# Application of multi- path CNN for brain tumor segmentation from MRI

**By**

Samia Chowdhury
ID: 151-35-902
&
Sm Shahjalal Shaju
ID: 151-35-1036

A thesis submitted in partial fulfillment of the requirement for the degree of Bachelor of Science in Software Engineering

**Department of Software Engineering**
**DAFFODIL INTERNATIONAL UNIVERSITY**

Fall – 2018

# APPROVAL

The "**Application of multi- path CNN for brain tumor segmentation from MRI**", submitted by **Samia Chowdhury**, **ID: 151-35-902** and **Sm Shahjalal Shaju, ID: 151-35-1036** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

## BOARD OF EXAMINERS

--------------------------------------------

**Dr. Touhid Bhuiyan**
**Professor and Head** — **Chairman**
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

--------------------------------------------

**K. M. Imtiaz-Ud-Din**
**Assistant Professor** — **Internal Examiner 1**
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

--------------------------------------------

**Asif Khan Shakir**
**Lecturer** — **Internal Examiner 2**
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

--------------------------------------------

**Dr. Md. Nasim Akhtar**
**Professor** — **External Examiner**
Department of Computer Science and Engineering
Faculty of Electrical and Electronic Engineering
Dhaka University of Engineering & Technology, Gazipur

# DECLARATION

It is hereby declared that this thesis has been done by Samia Chowdhury and Sm Shahjalal Shaju under the supervission of Md. Anwar Hossen, Lecturer (Senior Scale), Department of Software Engineering, Daffodil International University. It is also declared that neither this thesis nor any part of this has been submitted elesewhere for award or any degree.


**Samia Chowdhury**

**Student ID: 151-35-902**

Batch: 16th

Department of Software Engineering

Faculty of Science & Information Technology
Daffodil International University

**Sm Shahjalal Shaju**

**Student ID: 151-35-1036**

Batch: 16th

Department of Software Engineering

Faculty of Science & Information Technology
Daffodil International University


Certified by:

22.12.2018


**Md. Anwar Hossen**

**Lecturer (Senior Scale)**

Department of Software Engineering

Faculty of Science & Information Technology

**Daffodil International University**

## ACKNOWLEDGEMENT

In this very special moment, first and foremost we are grateful to the almighty Allah for the good health and wellbeing that were necessary to complete this dissertation.

We would like to extend sincere gratitude to our supervisor Md. Anwar Hossen for his continuous support during our undergraduate thesis study and research. We are extremely thankful and indebted to him for sharing his expertise, and sincere and valuable guidance and encouragement extended to us. Without his assistance and dedicated involvement in every step throughout the process, this paper would have never been accomplished.

We would also like to take the opportunity to express gratitude to all of the department faculty members and our department head Professor Dr. Touhid Bhuiyan for their help and support.

Last but not the least, we would like to thank our parents for the never ending support and encouragement.

# ABSTRACT

The automation of brain tumor segmentation from MRI is an active topic in the field of medical research. Different approaches and methods are being proposed throughout the years to address this challenging task. The application of convolutional neural network has caught the attention of many researchers for the solution of this particular problem due to its extraordinary performances in the field of computer vision. Many of the state-of-the-art techniques use different approaches based on CNN. One of such approaches is the multi-path CNN architecture.

In this paper, we propose a novel multi-path CNN architecture that allows flow of information in two different pathways resulting in the exploitation of both local and global features simultaneously, hence this architecture can also be called two-pathway architecture. We use this architecture to train on the dataset collected from BRATS 2013 challenge. Our model when tested on BRATS 2013 training images, showed on an average 95.654% accuracy.

***Keywords***: Brain tumor, MRI, convolutional neural network, segmentation

# Table of Contents

# List of Figure

## List of Table

## Chapter 1

# Introduction

### 1.1 Overview

Magnetic resonance imaging (MRI) of the brain is a process to produce detailed and comprehensive images of the brain and the brain stem using a magnetic field and radio waves. It is used to diagnose many brain conditions that include tumors, lesions, swelling etc.

A tumor is an abnormal growth that may occur in different parts of the body. It can be both noncancerous (benign) and cancerous (malignant).Similarly a brain tumor is a cancerous or noncancerous growth in the brain which may have originated in the brain or has spread to the brain from any other part of the body.

Brain tumor segmentation means to separate the different tumor tissues (solid or active tumor, edema and necrosis) from normal brain tissues. In healthy brain there are normally three types of tissues - Gray Matter (GM), White Matter (WM) and Cerebrospinal Fluid (CSF). In tumor studies, there are some tumors that can be segmented easily e.g. meningiomas whereas some others like gliomas and glioblastoas are difficult to localize. Various factors like diffusion, poor contrast, variability in shapes, structures, sizes, location etc. make the segmentation of these tumors difficult.

It is admitted that the most accurate segmentation of such medical images can be done by human experts. But the manual process is tedious, expensive and time-consuming. Moreover, in case of larger studies when experts have to handle a large number of cases with the same accuracy i.e. keeping the results unaffected by factors like fatigue, missing manual steps, inter-operator variability etc. is almost impractical. This is where the necessity for automation of segmentation comes in. Nowadays high-speed computers are available at a moderate cost. The goal of automated segmentation process is to generate fast and accurate results and allow to perform analysis over large number of data in a very short time.

Application of convolutional neural network is a deep learning approach to address this task. Deep learning is a computer science branch that deals with the algorithms that can learn. It is a subfield of machine learning inspired by Artificial Neural Networks (ANN), which are based on biological neural networks. Different real life applications such as image classification, object detection, facial recognition, segmentation, self-driving cars etc. are the examples of success of CNN

In this paper we are going to implement multi-path CNN architecture to perform tumor segmentation from 3D brain MRI. Multi-path CNN architecture means processing different resolutions of images at a time by feeding them into convolutional neural network layers. More details about this architecture will be described in Chapter 3 (Methodology).

## 1.2 Contribution

Medical image analysis is a vast field of research at present. One of the efficient CNN based approaches to perform tumor segmentation from brain MRI is the multi-path CNN architecture. For our research, we have considered the paper "Brain Tumor Segmentation with Deep Neural Networks" (Mohammad Havaei, 2016)[1] as the baseline of our work. We implemented the two-pathway CNN model as described in the above mentioned paper and fine tuned the model with necessary changes in order to improve accuracy on that particular model. The accuracy score of our paper meets the standards of the state-of-art- methods implemented to address the same task. The score is 95.654%.

## 1.3 Motivation

It needs no telling that one of the most impactful innovations in the field of computer vision is Convolutional Neural Network. In comparison with the traditional computer vision, it has shown record-shattering performances in various fields including image segmentation. It has proven to be one of the most efficient models to perform the difficult and challenging task of medical image analysis in recent times. It has bright prospects to reach many significant milestones in the future. This is the reason why we have taken interest to explore in this particular field.

## 1.4 Organization of the Thesis

The dissertation is organized as follows:

**Chapter 1 Introduction:** In this chapter an over view of brain tumor, MRI, tumor segmentation using convolutional neural network is given. It also includes contribution and motivation.

**Chapter 2 Literature Review:** This chapter includes all the important topics from background study and related works are discussed.

**Chapter 3 Methodology:** This chapter describes in details the proposed model architecture and data preprocessing.

**Chapter 4 Implementation & Result:** This chapter describes the implementation environment, training and testing procedure, result of the experiment and its analysis.

**Chapter 5 Conclusion:** It tells in brief the constraints of present work and future plan

# Chapter 2

## Literature Review

### 2.1 Background Study

#### 2.1.1 Deep Learning

At present Artificial Intelligence and machine learning are the fields undergoing intense study. Before we jump into what deep learning is, let's have a look at the definitions of these important terms.

Artificial Intelligence: It refers to the replication of human intelligence in computers.

Machine Learning: The ability of a machine to learn itself from a large amount of data instead of rules that are hard coded.

From the basic definitions of machine learning, we know algorithms that parse data, learn from the data and then apply what they have learned to make informed decisions fall under machine learning. Deep learning is a subset of machine learning. Machine learning is subset of artificial intelligence. Though deep learning is similar to machine learning as it functions in the same way but its capabilities are different than that of machine learning. It teaches computers to do the things that come naturally to humans- learning by example. It examines computer algorithms that learn and improve on their own. We have seen that Facebook automatically finds and tags friends in photos. Google Deepmind's AlphaGo computer program trounced champions at the ancient game of Go last year. Skype translates spoken conversations in real time and pretty accurately too. Behind all this is deep learning. The reason deep learning attracted so much attention is that it has the potential to be very useful for real-world applications, for example, self-driving cars. Deep learning networks can be successfully applied to big data for knowledge discovery, knowledge application and knowledge-based prediction. In other words, for producing actionable results, deep learning is a powerful engine.

Recently deep learning has reached to a point from where it can outperform humans in task such as object classification or recognition in images.

### 2.1.2 Neural Network Concepts

The human brain is considered as a type of computer by modern neuroscientists. Neural networks are implemented in order to do the reverse, that is, to build a computer that works like a human brain. We have only a limited understanding of how the brain functions as it is very complex. But a computer can be built the functioning of which will be very different from a normal computer if we can simplify the simulation of how a brain processes data.

A typical brain consists of hundreds of billions nerve cells called neurons. A neuron is composed of three parts: A **cell body**, the central mass of the neuron. From it comes off a number of connections that are known as **dendrites**. There are numerous dendrites that carry information towards the cell body (a cell's input). And finally there is a single **axon** in a neuron which carries information (a cell's output) away. So basically there are innumerable neurons in a human brain that takes information, processes it and transmits the output to another neuron. The human brain learns, analyses and make decisions with the help of these interconnected neurons. This is the basic idea for a neural network.

### 2.1.3 The components of a neural network

A typical artificial neural network consists of a few to almost millions of neurons arranged in a series of layers, each of which connects to the layers on either side. The first layer, known as the input layer take various forms of information from the outside world that the network aims to learn or recognize. The final layer is known as the output layer which gives out its response to the information it has learned. In between there are one or more layers known as the hidden layers. The neurons of these layers are known as the units. In majority of the artificial neural network, most of the units are connected to the units of the layers on either side,

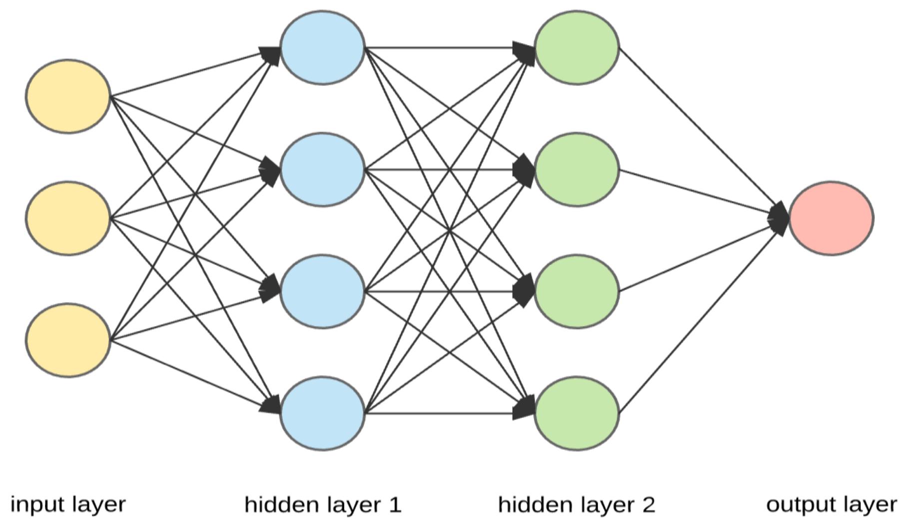


Figure 1: Neural network

The connection represented between two such unit is known as the weight which can be both a positive or negative number. The higher the weight, the more influence one unit has on another.

### 2.1.4 How does a neural network learn

#### 2.1.4.1 Feedforward network

When information are fed into the units of the input layer, the layers of hidden units are triggered which in turn arrive at the output units. This is known as a feed-forward network. Each unit receives input from the units at its left which are multiplied by the weights of the edges they travel along . Every unit adds up all the inputs it receives this way and if the sum is more than a threshold value, the unit fires (performs activation function e.g. sigmoid function, ReLu etc.) and triggers all the units it is connected to at the right side. This way the output units initially give out an output which is arbitrary and nonsensical. At this point we have to give a feedback to the network that what the actual output should have been and what it has calculated. So, the network can figure out the difference and try to make necessary changes. This way of learning with the help of feedback is known as back propagation.

#### 2.1.4.2 Backpropagation

During back propagation, the network changes or updates its weight value using the difference between desired and given output working from the output units through the hidden layers to the input units. Its aim is to reduce the difference between actual and intended output where the two exactly coincide. As long as it does not happen, the network keeps going backward to update its weights. This is how a neural networks learn.

#### 2.1.4.3 Loss and optimizer

The difference between actual and intended output is known as the loss. The function used to measure the inconsistency between the actual output (y) and the intended output(^y). It is a positive value that when decreases, increases the robustness of the model. The main aim of a neural network is to minimize this loss. This is where the optimization algorithms works.

We know, the neural network learns by updating its weights, hence weights are called the learnable parameters. The optimization algorithms and strategies update and calculate appropriate and optimum learnable parameters that influence the model's learning process and the output of the model. There are different optimization algorithms used in neural networks e.g. Gradient descent, Stochastic gradient descent, Adam etc.

### 2.1.5 Convolutional Neural Network

Possibly the most popular deep learning architecture is the convolutional neural networks.CNN is a deep, feed forward artificial neural network that are inspired by the biological visual cortex. The cortex has small regions of the cells that are sensitive to specific areas of the visual field. In 1962, Hubel and Wiesel in an experiment showed that some neurons in the brain are activated or fired only in the presence of edges of a particular orientation like horizontal or vertical edges. Hubel and Wiesel found that all of these neurons are well ordered in columnar fashion and together they were able to produce visual perception. This idea of specialized components inside of a system having specific tasks is found in CNNs.

CNNs are very similar to that of the regular neural networks. They are also made up of neurons and have weights and biases. Loss function, optimizers everything work in the same way as the regular neural network.

### 2.1.6 Differences between a regular neural network and CNN layers

Unlike regular networks, the convolutional neural networks have layers of convolution units in specific. The architecture of the layers of CNN are different than that of neural networks. The layers of CNN that are different are given below:

Input layer: This layer contains the raw pixel values of the images.

Convolutional layer: This layer computes the dot product of the weight matrix with the portion of the image that the kernel convolves. The output of a convolutional layer in known as the feature map.

Pooling layer: It performs a downsampling operation along the spatial dimensions resulting in the decrease of input dimensions.

Fully connected layer: This layer computes the class score where each neuron is connected to all the neurons in the previous layer

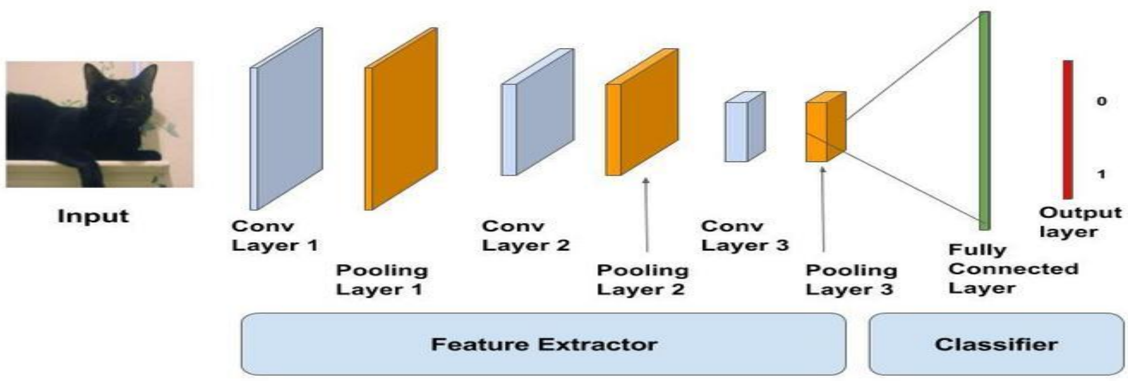


Figure 2: Convolutional Neural Network

So we can see how the convolutional neural network differs from that of a regular neural network.

### 2.1.7 How CNN works

In this section we will see how a CNN works for image classification in details.

#### 2.1.7.1 Convolution layer

The first operation performed in the CNN is convolution which is done in the convolutional layers. The input to a CNN model is an image. The raw pixels of the image are taken as the input units of the input layer. Instead of weight, the terms kernel or weight matrix is used. Kernel is a matrix of weights of defined size. This is also known as the feature detector. This kernel convolves over the input image. So, there are three elements that enter into the convolution- input image, feature detector/kernel, feature maps.

Let us consider we have a 7x7 input image and the kernel size is 3x3. We will now put the kernel over the input image and we begin from the top-left corner. In the kernel there are total 9 (3x3) cells. After we put the kernel over the input image, the 9 cells of the kernel multiply with corresponding 9 cells of the input image. The output of this operation is the sum of all the products. Thus we can see that the output is actually the dot product of the kernel and its corresponding input image patch. Next the kernel slides one cell to the right and perform the same operation and the process keeps on repeating as long as the whole input image is not slided over by the kernel. After the kernel convolves over the entire input image, we get a downsampled image with different pixel values. This is known as the feature map. It is to be noted that the shift of cells when we convolve the kernel over the input image is known as stride. When we move one cell right, the value of stride is 1. If we move two cells right, the stride value is 2 and so on. We can visualize the description in Figure 3.

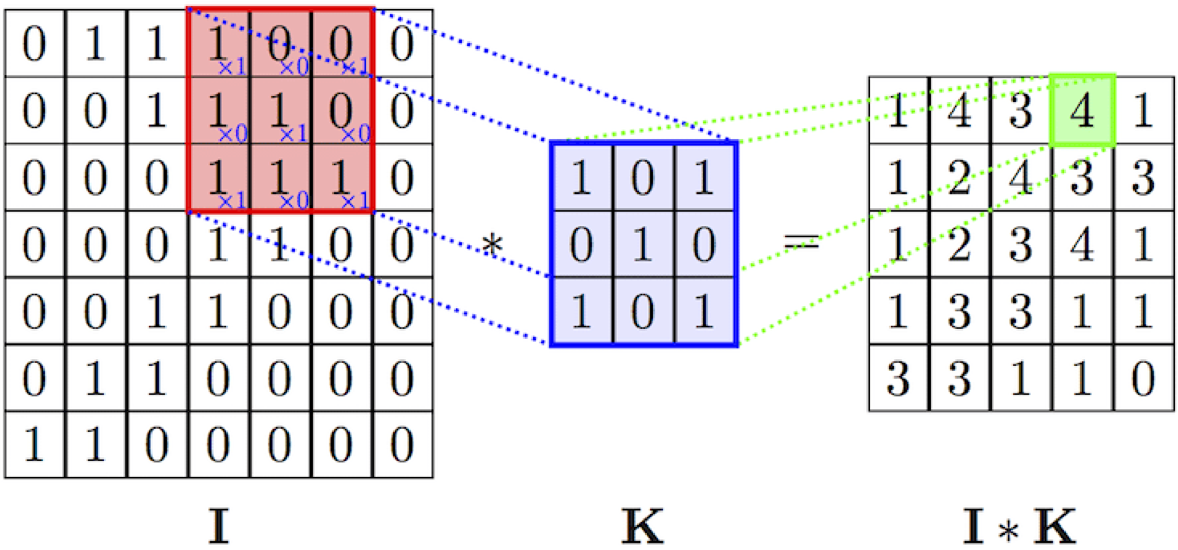

Figure 3: Convolution operation

In the above figure the " * " represents dot product.

The convolution operation has downsampled the image. The larger stride we take , the smaller the feature map. No information is lost when the original image is downsampled. Because the kernel sifts through the information in the in the input image and filter the parts that are integral to it and exclude the rest.

This operation helps to detect various features- simple features like edges in the initial layers and as the model gets deeper, more sophisticated features are learnt.

**2.1.7.2 Pooling layer**

There are three different types of pooling-Mean, max and sum pooling. We will discuss about max-pooling. The pooling layer also uses a kernel or filter of a definite size. The filter then slides over the input image but like convolution, it doesn't perform dot product of the cells. It picks out the maximum value from the cells that correspond to the filter. Let us consider a

4x4 input (feature map from convolution layer) and 2x2 filter. We can see the pooled feature map in figure 4.

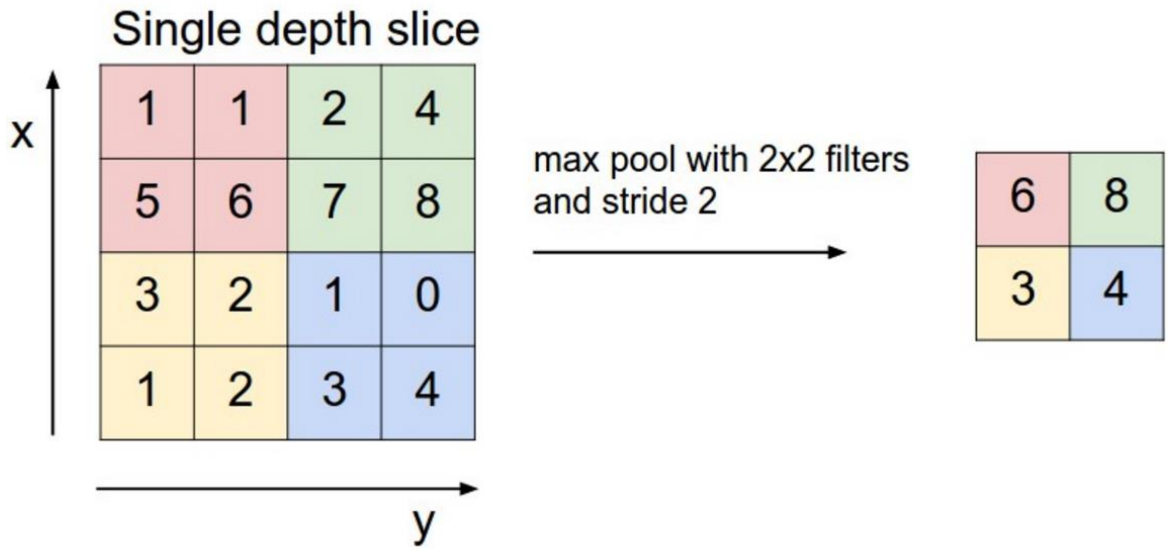


Figure 4: Max pooling

The max pooling layer teaches the network a property "spatial variance". It makes the network capable to detect an object in an image without being confused by the difference in textures, distance or angles of the image.

**2.1.7.3 Zero padding layer**

Zero padding layer is used to add rows and columns of zeros at the top, bottom, left and right side of an image. This is often done to increase the size of a downsampled image to match the input size.

**2.1.7.4 Flattening**

Another operation performed in a convolutional neural network is flattening. After the convolution and pooling operation, the pooled feature map is flattened into a column because we are going to need this data to feed in our neural network for further processing.

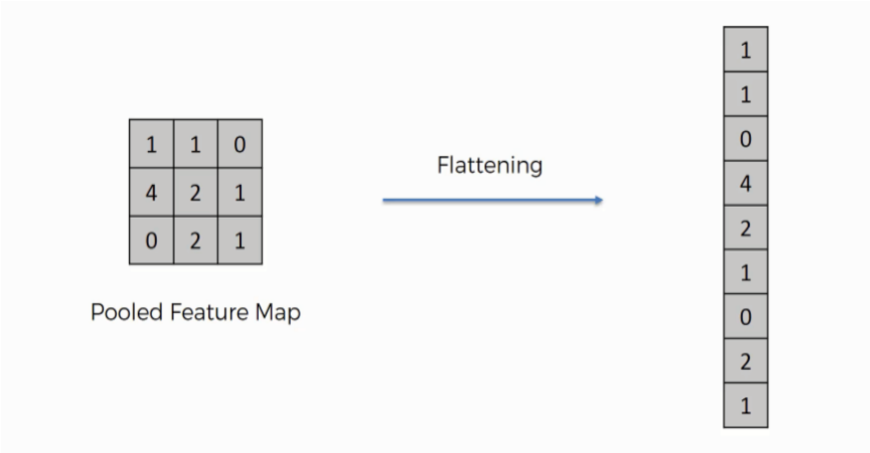


Figure 5: Flatten

#### 2.1.7.5 Fully connected layer

After flattening, comes the fully connected layer. This is the artificial neural network part in our CNN. The input to this layer is the flattened pooled feature map that we extracted from the previous steps. The purpose of this layer is to take the data and combine the features into a wider variety of attributes that make the convolutional neural network more capable of classifying images. The model performs classification after this layer. Hence calculating loss function, optimizing everything is done in this step which allows a network to predict accurately. The visualization of this layer for a dog/cat classifier is shown in figure 6.

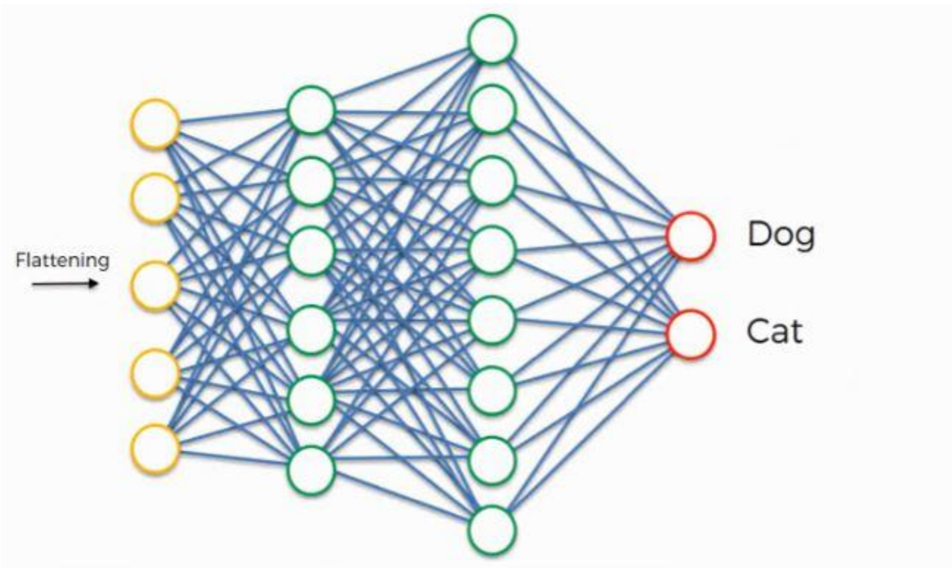


Figure 6: Fully connected layer

#### 2.1.7.6 Classifier

For the classification, after the fully connected layer a function is used which is known as a classifier that outputs the probability for multiple labels. It is usually used in the last layer of the convolutional neural network. The softmax classifier generates a probability distribution for each class. Its output values add up to 1.

#### 2.1.7.7 ReLU activation

In neural networks, the output of a node/neuron is defined by the activation function for a given input or a set of inputs. The output then in given as input to the next neuron and it keeps on going until a desired solution to the original problem is found. There are different types of activation function, for example sigmoid function, ReLU function etc.

ReLU stands for Rectified linear unit. The function can be written as below:

$f(x) = \max(0,x)$

It outputs a zero for a negative number input and if the input is positive it outputs the input itself. This function is often used after a convolution layer in CNN. The advantage of using this function is that it speeds up training since the computation is very simple and easy.

### 2.1.8 Benefits of CNN

I. They are able to capture relevant features from an image/video at different levels similar to a human brain that a regular artificial neural network cannot perform.

II. Compared to regular NN, CNN produces lesser number of parameters. That makes CNN more efficient in terms of memory and complexity.

III. In applications such as image, video, text and speech processing, it is important to consider the context or shared information in small neighborhoods which is done by CNN.

IV. In terms of performance, CNN is better than a regular NN for image recognition.

### 2.1.9 Different learning methods

There are three broad categories of learning methods in neural networks. These are:

I. Supervised learning

II. Unsupervised learning

III. Semi supervised learning

When a neural network model is trained with sample data and their corresponding labels so that it learns the relationship and dependencies between the target prediction and input features. For example, if we train a model to classify cats and dogs images we need to provide images of cats and dogs to the model, along with the images, we also provide the label of the images (0-cat/1-dog) then this is a supervised learning.

In unsupervised learning the model is trained with unlabeled data. In this method, the model looks for rules, detect patterns and summarize and group the data points that give meaningful information to the user. These are mostly useful when the human expert does not know what to look for in the data.

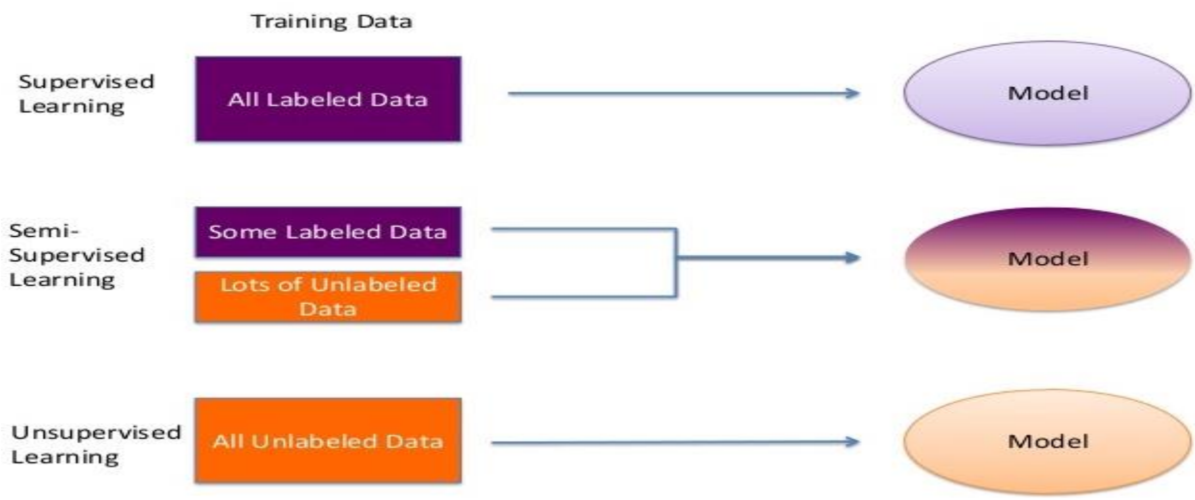


Figure 7: Different learning methods

In semi supervised learning method, the model is trained with both labeled and unlabeled data. When labeling data is expensive for requiring an human expert and most of the data are unlabeled but there are few labeled data as well, this method is very suitable.

### 2.1.10 Segmentation

We know that digital images are actually composed of pixels. Segmentation is the process of partitioning an image into multiple sets of pixels (segments). This segmentation is done on the basis of similarity among the pixels in different factors like shape, color, intensities etc. The purpose of performing segmentation is to simplify or change an image representation into something more meaningful and easier to analyze. There can be different types of segmentation like semantic segmentation, instance segmentation etc. Semantic segmentation means per-pixel mark-up of the image where each label corresponds to a specific object. On the other hand, in instance segmentation, all individual instances of objects of a given class are labeled and each object is marked with its own id.

Fig 8(a) Semantic Segmentation Fig 8(b): Instance Segmentation

### 2.1.11 Segmentation using CNN

We have seen in the previous section how CNN performs classification. Segmentation is also a kind of classification. In case of classification, CNN outputs a label for a given input image whereas in the task of segmentation, the network generates a label for each pixel in the given input image. Thus segmentation is nothing but pixel-wise classification. We are going to perform semantic segmentation in this paper.

## 2.2 Related Work

Different algorithms and approaches have been proposed to automate the segmentation from brain MRI. Before convolutional neural network, there are many methods that were in use to perform the same task. FreeSurfer, one of the most commonly used automatic brain segmentation tool that uses non-linear registration and an atlas- based segmentation approach (Fischl et al.,2002)[2], FIRST (Patenaude et al., 2011)[3] based on principles of Active Shape (Cootes et al., 1995)[4] and Active Appearance Models (Cootes et al., 2001)[5] etc. are some of the examples.

It is not too long since the convolutional neural network has emerged as an appealing solution to address this challenging task. The recent advancement in the application of CNN has mainly been possible of two factors: large number of labeled data and high computational power of computers. We know, in today's world due to billions of people interacting over the internet enormous amount of data are being generated. Moreover, the availability of high-performance computers at a moderate cost has made it even easier.

Seminal work on CNN for segmentation in brain MRI date back to 2015 when Zhang et al. (2015)[6] proposed a CNN model to perform fast brain tissue segmentation on MRI. Since then many more sophisticated proposals are being put forward. Some of the significant state-of-the-art CNN approaches include single-path architecture as well. "Deep MRI brain extraction : A 3D convolutional neural network for skull stripping" (Kleesiek et al.,2016)[7] , "Adversarial Training and Dilated Convolutions for Brain MRI Segmentation" (Moeskops et al., 2017)[8] etc. implemented single-path architecture.

Multi-path architecture came out as an efficient solution to perform better segmentation results by incorporating both local and contextual information from more than one processing pathways. The main inspiration of our work is taken from the paper "Brain Tumor Segmentation With Deep Neural Networks" (Mohammad Havaei, 2016)[1] . In this paper the authors implemented a two-pathway CNN architecture for the segmentation task. Later in 2017, *deepmedic*, a software proposed in "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation" (Kamnitsas et al., 2017)[9] that exploited multi-scale processing via parallel convolutional pathways has also set a benchmark.

## Chapter 3

# Methodology

### 3.1 Dataset

For the implementation of the method we proposed in this paper, we have used BRATS 2013 3D training dataset. This dataset consists of both real patient images and synthetic images. Every image is stored as a folder which is then subdivided into two folders- High Grade (HG) and Low Grade (LG) images. Each of these folders have five images. The first four images are the four modalities of the MRI: T1, T1-C, T2 and FLAIR. The fifth image is the ground truth for each pixel of the image. The dimension of the HG images are (176,261,160) and that of LG images are (176,196,216).

Sample:

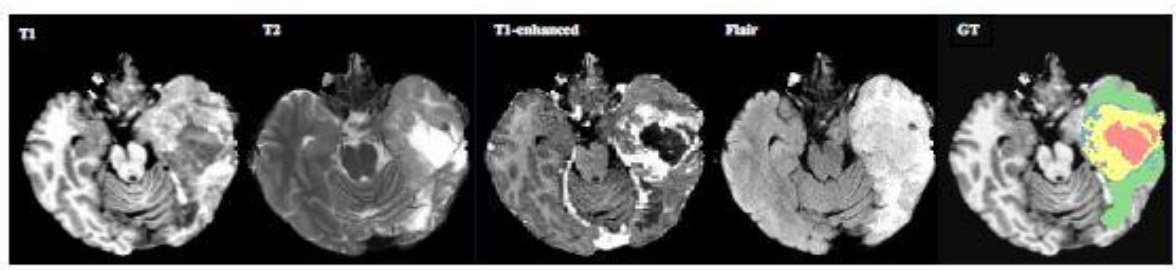


Fig 9 : A patient's brain MRI. The picture displays four modalities of the MRI and the ground truth.

### 3.2 Data Pre-processing

The third dimension in the brain MRI of the BRATS 2013 lack resolution. The segmentation is performed by generating 2D slices from the 3D images. These slices are generated with the four modalities (T1,T1-C,T2,FLAIR) as channels. During training and testing, these patches are generated in a way that the pixel to perform classification on is at the centre. Thus the model will predict a pixel that is centered at the 2D patch. The border pixels of the images are ignored and the inside pixels only are taken into account. The blank slices and patches are filtered out during the generation of 2D slices per 3D image. The slices containing non-tumor pixels are not taken into account.

## 3.3 Proposed Architecture

### 3.3.1 Approach

The main architecture of our model is a two-pathway CNN architecture. These two pathways are known as the local pathway and global pathway respectively. The two pathways processes input patch with smaller receptive field and larger receptive fields respectively. The concept of such architecture is based on the idea that the path with smaller receptive fields i.e. the local path provides detailed visual information of the surrounding region around the pixel and the global path with larger receptive field provides more contextual information i.e. localize the patch roughly.

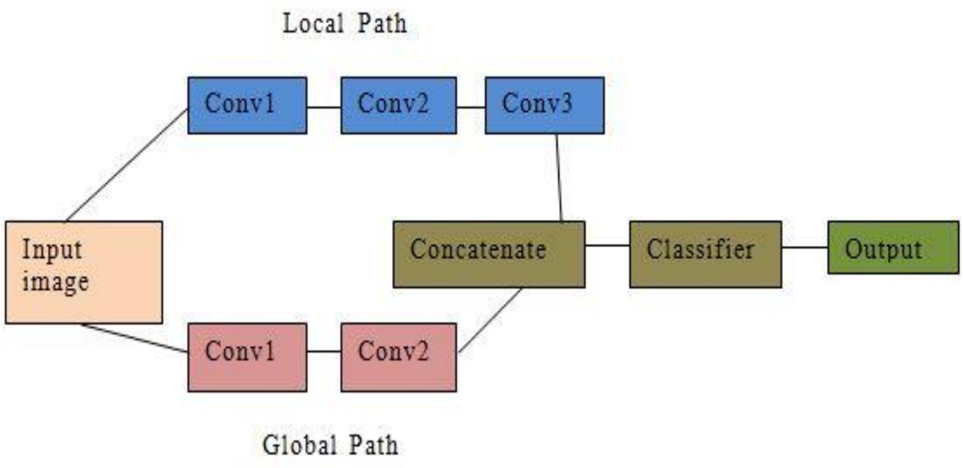


Figure 10: Simple two-path architecture

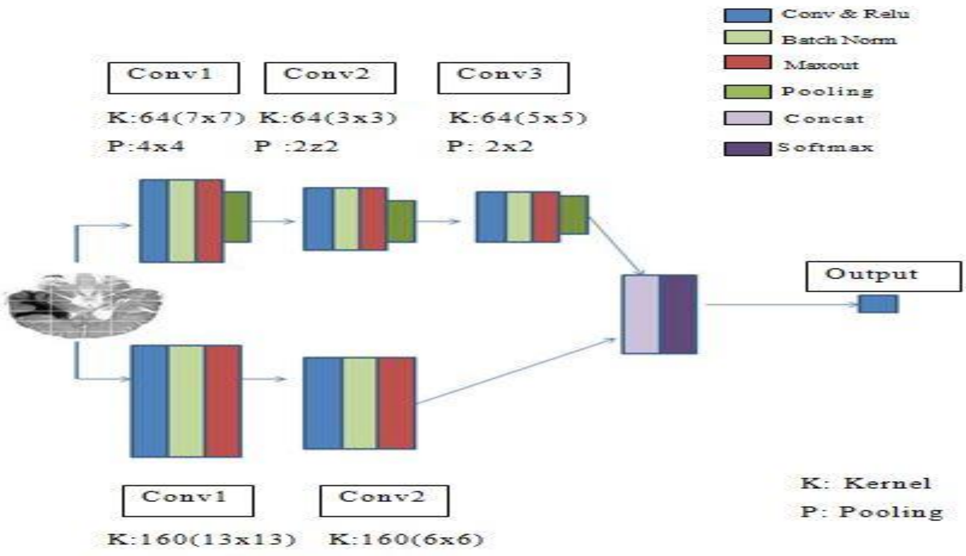


Fig 11: Visual representation of two path CNN

### 3.3.2 Calculation of a single block of convolution layer

Let us consider we have an input of (7x7) size. We apply 5 kernels of size (3x3). There is no padding and the value of stride is 1. After that we perform max-out. And finally we apply a max-pooling of (2x2).

We are now going to calculate the output dimension for this layer.

Input dimension (WxH): 7x7

kernels ($K_{W,}K_H$): 5 (3x3)

Padding (P) : no padding

Stride(S) : 1

Feature map dimension: $(W-K_{W\,+}\,2P)/S\ +1 = (7-3)+1 = 5$. So, output (5x5)

Pool_size = (2x2)

Pooled feature map dimension: (5-2)+1 = 4. So, the output dimension is (4x4).

### 3.3.3 Local Path

The dimension of the input image slice is (65x65x4). Here the value 4 represents the number of channels due to the four modalities of the MRI image. In the local path there are three convolutional layers.

**Conv1:** The first convolutional layer consists of 64 kernels of 7x7 size. This is followed by a batch- normalization and max-out operation. After that max-pooling is performed with pool_size 4x4. Finally activation ReLU function is applied. Here, no padding is used and the value of stride is 1. The dimension of the output feature map for this layer is (56x56x64).

**Conv2:** The input for the second convolutional layer is of size (56x56x64). The same number of kernels are used but of size (3x3). After batch normalization, max-out, pool_size (2x2) is used for max-pooling operation. In this layer also, the same activation function is used. The dimension of the output feature map is (53x53x64).

**Conv3:** Third and final layer of the local path takes input of (53x53x64). The size of kernel is (5x5) and the number of kernels and the rest of the operations are done just like in the previous layers. For max-pooling the pool_size is (2x2). The output dimension is (48x48x64)

### 3.3.4. Global Path

In the global path, there are two convolutional layers.

The two convolutional layers have 160 kernels of (13x13) size and (6x6) size respectively. The input dimension is the same as the local path, (65x65x4). Maxout and batch normalization are done. The dimension for the output feature map is (48x48x160).

### 3.3.5. Merging of Local and Global Path

We notice that the dimensions of the final feature map from local path and that of the global path are the same. So, we can concatenate the feature maps from both paths. The final dimension will be (48x48x224). Here 224 is obtained by adding the number of kernels.

### 3.3.6. Implementation with 2 CNNs

The main architecture is composed of two two-pathway CNNs. The output probability of the first CNN is fed to the input of the second CNN. This is done because if we implement only one two-pathway CNN, it will predict label for each segmentation separately. But to influence the prediction by the model's beliefs about the nearby labels, the output of first CNN is fed as an additional input to the second CNN. This will allow to take the dependencies between adjacent labels to be taken into consideration during prediction. The complete architecture is shown in Figure 11.

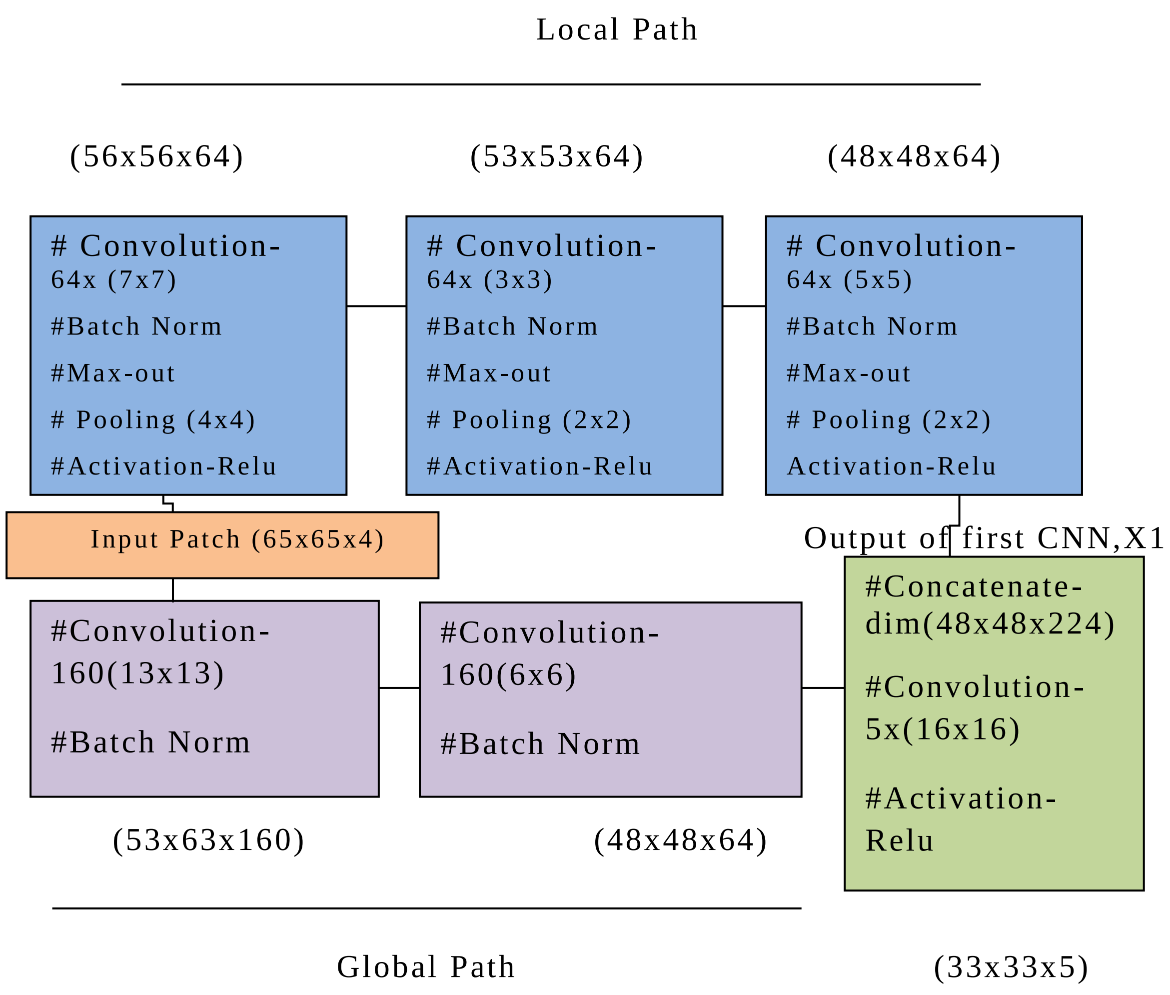


Fig 12(a): First CNN

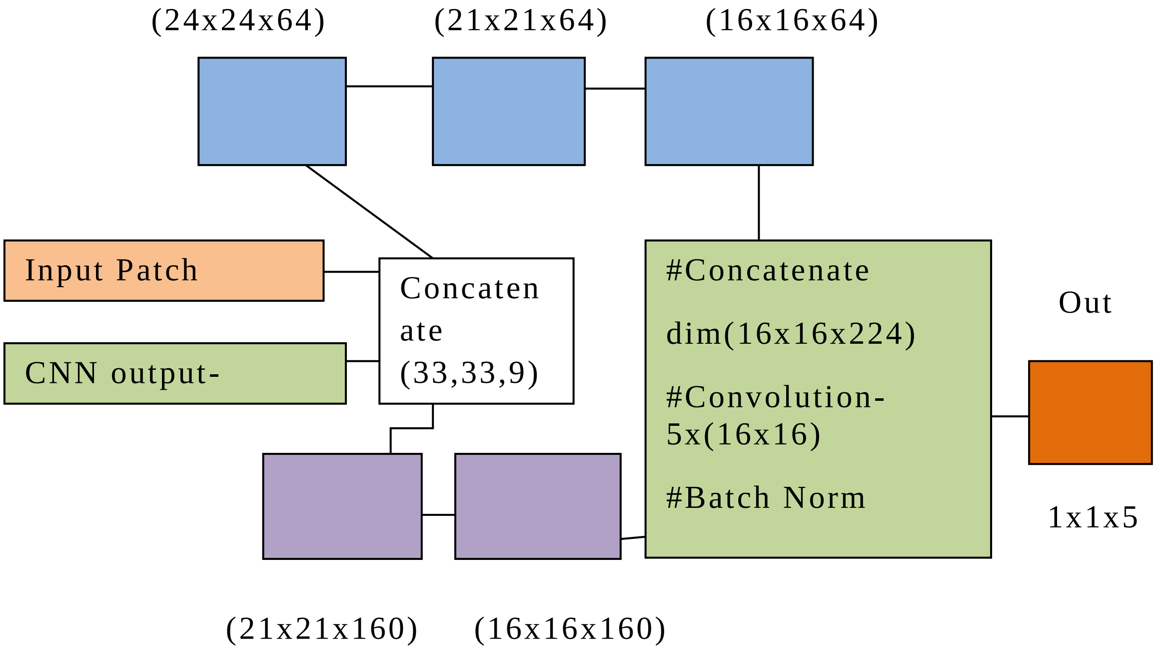


Figure 12 (b): Second CNN

The figure illustrates the model architecture completely. The input size for first CNN is (65x65x4). The number of channels here is 4 because four input modalities are associated as four channels with the input slice. We can see in the local path there are three convolutional layers. First convolutional layer has 64 (7x7) kernels. After that batch-normalization, max-out and finally pooling is done with (4x4) window size followed by activation function ReLU. In the second layer, kernel(3x3) & pooling (2,2) and in the third layer kernel (5x5), pooling (2x2) are used and the other parameters remain the same as the first layer.

In the global path. there are two layers that uses 160 kernels of size (13x13) and (6x6) each respectively.

The outputs of the final local and global path convolutional layers are concatenated which is then in turn concatenated with the input of the second CNN which repeats the same steps again of the two-pathway CNN model. Finally the output of the 2nd CNN applies Softmax activation function.

**A short list summary of our implemented model is:**

**The number of total parameters**: 5,808,754

**The Number of trainable parameters:** 5,804,638

**Non-trainable parameters**: 4,116

# Chapter 4

## Implementation & Result

### 4.1 Environment Details

The experiment is performed on a windows machine with 12 GB RAM. However, the code is executed in Google Colab accessing its free GPU. The implementation is done from scratch in Python, using the Keras library

### 4.2 Training & Testing

For the training purpose, we generate per-pixel wise slices. For an image, filtering out all the zero and non-tumour slices, there are approximately 180-200 slices per image. We train the model for 15 HG images. Hence, the total number of slices i.e. training data is approximately 3000. The initial dimension for each image is (176,216,160). After extracting 2D patches, we generate slices for each pixel located at the centre. As there are two CNNs, the input dimension for the first CNN is (65,65,4) and the output is (33,33,5). This output is concatenated with the input of the second CNN which is (33,33,4). The output of final layer after performing Softmax classification is (1,1,5). The number of epochs for each slice is 5. During training, the accuracy for each slices gradually increases to 90-100%.

We have performed prediction on 5 HG training images. The data are pre-processed in the same way as in the training. And prediction is done for each slice of an image. The performance is described in the Results section.

### 4.3 Results

To measure the performance of the model two metrics are used which are imported from the sklearn.metrics.. These are accuracy_score and f1_score.

f1_score is the weighted average of the precision and recall and its best value is at 1 and worst score is 0. The formula is :

f1= 2 x (precision * recall) / (precision + recall)

On the other hand, accuracy_score computes subset accuracy in multi-label classification. The set of labels predicted for a sample must exactly match the corresponding set of labels in ground truth.

The prediction is performed slice wise for an image and thus the accuracy is calculated for each slice prediction and then the average is calculated which is taken as the performance score for the image. The results for the 5 images are shown below:

| | Test 1 | Test 2 | Test 3 | Test 4 | Test 5 |
|---|---|---|---|---|---|
| f1_score | 0.9933 | 0.9298 | 0.9768 | 0.8828 | 1.00 |

Table 1: Test performance of the proposed architecture.

## 4.4 Result Analysis

### 4.4.1 Baseline model and effects

The baseline architecture of our work is the implementation of the model proposed in "Brain Tumor Segmentation with Deep Neural Networks"[1]. The model we have implemented in this paper is inspired from the above mentioned paper and is modified to create a result with better accuracy.

The baseline model is a two-path CNN architecture. This model processes information in two different paths, local and global path. The local path is intended to process features from an input with smaller receptive fields (7x7) whereas the global path is intended to do the same from the same input with larger receptive fields (13x13).

The local path has two convolutional blocks (Conv1 & Conv2) whereas the global path has only one convolutional block.

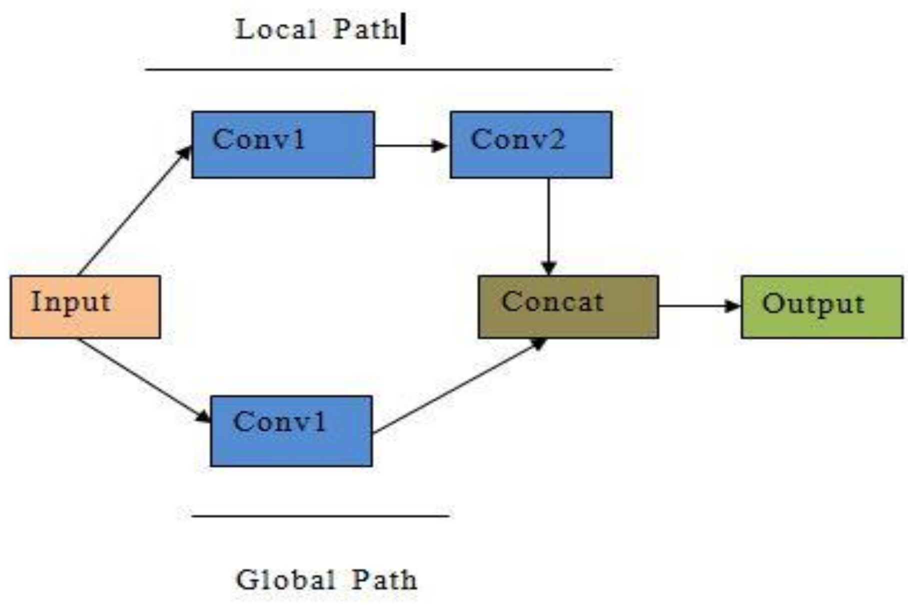


Fig 13: Baseline model architecture inspired from "Brain Tumor Segmentation with Deep Neural Networks"[1].

The main idea behind this architecture is to merge both local and contextual information efficiently. Thus, the local path provides detailed information around the targeted pixel and the global path helps to localize the pixel. These factors contribute in better segmentation results of the brain MRIs.

In the methodology, we can see there are two CNNs. The output of first CNN is given as an additional input to the second CNN. This allows to consider the dependencies between the adjacent labels in the segmentation. That is, the final output of the model is influenced by the model's predictions about the adjacent labels. This also plays an important role in producing better segmentation results.

### 4.4.2 Reproduced model and its effects.

Our main contribution in this model is making the network deeper i.e. adding more layers to it. It is to be noted that a convolutional block is added in both the local and global path respectively.

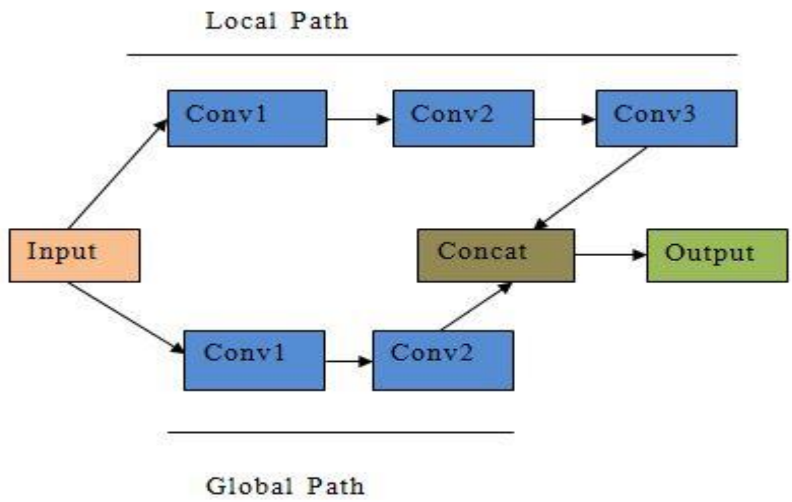


Fig 14: Reproduced model

The convolutional layers in CNN learn the features of an image. The more the number of layers, the more sophisticated and complex features are learnt. But with that, one disadvantage is that it learns the training images too well which cause poor performance of the model when generalizing or predicting on unseen images (overfitting). The idea is to learn the important features as accurately as possible without overfitting. We have successfully reproduced the model by making it deeper and enhance accuracy of the model compared to the baseline model. The additional layers help the model to learn more complex features without overfitting and performs segmentation with better accuracy.

### 4.4.3. Comparison of performance between baseline model and reproduced model

We tested both the models on same images as mentioned in the Result section. We will now have a look at the comparison between these two models in Table 2. Here, the unit of measuring accuracy is f1_score for both the model

| | Test 1 | Test 2 | Test 3 | Test 4 | Test 5 |
|---|---|---|---|---|---|
| Reproduced model | 0.9933 | 0.9298 | 0.9768 | 0.8828 | 1.0000 |
| Baseline model | 0.9661 | 0.9055 | 0.9402 | 0.6201 | 0.9990 |

Table 2: Baseline and reproduced model comparison

We can see the model has shown high accuracy in prediction. The layers in a CNN model are responsible for learning the features of the images. The initial layers learn simple features and the deeper the network gets, the more complex and sophisticated features are learnt by the layers. Thus, the more the number of layers are, the more features are learnt by the model. But one disadvantage of a deeper network is that the model learns the features of the training images too well and hence it cannot generalize well on the data it hasn't seen before. This is known as overfitting.

We have seen the number of parameters in our model is quite large. To prevent the model from overfitting we have to provide large number of training data as well. Though the number of images we use for training are not many, but since we generate 2D slices for each pixel of an image, the number of training data increases. So for 5 images, we are actually training the model on 5x200 i.e. 1000 slices. Moreover, the brain MRI does not vary from one patient to another that much. Hence our model is not overfitting and the accuracy is higher compared to the baseline model.

## Chapter 5

# Conclusion

### 5.1 Result Summary

We have proposed a model in this paper that conforms with the performances of the state-of-the-art methods. First we implemented our baseline model and tested it on 5 training images. The model has two pathways- local and global. In the local path there were 2 convolutional blocks and in global path there was just one convolutional block. The feature maps from both pathways were concatenated and given as an additional input to the second CNN. The output of this second CNN gives the segmentation result. The performance was good because of the two-path architecture and implementation with 2 CNNs.

We reproduced the model by adding one convolutional block in each of the local and global pathways. The addition of the layers in both pathways made the network deeper. These layers helped the model to learn more complex features and produced better segmentation results as shown in Table 2. The model did not overfit because we generated patches per pixel resulting in an increase of training images. The average accuracy for the old model is 0.88618% and that of the reproduced model is 95.654%. Thus we notice a significant enhancement in the accuracy.

### 5.2 Future Work

During implementation of this model there were some constraints. The main constraint was the lack of high performance computation resources. As we mentioned in the Implementation section of our paper, we have used free GPU access provided by Google colab. We plan to extend our work in the future where we will be able to implement our model with enough computation resources in order to speed up our training process. It will result in the training of more images and improve the overall accuracy.

# Bibliography

[1] Mohammad Havaei, Axel Davy, David-Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, Chris Pal, Pierre-Marc-Jodoin, Hugo Larochelle- "Brain Tumor Segmentation with Deep Neural Networks," 2016.

[2] B Fischl- "FreeSurfer," Neuroimage 2012

[3] Patenaude B., Smith SM. Kennedy DN, Jenkinson M.- "A Bayesian model of shape and appearance for subcortical brain segmentation", 2011.

[4] T.F, Cootes, C.J. Taylor, Cooper, D.H, Graham, J.- "Active Shape Models -Their Training and Application," 1995.

[5] T.F. Cootes, G.J. Edwards and C.J. Taylor- "Active Appearance Models," 2001.

[6] Zhang W, Li R, Deng H, Wang L, Lin W, Ji S, Shen D-"Deep convolutional neural networks for multi-modality isointense infant brain image segmentation", NeuroImage 2015

[7] Kleesiek J, Urban G, Hubert A, Schwarz D, Maier-Hein K, Bendszus M, Biller A-"Deep MRI brain extraction: A 3D convolutional neural network for skull stripping" , 2016.

[8] Pim Moeskops, Mitko Veta, Maxime W. Lafarge, Koen A.J Eppenhof, Josien P.W. Pluim - "Adversarial training and dilated convolutions for brain MRI segmentation", 2017.

[9] Konstantinos Kamnitsas, Christian Ledig, Virginia F.J. Newcombe, Joanna P. Simpson, Andrew D. Kane, David K. Menon, Daniel Rueckert, Ben Glocker-"Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation", 2017.

[10] Jose Bernal, Kaisar Kushibar, Mariano Cabezas, Sergi Valverder, Arnau Oliver, Xavier Llado-"Quantitative analysis of patch-based fully

convolutional neural networks for tissue segmentation on brain magnetic resonance imaging", 2018.

[11] Kaisar Kushibar, Sergi Valverde, Sandra Gonzalez-Villa, Jose Bernal, Mariano Cabezas, Arnau Oliver, Xavier Llado-"Automated sub-cortical brain structure segmentation combining spatial and deep convolutional features", 2017.

[12] Pim Moeskops, Max A.Viergever, Adrienne M.Mendrik, Linda S.de Vries, Manon J.N.L Benders, Ivana Isgum-"Automatic segmentation of MR brain images with a convolutional neural network", 2017.

[13] Snehashish Roy, John A.Butman, Daniel S. Reich, Peter A. Calabresi, Dzung L.Pham-"Multiple Sclerosis Lesion Segmentation from Brain MRI via Fully Convolutional Neural Networks", 2018.

[14] Jose Dolz, Ismail Ben Ayed, Jing Yuan, Christian Desrosiers-"Isointense infant brain segmentation with a hyper-dense connected convolutional neural network", 2017.

[15] Pim Moeskops, Jeroen de Bresser, Hugo J. Kuijf, Adrienne M. Mendrik, Geert Jan Biessels, Josien P.W. Pluim, Ivana Isgum-"Evaluation of a deep learning approach for the segmentation of brain tissues and white matter hyperintensities of presumed vascular origin in MRI", 2017.

[16] Amirreza Mahbod, Manish Chowdhury, Orjan Smedby, Chunliang Wang-"Automatic brain segmentation using artificial neural networks with shape context", 2017

[17] Jose Dolz, Christian Desrosiers, Li Wang, Jing Yuan, Dinggang Shen, Ismail Bin Ayed-"Deep CNN ensembles and suggestive annotations for infant brain MRI segmentation",2017.