the classification results of the benign

tumor and the malignant tumor based on the pathology ones. These classification results are extracted after 50 training epochs.

Table 8 Performance Summary of Convolution Neural Network_Part 1

Heathy Tissue & Pathologic Tissue Classification		
Actual Result	**Predicted Result**	
	Normal	**Abnormal**
Normal	45	15
Abnormal	13	87
Sensitivity	87.0%	
Specificity	75.0%	
Accuracy	82.5%	

Table 9 Performance Summary of Convolution Neural Network_Part 2

Benign & Malignant Tissue Classification		
Actual Result	**Predicted Result**	
	Benign	Malignant
Benign	32	5
Malignant	9	54
Sensitivity	85.7%	
Specificity	86.5%	
Accuracy	86.0%	

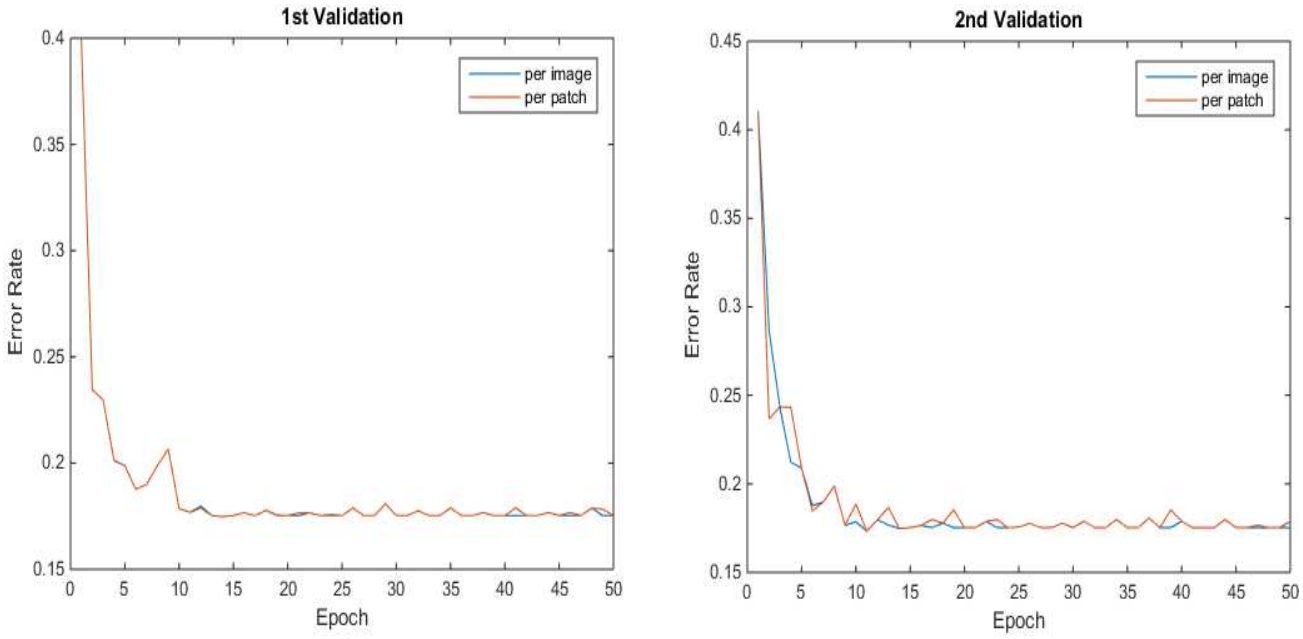


Figure 40 1st and 2nd Cross Validation Result

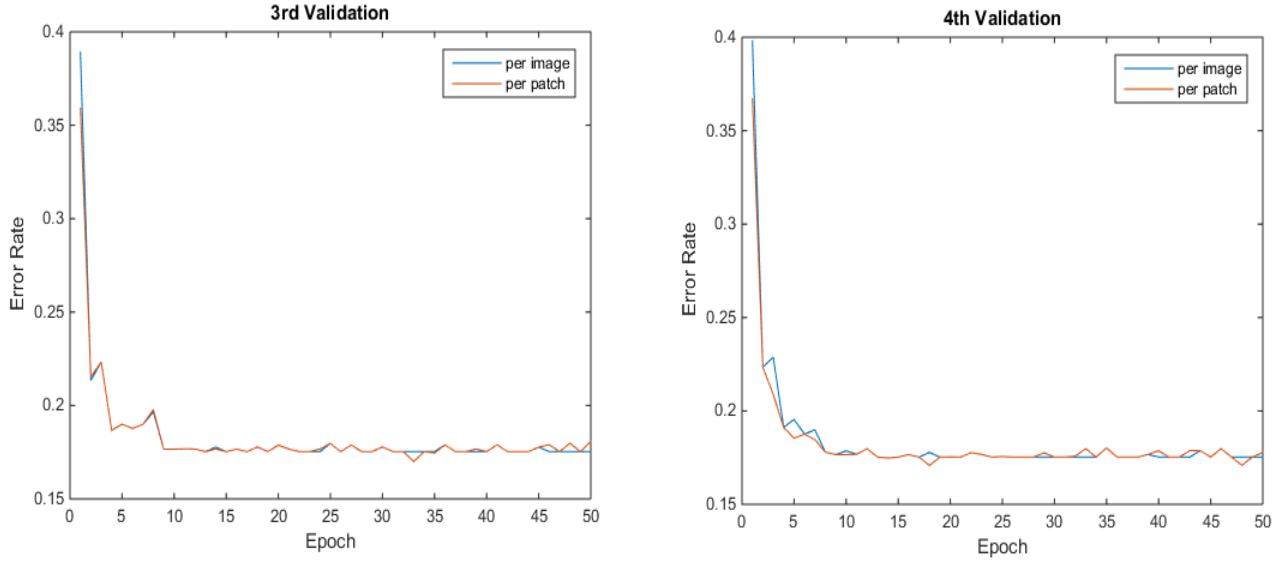


Figure 41 3rd and 4th Cross Validation Result

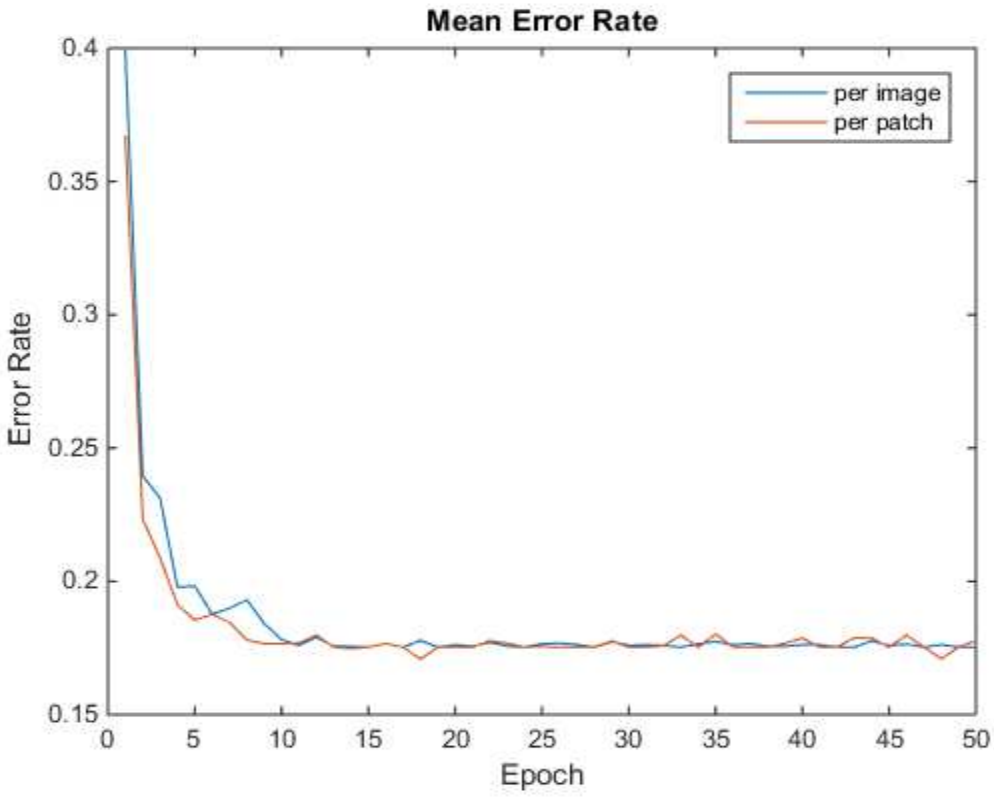


Figure 42 Cross Validation Average Result

According to the classification results, it is clear to see that the convolution neural network actually achieved great performance in detecting the brain tumor from the MRI images and classifying the pathology tissue into two categories: benign and malignant.

As this thesis focuses on testing the performance of convolution neural network in detecting brain tumor images, there should be a part of performance comparison between convolution neural network with other traditional machine learning algorithms. In this thesis, the performance of support vector machine and artificial neural network have been shown and the results are shown in the table as followed.

Table 10 Performance Comparison on Different Algorithm_Part 1

Heathy & Pathologic Tissue Classification		Heathy & Pathologic Tissue Classification		Heathy & Pathologic Tissue Classification	
SVM		**ANN**		**CNN**	
Sensitivity	63.0%	Sensitivity	59.0%	Sensitivity	87.0%
Specificity	58.3%	Specificity	63.3%	Specificity	75.0%
Accuracy	61.3%	Accuracy	60.6%	Accuracy	82.5%

Table 11 Performance Comparison on Different Algorithm_Part 2

Benign & Malignant Tissue Classification		Benign & Malignant Tissue Classification		Benign & Malignant Tissue Classification	
SVM		**ANN**		**CNN**	
Sensitivity	82.6%	Sensitivity	77.5%	Sensitivity	84.7%
Specificity	70.6%	Specificity	71.1%	Specificity	82.1%
Accuracy	79.4%	Accuracy	59.3%	Accuracy	83.9%

According to the results, it is obvious to see that the convolution neural network takes a lead in detecting the different types of the tissue from the brain MRI images so that the potential of applying convolution neural network in medical image analysis is huge.

5.1.4 Failure Case Analysis

Until now, CNN plays an important role in recognizing the brain tumor with MRI images. However, an error rate nearly of 17% is not good enough for clinical use. For this consideration, there is a need of investigating the failure cases.

A common way of analyzing the classification errors is to look at the confusion matrix. The rows represent the actual classes and the columns represent the predicted classes. The table below displays the confusion matrix for CNN in tumor detection after the last training epoch ends.

Table 12 Confusion Matrix for CNN in Tumor Detection

Heathy Tissue & Pathologic Tissue Classification		
Actual Result	**Predicted Result**	
	Normal	**Abnormal**
Normal	45	15
Abnormal	13	87
Benign & Malignant Tissue Classification		
Actual Result	**Predicted Result**	
	Benign	**Malignant**
Benign	32	5
Malignant	9	54

According to the results of confusion matrix, for both cases of using CNN in tumor detection, there are some failure cases. The figure below illustrates a certain failure case. On the left side, it is a healthy brain MRI image, and on the right side it is a benign tumor image, which was misclassified as healthy. It is easy to see the tumor area in the right figure, however, the contrast of the tumor is not apparent from other tissue of the image. By calculating the important texture features of both images, the differences between the two types of the tissue are quite minor. As a result, the CNN fails to classify the image into the right category.

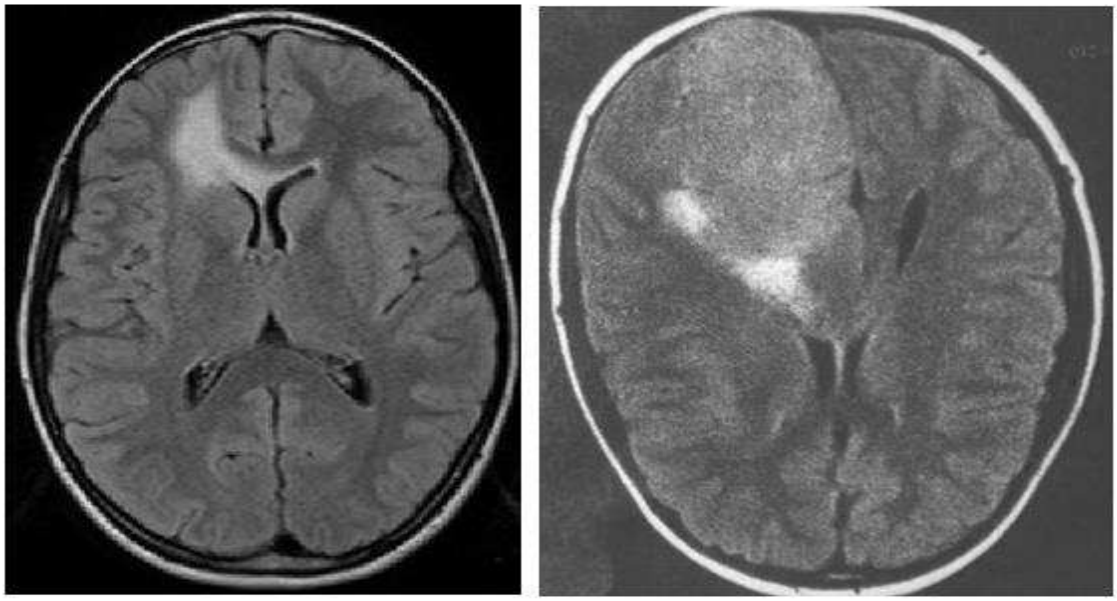


Figure 43 Benign Tumor Detection Failure Case

Table 13 Feature Evaluation on Brain Tissue_Case 1

Feature	Benign Tumor	Normal Brain
Contrast	0.194	0.208
Correlation	0.945	0.942
Cluster Prominence	119.45	108.169
Cluster Shade	0.246	0.603
Dissimilarity	0.093	0.16
Energy	0.254	0.221
Entropy	1.961	1.908
Homogeneity	0.984	0.927
Sum of squares	8.977	9.143
Sum average	4.850	5.454
Sum variance	19.663	20.795
Sum entropy	1.802	1.734
Coarseness	0.216	0.22
DM	1.204	1.55
IDM	0.458	0.454

The figure below shows another case of CNN misclassification. In the figure on the right side is the healthy brain image, same as the previous case. The image on the left side is the MRI image containing the malignant tumor. The image in the middle is the same tumor on another MRI image with different view by comparison. By rotating the view, the picture looks pretty different with the same tumor. In addition, through the calculation of the features extracted from the original images, it has minor differences from the healthy one. So in this case, the CNN fails to categorize the tumor into right class.

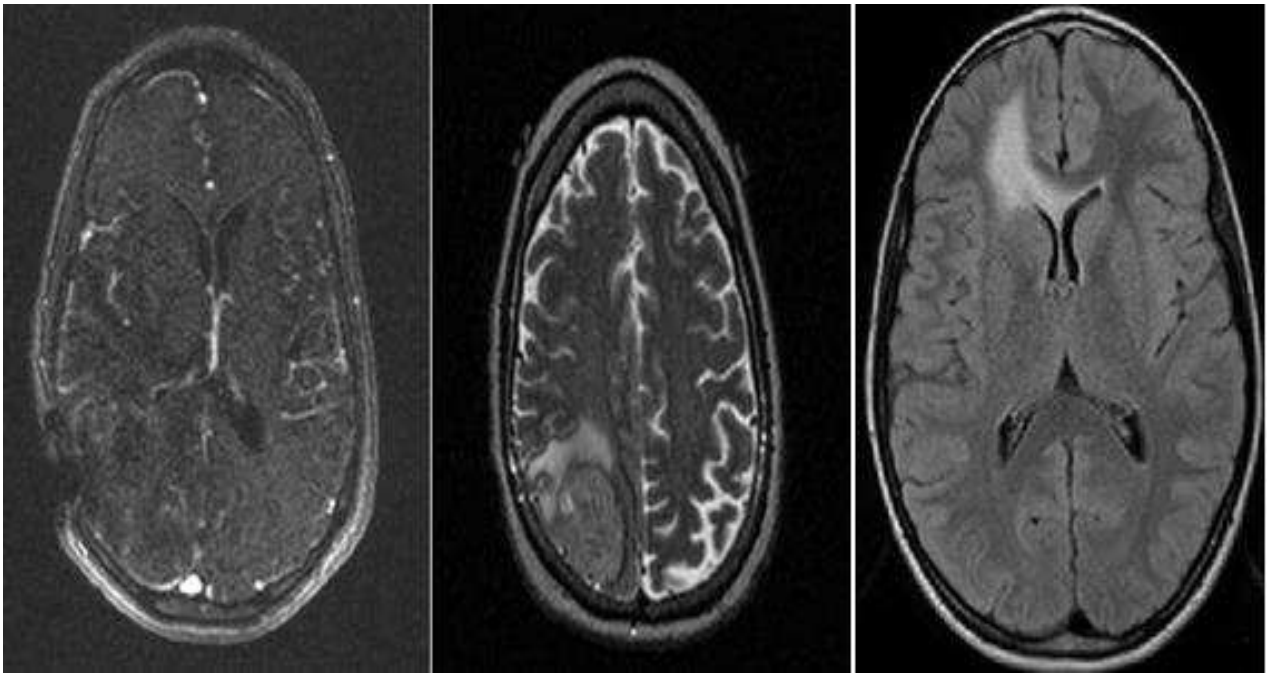


Figure 44 Malignant Tumor Detection Failure Case

Table 14 Feature Evaluation on Brain Tissue Case 2

Feature	Malignant Tumor	Brain Tissue
Contrast	0.234	0.208
Correlation	0.955	0.942
Cluster Prominence	109.433	108.169
Cluster Shade	0.666	0.603
Dissimilarity	0.123	0.16
Energy	0.334	0.221
Entropy	1.879	1.908
Homogeneity	0.956	0.927
Sum of squares	9.023	9.143
Sum average	5.343	5.454
Sum variance	21.663	20.795
Sum entropy	1.442	1.734
Coarseness	0.196	0.22
DM	1.364	1.55
IDM	0.509	0.454

According to the analysis of the failure cases of CNN misclassifying the brain MRI images, it is shown that if the quality of the MRI images is not guaranteed, the CNN may fail to recognize the brain tumor or classify them into wrong categories. As the CNN works on the training results learnt from the images, so in order to enhance the successful rate of CNN in tumor detection, there is a need to enhance the quality of the inputs.

5.1.5 Noise Robustness

Based on the results shown, the convolution neural network plays an effective role in recognizing the tumor type from the brain MRI images. For medical image processing, there is always noise with the MRI images, which affects the quality of the images. Based on this concern, it is necessary to find out whether CNN acts well in recognizing the tumor from the images containing noise.

The noise added on the MRI images has a Gaussian distribution with mean μ= 0 and variance $\delta^2 = 0.01$. The results of the experiment are shown as followed, which include the tumor detection and performance evaluation.

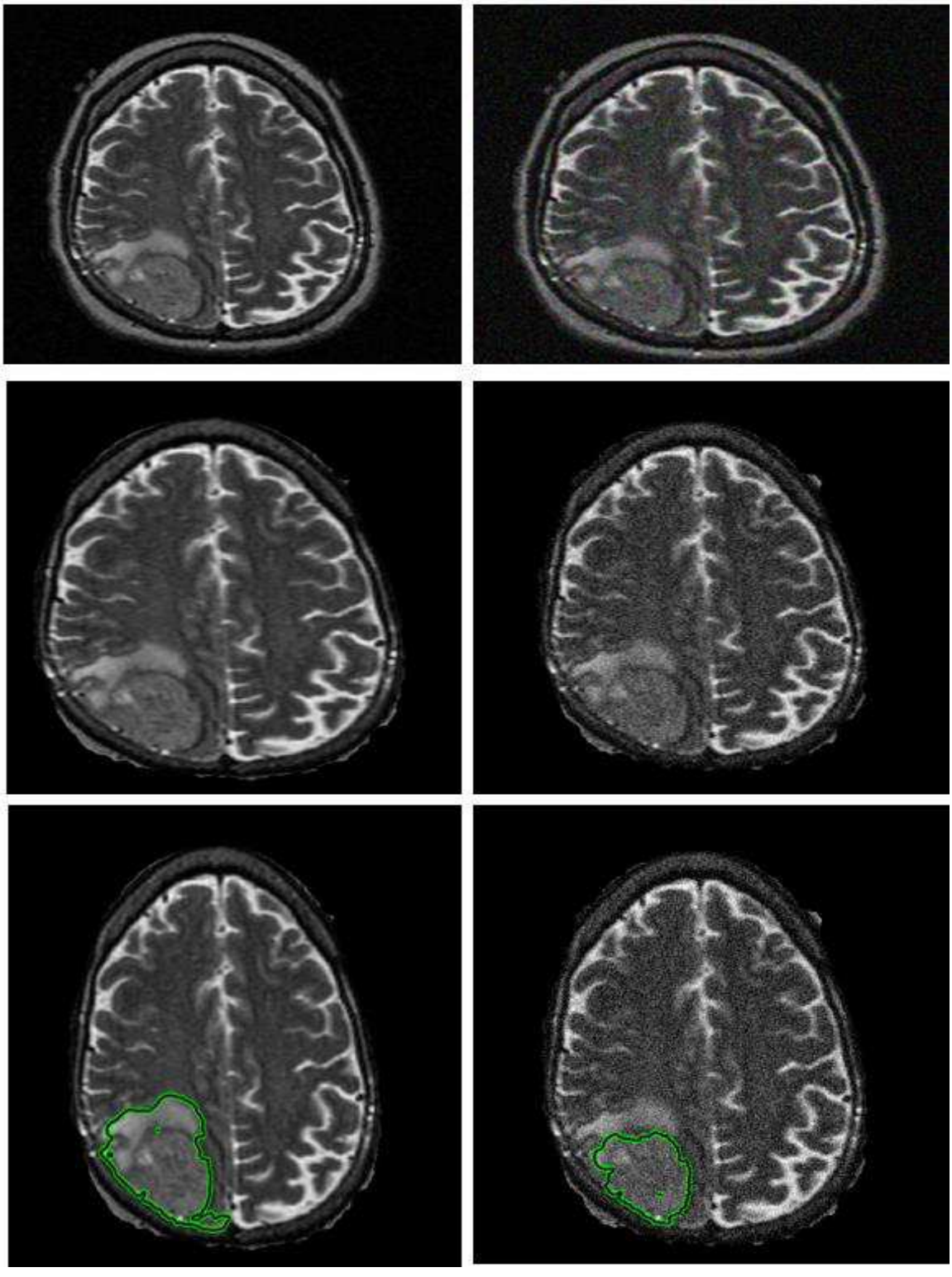


Figure 45 Tumor Detection Comparison in MRI Image with Noise

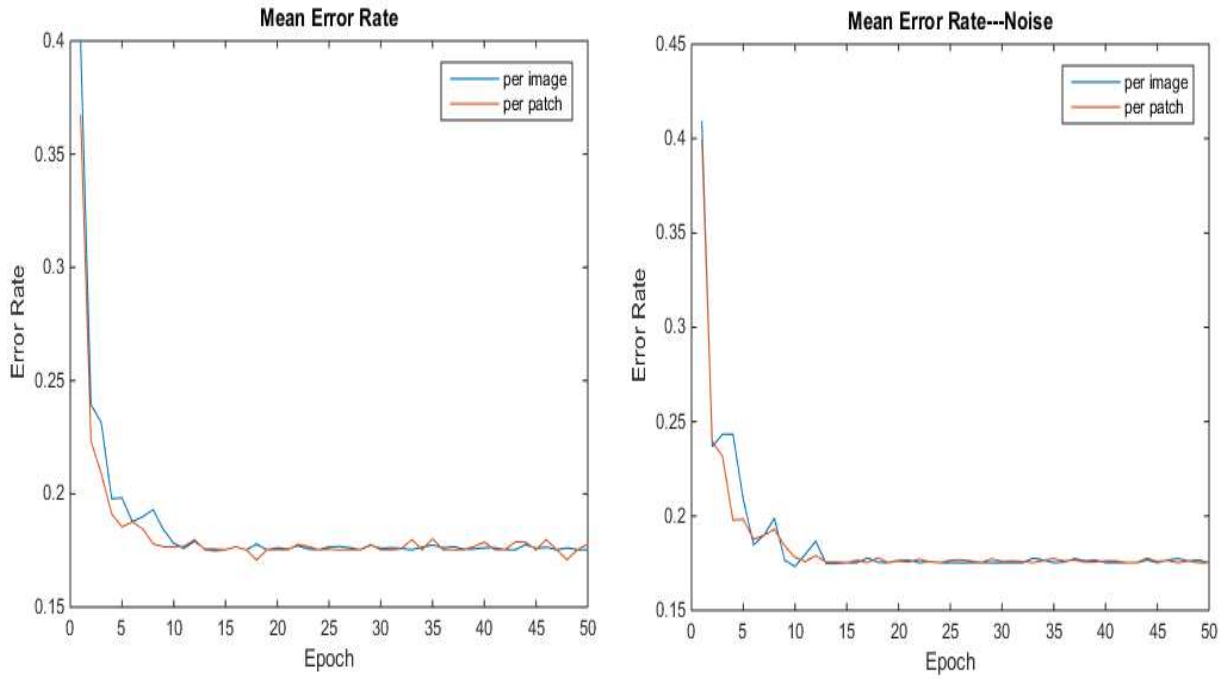


Figure 46 Detection Result Comparison in MRI Image with Noise

According to the results, it is apparent to see that the convolution neural network shows good performance in noise robustness. The major reason behind this phenomenon is that by using convolution and max pooling, the effect of noise on the images has largely been decreased. In addition, through the training process, the parameter settings for CNN architecture have been learnt, as a result, some certain abnormal data can't affect the whole performance of the CNN architecture.

5.2 CNN Training Evaluation

According to the results of the classification, it is shown that the convolution neural network actually achieved great performance in detecting the brain tumor from the MRI images. Concerning the neural network, how deep this network is becomes a major consideration when it starts to design the convolution neural network, which has a great

impact on the performance of the network. As a result, there is a testing on CNN performance on different settings within the same application case.

5.2.1 Learning Rate

There are several parameters needs consideration during the training process of CNN. Among these, the learning rate is the most important one to tune. If the learning rate is set too large, it may cause the problem of divergence, and if it is set a too small, it will result in the issue of slow speed of reaching the optimum. To make it worse, the network may get stuck in a local minima [45]. For this understanding, during the training process, the learning rate needs to be updated to reach the optimum. The figure below shows the result of the learning rate changes as the training process continues.

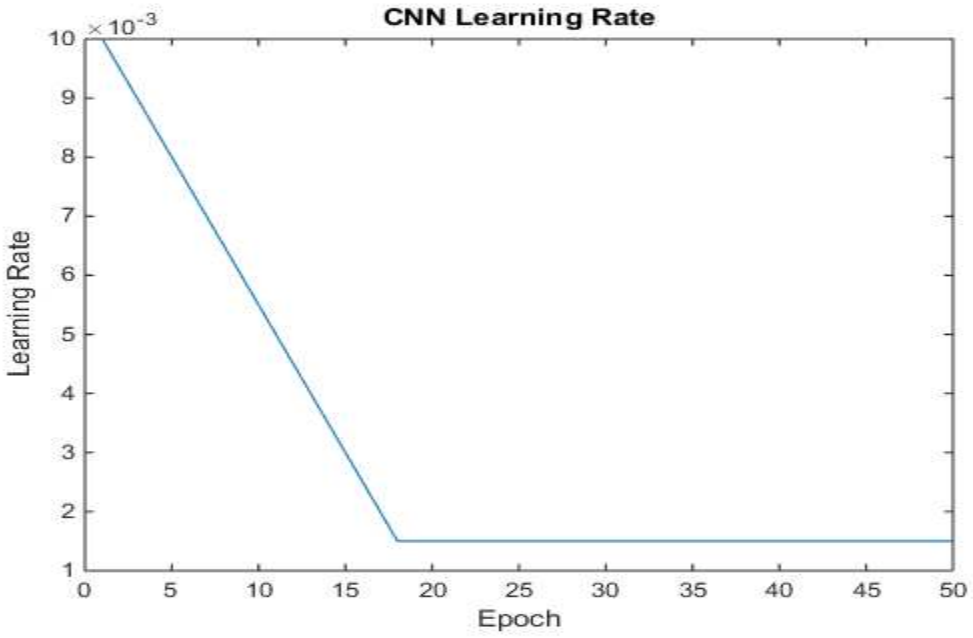


Figure 47 CNN Learning Rate

It is shown that after the 18 epochs, the learning rate become steady at 0.0015, which indicates that the learning rate has reached the optimum and won't change as the training goes on.

5.2.2 ReLU and ELU Comparison

As mentioned in the previous chapter, in this thesis, there is an improvement of using ELU function instead of ReLU for non-linear transformation. The ELU function has a hyperparameter α, which controls the saturation on negative inputs and another hyperparameter β, which controls the scale of the exponential decay. By this means, the vanishing gradient problem is alleviated and the learning speed is accelerated [44, 72].

ReLU function:

$$f(x) = \max(x, 0) \tag{5.2.1}$$

$$f'(x) = \begin{cases} 1, & x > 0 \\ 0, & otherwise \end{cases} \tag{5.2.2}$$

Revised ELU function:

$$f(x) = \begin{cases} x, & x > 0 \\ a(e^{\beta x} - 1), & otherwise \end{cases} \tag{5.2.3}$$

$$f'(x) = \begin{cases} 1, & x > 0 \\ a\beta e^{x}, & otherwise \end{cases} \tag{5.2.4}$$

There is minor restriction on the settings of these two hyperparameters, as a result, after several tests on different settings of these, a is defined as 1 and β is defined as 0.1. The figure below shows that ELU consistently outperforms ReLU. By using ReLU, the units are easy to output zero after only a few training epochs, caused by a large magnitude gradient flowing backwards through the ReLU. If this happens, the inputs are likely to be negative so that the ReLU seldom activates them in the forward pass. Since the gradient

of an inactive ReLU is zero, the unit is not able to update its input parameters. By comparison, ELU doesn't have this issue, as its gradient is non-zero for negative input, and the units can thus slowly recover with time.

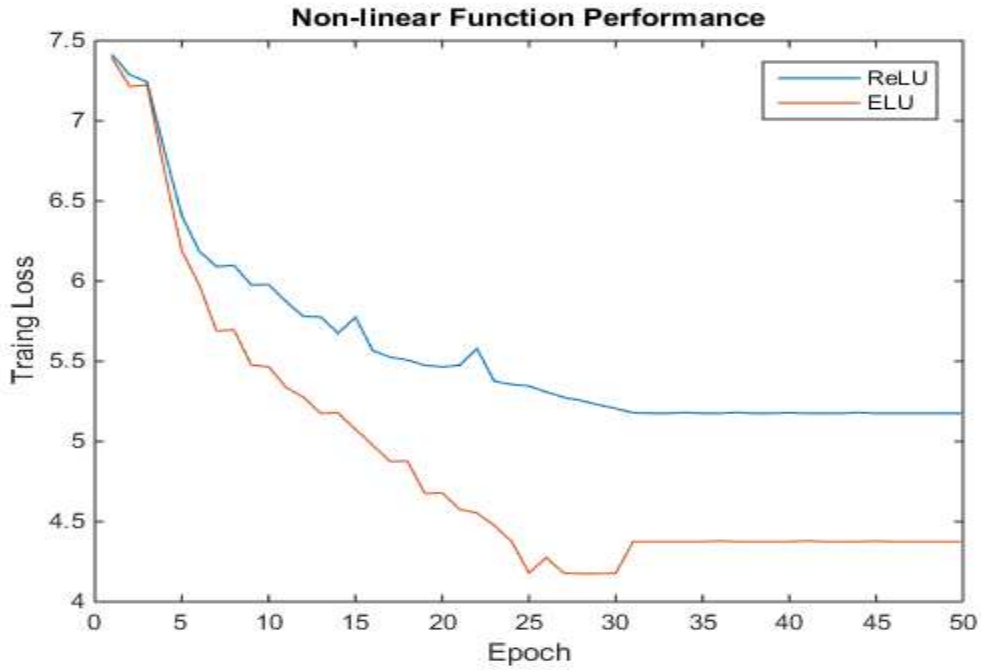


Figure 48 ReLU and ELU Learning Performance

5.2.3 CNN Performance on Different Configuration Setting

Besides the learning rate, there are other parameters needing to be tuned. In this section, several classic parameters are tested to see how they influence the final performance of CNN.

As mentioned, when designing the CNN, how deep the network should be becomes the major consideration, which involves the activities of balancing out the complexity of the CNN architecture, the cost consumed during the training process and the performance CNN appears on the application. The first consideration comes to t how many layers within the CNN architecture and how many training epochs the CNN should go through.

As there is no strict rules on both parameters, the only reference here is to see how these parameters act on the CNN performance. As the target role in this thesis is to detect the brain tumor, the accuracy of CNN on tumor detection becomes the major indictor to evaluate these parameter settings.

The figure below shows the results of different settings of the number of layers within the CNN architecture and the number of the training epochs on the performance of CNN. The number of the layers depends on how many repeats of the convolution layers, pooling layers and non-linear layers as the previous chapter mentioned. As it is shown that, after 11 layers setting and 50 epochs, the performance of CNN do not improve, which indicates that adding more layers or training epochs won't enhance the performance of CNN on detecting brain tumors.

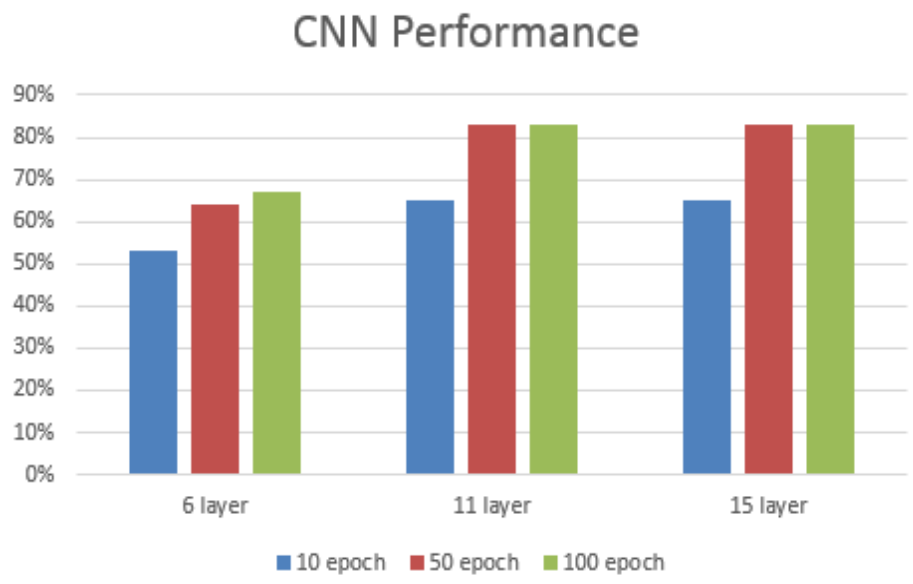


Figure 49 CNN Performance on Layer and Epoch Setting

Another aspects of parameters settings is how many filters applied in the CNN architecture. Within the convolution layer, there is still no restriction on this so that this setting needs consideration during the phase of CNN setup. In addition, convolution layer

carries on the heaviest calculation within the CNN, so the setting of the convolution kernel size also needs confirmation. There are three classic settings of kernel size: 3×3, 5×5 and 7×7.

The figure below summarizes the different settings of the number of filters and the kernel size within the convolution layer on CNN performance. According to the final results, the kernel size doesn't have too much effect on the CNN performance. As the image has been extracted into 32×32, based on the consideration of the calculation complexity, in this thesis, 5×5 is chosen as the kernel size.

Concerning the number of the filters within the convolution layers, as the figure illustrated, the 128 filters don't show any improvement of the performance, 64 filters are used within the convolution layer.

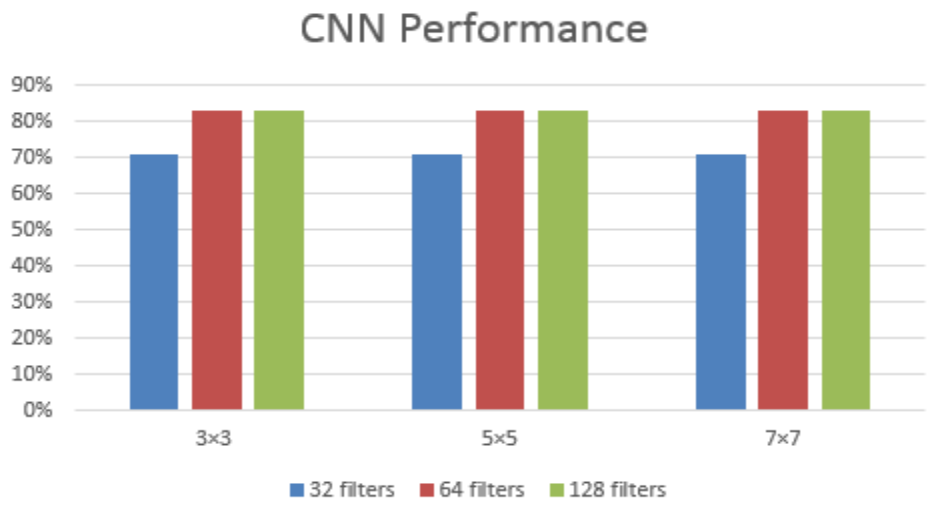


Figure 50 CNN Performance on Filter and Kernel Size Setting

According to the results of the accuracy comparison between different settings for convolution neural network, it is shown that adding the number of the layers, training epochs and filters enhances the performance of the network, however, up to a certain

level, this method won't improve the CNN performance. So during the designing phase, there should be a trade-off between the complexity of the model and its performance.

5.3 Summary

By applying the convolution neural network, it shows good performance in detecting the brain tumor within the MRI images and classifying the tissues into different categories: normal tissue, malignant tumor and benign tumor. Compared with the artificial neural network and support vector machine, CNN shows the higher accuracy of tumor detection. In addition, this chapter covers the CNN performance on different parameter settings. According to the results, it is recommended to balance out the accuracy with the efficiency when designing the CNN architecture.

6 Conclusions and Future Work

The purpose of this thesis is to detect the brain tumor from the MRI images and find out whether it is benign or malignant if the tumor exits. According to the results of the experiment, it proves that CNN acts well on accurate detection on tumor, compared with artificial neural network and other machine learning algorithm like support vector machine.

After applying convolution neural network in brain tumor detection, there are several conclusions drawn as followed:

- It is observed that convolution neural network has obtained good performance in detecting the brain tumor compared with the traditional machine learning algorithm, with the average accuracy rate over 80%. Besides, it successfully classifies the brain tumor into different categories: benign or malignant.
- By comparison of various settings of the parameters, it is clear to see that CNN shows different performance in medical image analysis. However, up to a certain level, increasing the training epochs or layers doesn't have effective impact on the performance of CNN. This implies that it is necessary to balance out the complexity and the efficiency when designing the CNN architecture.

Reviewing the development of medical images, it faces the challenges of gigantic volume of data and the requirement for smart customization. As a result, the deep learning application like CNN on medical image processing will have a bright future for its high efficiency. This thesis focuses on detecting the brain tumor within the MRI images, to extend its application, there are several points needing further improvement in future within this field so that it will meets the expectation of smart solution.

- This thesis works on the 2-dimensional MRI images. With the development of the medical equipment, there is huge volume of data encapsulated in the 3 dimensional images plus the video, so there is a need for applying deep learning on these kinds of data.
- The detection on brain tumor belongs to the categories of computer vision on object recognition. There are several other aspects within this research field: semantic segmentation, object segmentation, image context interpretation. Based on the basis of object recognition within this thesis, some extension of the related research is recommended to be studied and implemented.
- The result of CNN detecting the brain tumor within the MRI images is not good enough for clinical use with 17% error rate. As a result, in the future, there is a need of improving the successful rate in the detection and enhancing the performance of CNN at the same time.

References

[1] Selvaraj, D., and R. Dhanasekaran. "A review on tissue segmentation and feature extraction of MRI brain images." International Journal of Computer Scienceand Engineering Technology 4 (2013): 1313-1332.

[2] Arel, Itamar, Derek C. Rose, and Thomas P. Karnowski. "Deep machine learning-a new frontier in artificial intelligence research [research frontier]." *IEEE computational intelligence magazine* 5.4 (2010): 13-18.

[3] Bengio, Yoshua, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives." *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013): 1798-1828.

[4] Cheng, Bing, and D. Michael Titterington. "Neural networks: A review from a statistical perspective." *Statistical science* (1994): 2-30.

[5] Michalski, Ryszard S., Jaime G. Carbonell, and Tom M. Mitchell, eds. *Machine learning: An artificial intelligence approach*. Springer Science & Business Media, 2013.

[6] Stone, Peter, and Manuela Veloso. "Multiagent systems: A survey from a machine learning perspective." *Autonomous Robots* 8.3 (2000): 345-383.

[7] Kotsiantis, Sotiris B., Ioannis D. Zaharakis, and Panayiotis E. Pintelas. "Machine learning: a review of classification and combining techniques." *Artificial Intelligence Review* 26.3 (2006): 159-190.

[8] Bezdek, James C., L. O. Hall, and L_P Clarke. "Review of MR image segmentation techniques using pattern recognition." *Medical physics* 20.4 (1993): 1033-1048.

[9] Gavrila, Dariu M. "The visual analysis of human movement: A survey." *Computer vision and image understanding* 73.1 (1999): 82-98.

[10] Wernick, Miles N., et al. "Machine learning in medical imaging." *IEEE signal processing magazine* 27.4 (2010): 25-38.

[11] Ünal, Muhammet, et al. *Optimization of PID controllers using ant colony and genetic algorithms*. Vol. 449. Springer, 2012.

[12] Ma, Zhen, et al. "A review of algorithms for medical image segmentation and their applications to the female pelvic cavity." *Computer Methods in Biomechanics and Biomedical Engineering* 13.2 (2010): 235-246.

[13] Li, Yuanzhong, Shoji Hara, and Kazuo Shimura. "A machine learning approach for locating boundaries of liver tumors in ct images." *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*. Vol. 1. IEEE, 2006.

[14] "Deep learning in medical image analysis." *Master's thesis, Vienna University of Technology, Faculty of Informatics* (2015).

[15] Shen, Dinggang, Guorong Wu, and Heung-Il Suk. "Deep learning in medical image analysis." *Annual Review of Biomedical Engineering* 0 (2017).

[16] Hinton, Geoffrey E., Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets." *Neural computation* 18.7 (2006): 1527-1554.

[17] Litjens, Geert, et al. "A survey on deep learning in medical image analysis." *arXiv preprint arXiv:1702.05747* (2017).

[18] Akram, Saad Ullah, et al. "Cell segmentation proposal network for microscopy image analysis." *International Workshop on Large-Scale Annotation of Biomedical Data and Expert Label Synthesis*. Springer International Publishing, 2016.

[19] Akselrod-Ballin, Ayelet, et al. "A region based convolutional network for tumor detection and classification in breast mammography." *International Workshop on Large-Scale Annotation of Biomedical Data and Expert Label Synthesis*. Springer International Publishing, 2016.

[20] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." *Nature* 521.7553 (2015): 436-444.

[21] Schmidhuber, Jürgen. "Deep learning in neural networks: An overview." *Neural networks* 61 (2015): 85-117.

[22] Ngiam, Jiquan, et al. "On optimization methods for deep learning." *Proceedings of the 28th international conference on machine learning (ICML-11)*. 2011.

[23] Bengio, Yoshua. "Learning deep architectures for AI." *Foundations and trends® in Machine Learning* 2.1 (2009): 1-127.

[24] Nagalkar, V. J., and S. S. Asole. "Brain tumor detection using digital image processing based on soft computing." *Journal of signal and image processing* 3.3 (2012): 102-105.

[25] Murugavalli, S., and V. Rajamani. "An Improved Implementation of Brain Tumor Detection Using Segmentation Based on Neuro Fuzzy Technique 1." (2007).

[26] Mustaqeem, Anam, Ali Javed, and Tehseen Fatima. "An efficient brain tumor detection algorithm using watershed & thresholding based segmentation." *International Journal of Image, Graphics and Signal Processing* 4.10 (2012): 34.

[27] Wu, Ming-Ni, Chia-Chen Lin, and Chin-Chen Chang. "Brain tumor detection using color-based k-means clustering segmentation." *Intelligent Information Hiding and Multimedia Signal Processing, 2007. IIHMSP 2007. Third International Conference on*. Vol. 2. IEEE, 2007.

[28] Prastawa, Marcel, et al. "A brain tumor segmentation framework based on outlier detection." *Medical image analysis* 8.3 (2004): 275-283.

[29] Ratan, Rajeev, Sanjay Sharma, and S. K. Sharma. "Brain tumor detection based on multi-parameter MRI image analysis." *ICGST-GVIP Journal* 9.3 (2009): 9-17.

[30] Kowar, Manoj K., and Sourabh Yadav. "Brain tumor detection and segmentation using histogram thresholding." *International Journal of Engineering and Advanced Technology* 1.4 (2012): 16-20.

[31] Hall, Lawrence O., et al. "A comparison of neural network and fuzzy clustering techniques in segmenting magnetic resonance images of the brain." *IEEE transactions on neural networks* 3.5 (1992): 672-682.

[32] Dahab, Dina Aboul, Samy SA Ghoniemy, and Gamal M. Selim. "Automated brain tumor detection and identification using image processing and probabilistic neural network techniques." *International journal of image processing and visual communication* 1.2 (2012): 1-8.

[33] Joshi, Dipali M., N. K. Rana, and V. M. Misra. "Classification of brain cancer using artificial neural network." *Electronic Computer Technology (ICECT), 2010 International Conference on*. IEEE, 2010.

[34] Kadam, Deepak Bhimrao, et al. "Neural network based brain tumor detection using MR images." *International Journal of Computer Science and Communication* 2.2 (2011): 325-331.

[35] Damodharan, Selvaraj, and Dhanasekaran Raghavan. "Combining Tissue Segmentation and Neural Network for Brain Tumor Detection." *International Arab Journal of Information Technology (IAJIT)* 12.1 (2015).

[36] John, Pauline. "Brain tumor classification using wavelet and texture based neural network."*International Journal of Scientific & Engineering Research* 3.10(2012): 1-7

[37] Zikos, M., E. Kaldoudi, and S. Orphanoudakis. "Medical image processing." *Stud. Health Technol. Inf.* 43.Pt B (1997): 465-469.

[38] Zhu, Hongmei. "Medical image processing overview." *University of Calgary* (2003).

[39] Jannin, Pierre, et al. "Validation of medical image processing in image-guided therapy." *IEEE Transactions on Medical Imaging* 21.12 (2002): 1445-9.

[40] Lehmann, Thomas Martin, Claudia Gonner, and Klaus Spitzer. "Survey: Interpolation methods in medical image processing." *IEEE transactions on medical imaging* 18.11 (1999): 1049-1075.

[41] McAuliffe, Matthew J., et al. "Medical image processing, analysis and visualization in clinical research." *Computer-Based Medical Systems, 2001. CBMS 2001. Proceedings. 14th IEEE Symposium on*. IEEE, 2001.

[42] Lawrence, Steve, et al. "Face recognition: A convolutional neural-network approach." *IEEE transactions on neural networks* 8.1 (1997): 98-113.

[43] Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *European conference on computer vision*. Springer, Cham, 2014.

[44] Jia, Yangqing, et al. "Caffe: Convolutional architecture for fast feature embedding." *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014.

[45] Crounse, Kenneth R., and Leon O. Chua. "Methods for image processing and pattern formation in cellular neural networks: A tutorial." *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 42.10 (1995): 583-601.

[46] LeCun, Yann, Koray Kavukcuoglu, and Clément Farabet. "Convolutional networks and applications in vision." *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*. IEEE, 2010.

[47] Vedaldi, Andrea, and Karel Lenc. "Matconvnet: Convolutional neural networks for matlab." *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 2015.

[48] Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

[49] Zacharaki, Evangelia I., et al. "Classification of brain tumor type and grade using MRI texture and shape in a machine learning scheme." *Magnetic resonance in medicine* 62.6 (2009): 1609-1618.

[50] Sachdeva, Jainy, et al. "Segmentation, feature extraction, and multiclass brain tumor classification." *Journal of digital imaging* 26.6 (2013): 1141-1150.

[51] Zhang, Nan, et al. "Kernel feature selection to fuse multi-spectral MRI images for brain tumor segmentation." *Computer Vision and Image Understanding* 115.2 (2011): 256-269.

[52] Joshi, Dipali M., N. K. Rana, and V. M. Misra. "Classification of brain cancer using artificial neural network." *Electronic Computer Technology (ICECT), 2010 International Conference on*. IEEE, 2010.

[53] Mohanaiah, P., P. Sathyanarayana, and L. GuruKumar. "Image texture feature extraction using GLCM approach." *International Journal of Scientific and Research Publications* 3.5 (2013): 1.

[54] Hua, B. O., Ma Fu-Long, and Jiao Li-Cheng. "Research on computation of GLCM of image texture [J]." *Acta Electronica Sinica* 1.1 (2006): 155-158.

[55] Zulpe, Nitish, and Vrushsen Pawar. "GLCM textural features for brain tumor classification." *IJCSI International Journal of Computer Science Issues* 9.3 (2012): 354-359.

[56] Kotsiantis, Sotiris B., I. Zaharakis, and P. Pintelas. "Supervised machine learning: A review of classification techniques." (2007): 3-24.

[57] Bhattacharyya, Siddhartha. "A brief survey of color image preprocessing and segmentation techniques." *Journal of Pattern Recognition Research* 1.1 (2011): 120-129.

[58] Bow, Sing T., ed. *Pattern recognition and image preprocessing*. CRC press, 2002.

[59] Ponraj, D. Narain, et al. "A survey on the preprocessing techniques of mammogram for the detection of breast cancer." *Journal of Emerging Trends in Computing and Information Sciences* 2.12 (2011): 656-664.

[60] Park, Jong Geun, and Chulhee Lee. "Skull stripping based on region growing for magnetic resonance brain images." *NeuroImage* 47.4 (2009): 1394-1407.

[61] Hartley, S. W., et al. "Analysis and validation of automated skull stripping tools: a validation study based on 296 MR images from the Honolulu Asia aging study." *NeuroImage* 30.4 (2006): 1179-1186.

[62] Vincent, Luc. "Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms." *IEEE transactions on image processing* 2.2 (1993): 176-201.

[63] Maragos, Petros. "Tutorial on advances in morphological image processing and analysis." *Proc. SPIE*. Vol. 707. 1986.

[64] Chan, Tony F., and Luminita A. Vese. "Active contours without edges." *IEEE Transactions on image processing* 10.2 (2001): 266-277.

[65] Park, HyunWook, Todd Schoepflin, and Yongmin Kim. "Active contour model with gradient directional information: Directional snake." *IEEE Transactions on Circuits and systems for video technology* 11.2 (2001): 252-256.

[66] Haralick, Robert M., and Linda G. Shapiro. "Image segmentation techniques." *Computer vision, graphics, and image processing* 29.1 (1985): 100-132.

[67] Ciresan, Dan C., et al. "Flexible, high performance convolutional neural networks for image classification." *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*. Vol. 22. No. 1. 2011.

[68] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.

[69] Hjelmås, Erik, and Boon Kee Low. "Face detection: A survey." *Computer vision and image understanding* 83.3 (2001): 236-274.

[70] Guo, Yanming, et al. "Deep learning for visual understanding: A review." *Neurocomputing* 187 (2016): 27-48.

[71] Abdel-Hamid, Ossama, Li Deng, and Dong Yu. "Exploring convolutional neural network structures and optimization techniques for speech recognition." *Interspeech*. 2013.

[72] Aghdam, Hamed Habibi, and Elnaz Jahani Heravi. *Guide to Convolutional Neural Networks: A Practical Application to Traffic-Sign Detection and Classification*. Springer, 2017.

[73] Han, Song, et al. "Learning both weights and connections for efficient neural network." *Advances in Neural Information Processing Systems*. 2015.