Detection of Unhealthy Brain Disease using MRI with Artificial Intelligence

Authors

Arslan	**2017-EE-366**
Muniba Kokab	**2017-EE-416**
Noman Nosher	**2017-EE-428**
Ahmed Sana	**2017-EE-444**

A Final year report submitted in partial fulfilment of the requirement for the degree of

Bachelor of Science in Electrical Engineering

Supervisor

Engr. Zain Shabbir (Lecturer)

Department of Electrical, Electronics and Telecommunication Engineering

UNIVERSITY OF ENGINEERING AND TECHNOLOGY, LAHORE

(Faisalabad Campus)

SESSION 2017-21

Detection of Unhealthy Brain Disease using MRI with Artificial Intelligence

A report submitted in partial fulfilment of the requirement for the degree of

B.Sc. Electrical Engineering

Internal Examiner **External Examiner**

Chairman

Department of Electrical, Electronics and Telecommunication Engineering

Department of Electrical, Electronics and Telecommunication Engineering

University of Engineering and Technology Lahore (FSD Campus)

June 2021s

DECLARATION

We solemnly declare that this report is written by us and is not copied from any online or printed material.

Arslan	______________________
Muniba Kokab	______________________
Noman Nosher	______________________
Ahmed Sana	______________________

ACKNOWLEDGEMENTS

First and foremost, we would like to express our gratitude to the University of Engineering and Technology, Lahore (UET) for providing complete project or study materials to make my research easier. As an undergraduate student, the wonderful atmosphere leads to my success. Second, we would like to thank our counsellor, Sir Zain Shabbir, who is also our project supervisor, for his continuous guidance and instruction. Sir Zain Shabbir has shared a lot of his research experience with us, which has inspired us to work on our project. Aside from that, we would like to thank all our colleagues, lecturers, technicians, and Uetians club members who helped make our project a success. It is greatly appreciated that they participated in the project's experimental phase. Finally, and most importantly, we would like to thank our family for all they've done. Throughout our undergraduate studies, they have supported us in academia. We are thankful for their endless moral support. Their inspiration and encouragement have been at the center of my drive to learn and create

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATION

ANN	Artificial Neural Network
CNN	Convolutional Neural Network
FNN	Feedforward Neural Network
GPU	Graphics Processing Unit
NN	Neural Network
MRI	Magnetic Resonance Image
WT	Whole Tumor
TC	Tumor Core
ET	Enhancing Tumor
HGG	High Grade Glioma
LGG	Low Grade Glioma

ABSTRACT

Analysis of the 3d volume of brain MRI of the patient is a time taking task for doctors. Deep Learning algorithms help doctors for investing interesting parts timely and then finalize the results more efficiently. Brain tumor segmentation separates the cancerous part of the brain from the normal brain. Brain tumor segmentation is classified into four parts such as preprocessing, segmentation, optimization, and feature extraction. The importance of MRI-based segmentation expanded in recent years. MRI is the most reliable, safe, and has good resolution. It has no side effects, no radiation, and not harmful to other parts of the body. The tumor is segmented after MRI is processed. In this report, we proposed an automatic segmentation method based on CNN. It includes many layers for feature extraction and brain tumor segmentation. For this, the BraTS 2018 dataset consists of 210 HGG (High-Grade Glioma) images and 75 LGG (Low-Grade Glioma).

CHAPTER 01
INTRODUCTION

1.1 Introduction

In our work, we have not only detected the brain tumor but we also segmented that where is tumor in the Human Brain. The task of detecting the position of the tumor in the body of the patient is the starting point for medical treatment. Brain tumor detection in an early stage can help to reduce the death rate in the medical field. The most common brain tumor is gliomas. It is categorized as HGG (high-grade glioma) and LGG (low-grade glioma). By the use of MRI, we get information on gliomas. For sub-regions and details of a tumor, the MRI has the following sequence such as T1-weighted image, T1-weighted with gadolinium contrast enhancement (t1gd), T2-weighted image, and fluid-attenuated inversion recovery- (FLAIR) weighted image.

1.1.1 Brain Tumor statistics

Brain tumor is a very well-known disease. Latest statistics about this disease are depicted below. It observed on June 08 every year across the globe, its aim to create public awareness on brain tumors and supports brain tumor patients. The health of ministry reported that in Pakistan, the total number of qualified neurologists is about sixty who are practicing in the seven populated cities of the country. According to World Health Organization, more than 1,000,000 people are diagnosed with brain tumor per year around the world. The death ratio due to brain disease increased every year in adults as well as in children. The most percentage of brain tumor has been observed in Australia and North America [1].

1.1.2 Brain Tumor

The brain tumor is an irregular and anomalous growth of cells in the brain and the most mortal types of cancer infections. A brain tumor occurs when abnormal cells form within the brain. A brain tumor MRI is shown in Figure 1.1:

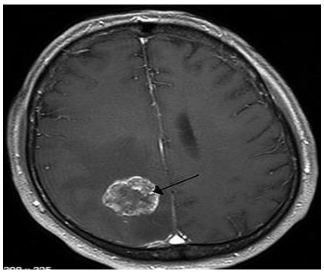


Figure 1.1 Brain tumor

1.1.2.1 Classification of tumor

The brain tumor is classified into primary and secondary; primary originates in the brain and composes of cells of that organ where the tumor locates, and secondary tumor spreads from other parts of the body in the brain. Normally tumors are classified as benign (non-cancerous) or malignant (cancerous), Non-Malignant tumors are Low Grade and Malignant tumors are High Grade [2]. The death ratio due to brain disease increased every year in adults as well as in children. The detection, feature extraction and segmentation of diseased area of human brain from MRI is difficult and time taking. Experts and Experienced radiologists or clinical are required. Brain tumors screen early is crucial. Magnetic resonance imaging (MRI) helps the radiologist to find and diagnosis the tumor [12]. The malignant and benign tumors as shown in the Figure 1.2:

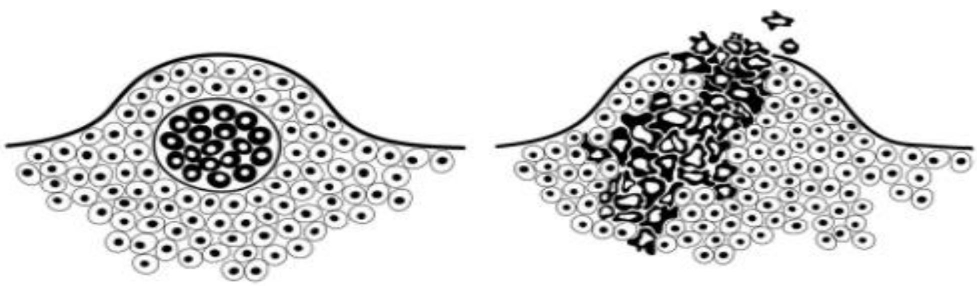


Figure 1.2 Benign and Malignant tumor

1.1.3 Different Brain Imaging Paradigms

There are a number of ways for imaging like X-rays, positron emission tomography (PET), computerized tomography (CT), electroencephalography (EEG), Magnetic resonance imaging (MRI), and functional magnetic resonance imaging (fMRI) [8].

(a) X-rays

It is used electromagnetic energy beams to produce images of internal tissues, bones, and organs. It is better for bones but normal brain does not allow x-rays.

(b) Computerized Tomography (CT)

It used computer processed x-rays to produce tomographic images. It detects tumor, bleeding and brain swelling. Sometimes material(dye) attacked on kidney.

(c) Positron Emission Tomography (PET)

It shows brain activity instead of brain structure. It is not good for diabetic patient because the radioactive material is combined with glucose and increase the sugar level.

(d) Electroencephalography (EEG)

It is the earliest brain scanning technology and shows the brain activity but it is difficult to find the electrical activity of the brain.

(e) Magnetic Resonance Image (MRI)

Magnetic resonance imaging is a radiology technique. It is useful for diagnosis tumors, infections, swelling and bleeding. MRI is the most reliable, safe and has good resolution. The comparison of MRI and CT scan of brain image as shown in Figure 1.3:

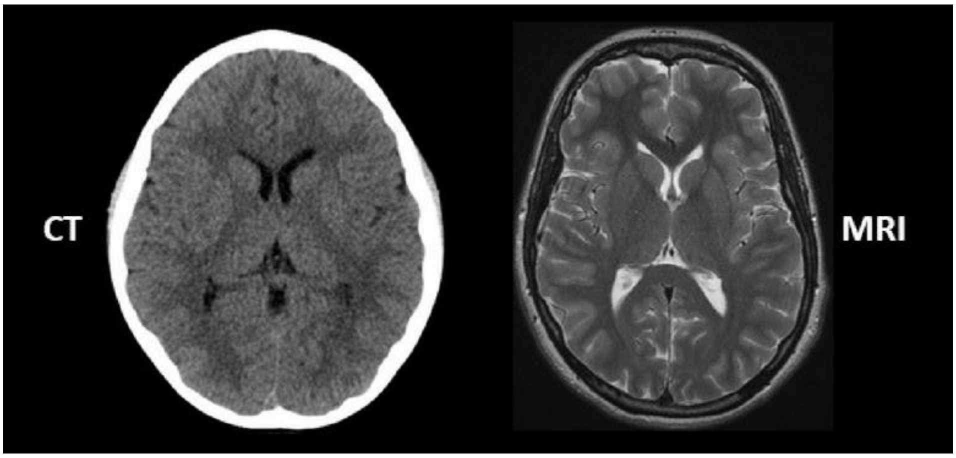


Figure 1.3 Comparison of the MRI and CT scan of brain image

1.1.4 Magnetic Resonance Imaging (MRI) for Brain

Magnetic resonance imaging is most beneficial and preferable technique used in the medical field to produce images of different parts of the human body i.e., brain. It is supported the principles of nuclear magnetic resonance. It is widely used in the medical field. MRI is the most reliable, safe and has good resolution. We prefer MRI because the brain has soft organs and tissues, to get detailed information of human brain. MRI has no side effects even with repeated exposure with magnetic field and has good resolution [9]. The brain MRIs are shown in Figure 1.4

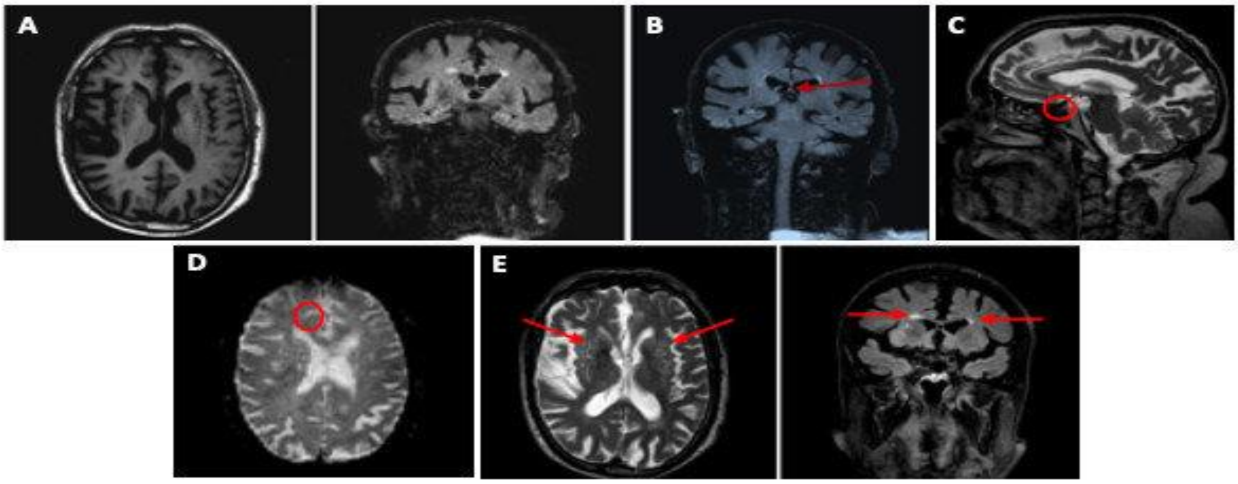


Figure1.4 Brain MRI from different angles

1.1.4.1 MRI Sequences

MRI is a sequence of events that happen in inside the MRI machine gives you images. Different magnetic resonance imaging (MRI) sequence images are used for diagnosis, including T1-weighted MRI, T2-weighted MRI, T1-weighted with gadolinium contrast enhancement (t1gd), and fluid-attenuated inversion recovery- (FLAIR) weighted MRI [10]. MRI images are categorized in following sequence:

(a) T1-weighted image

In MRI, T1-weighted image provides low signal to specified lesion area(edema) of brain. Because, water and collagenous tissues have high protein. The T1-weighted images provide better anatomical details than other images. T1-weighted MRI image shows that fat is bright, water is dark and new blood vessels are also bright. It helps for observing the vascular changes and useful for disruption of blood-brain barrier (BBB) in contrast.

(b) T2-weighted image

In MRI, T2-weighted image provides high signal to water. In T2-weighted image, CSF shows high intensity signal. T2-weighted MRI image shows that fat is dark, water is bright and new blood vessels are also dark. With the help of this sequence, we differentiate the brain lesions from normal brain than T1-weighted image but we cannot distinguish the lesion from CSF because it is also bright.

(c) FLAIR image

It is similar to T2-weighted image but CSF is in dark area. We can improve gray-white differentiation. With the help of Fluid-attenuated inversion recovery (FLAIR), we find brain lesions (edema) that cannot be seen in T2-weighted sequence. It helps to differentiate the edema from hyper-intense area. Fluid image is mostly used in medical field.

All the sequence of MRI are shown in following Figure 1.5 (a), (b), (c):

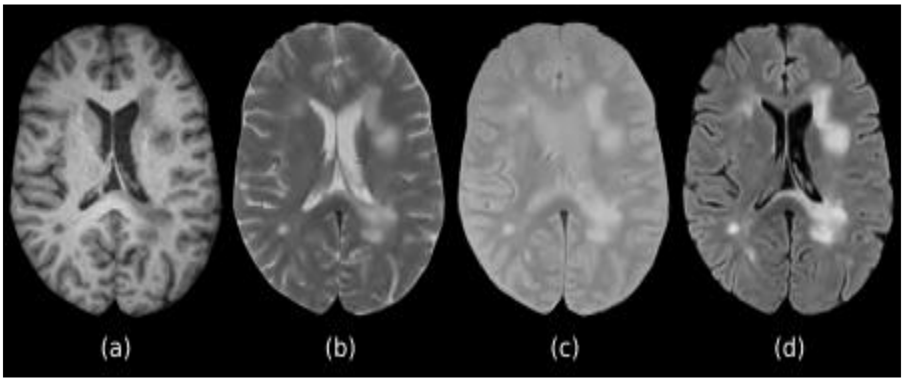


Figure 1.5 a) T1-w b), T2-w c), t1gd, d) Flair

1.1.5.2 Plane views of MRI

Magnetic resonance imaging is comparatively defined in plane. The position of the human is described in the planes like a Cartesian coordinates system. In plane system the MRI is classified into the axial, sagittal, and coronal planes [11]. The axial plane means top view, sagittal means side view, and coronal means front view of the body. With the help of these plane, we evaluate and inspect the disease to some extent. The basic plane view of MRI of the human body is categorized in sagittal plane, coronal plane, and transverse plan and shown in Figure 1.6.

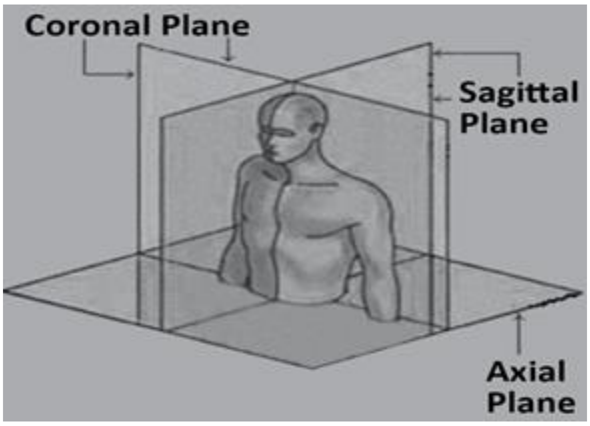


Figure 1.6 Plane views of human body

(a) Sagittal plane

A sagittal plane is sideway plane, from top to down, which separates left from right. The sagittal plane gives the side view of the body. You can view either left or right side of human body.

(b) Coronal plane

A coronal plane gives the font view of the body plane, from front to back, which separates the anterior from the posterior.

(c) Transverse plane

A transverse plane, is an x-y-z plane, parallel to the ground, which separates the superior from the inferior, or put another way, the head from the feet.

The MRI head scans in axial, coronal, and sagittal planes shown in the Figure 1.7.

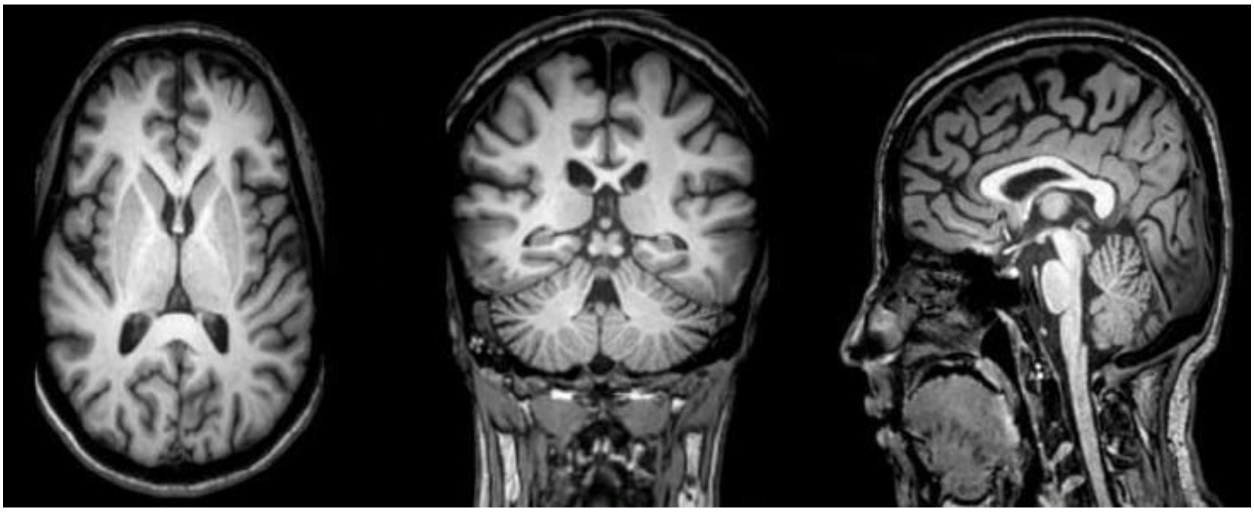


Figure1.7 Brain MRI different planes. Left to right: transverse, sagittal and coronal

1.2 Literature Review

The study of brain tumors has attracted the attention of many researchers around the world. Hundreds of researches are being published yearly that discuss issues related to the brain tumor and the different methods for its early detection. Some of these researches rely on the use of image processing techniques like segmentation in their proposed works. Others use artificial intelligence techniques to perform such tasks. For a complete understanding of brain tumor segmentation, classification, and feature extraction, we study no of the research paper and also studied no of the technique to achieve our goal.

A Hybrid Feature Extraction Method helps physicians and radiologists to identify the brain tumor. This method helps to extract the features from images. then, computing the covariance matrix of these features to project them into a new significant set of features using principal component analysis (PCA). This approach has three steps image preprocessing, feature extraction, and tumor classification. RELM classifier used for this approach [6].

The class imbalance has emerged as a complex problem to overcome this problem the model cascade (MC) strategy, a popular scheme is used. This method leads to undesired system complexity and also ignores the correlation among the models. To solve these problems, we can use One-pass Multi-task Network (OM-Net). It deals with only one deep model rather than separately [7].

Some approaches used the pre-trained model to extract features from medical images but their results are different from natural images. To overcome this problem a new method of multilevel features extraction and concatenation used for the diagnosis of brain tumors. This approach has two pre-trained models i.e., inception-v3 and DensNet201 to extract features then SoftMax classifier classified brain tumor [3].

Brain tumor segmentation using multi-cascaded convolution neural network (MCCNN) and conditional random fields (CRFs) has two steps. First, we design a multi-cascaded network that use inputs to take label into account and make use of multiscale features for segmentation. Second, we apply conditional random fields to consider the spatial contextual information and eliminate the spurious output for fine segmentation. In this we take images patches from different angles for final segmentation [4].

Sparse representation-based radiomics method used for the diagnosis of brain tumor. This method has four steps such as image segmentation, feature extraction, feature selection, and multi feature collaborative classification. In this method, first, we developed a sparse representation-based feature extraction method and dictionary learning to find the damage area of brain or brain lesion and extracted the feature. For this process K-singular value decomposition (KSVD) learning technique is sued [5].

1.3 BraTS dataset

We use BraTS 2018 data which consists of 210 HGG (High Grade Glioma) images and 75 LGG (Low Grade Glioma) along with survival dataset for 163 patients. We use only HGG images 180 for training and 30 for testing. MRI images are categorized in following sequence i.e., T1-weighted image, T1-weighted with gadolinium contrast enhancement (t1gd), T2-weighted image, and fluid-attenuated inversion recovery- (FLAIR) weighted image. Each image is a 3D image with size (240,240,155).

Figure1.8. BraTs dataset logo

CHAPTER 02
ARTIFICIAL NEURAL NETWORKS

2.1 Introduction

Brain of a human being is the most significant body part which is responsible for taking every day routine work decisions. A brain is a soft tissue in our body which is made up of a very complicated structure in which there are countless nerves connected to each other and working together to perform tasks. Even most powerful computers that are manufactured and are used to perform billions of calculations cannot fully compete with our brain. A Human brain which is so smaller in size but still it is so strong that even with small effort in parallel manner it can do those tasks that computer is unable to do so. The anatomy of the brain shows that it is a cluster of many neuron layers, all the layers cooperate with each other in order to achieve parallel processing tasks. [13] It must be taken into account that for information data processing parallel processing has shown to be faster and precise. Our human brain is very good when it comes to images, it can easily and in no time memorize and recognize human and other things. One neuron sends its information to other neurons as a brain structure is made up of billions of neurons so information passes in whole network very efficiently.

2.2 A Biological Neuron Structure

The main computational element of the human brain is a **Biological Neuron**. The chief parts of a biological neuron are the its cell body (Soma), Synapses weights, Dendrites and axon [14]. There are billions of biological cells and all the neurons is interlinked with all neighbor neurons in our brain nervous system. The biological neuron shown below in Figure 2.1 consists of different parts that support it to complete its purpose.

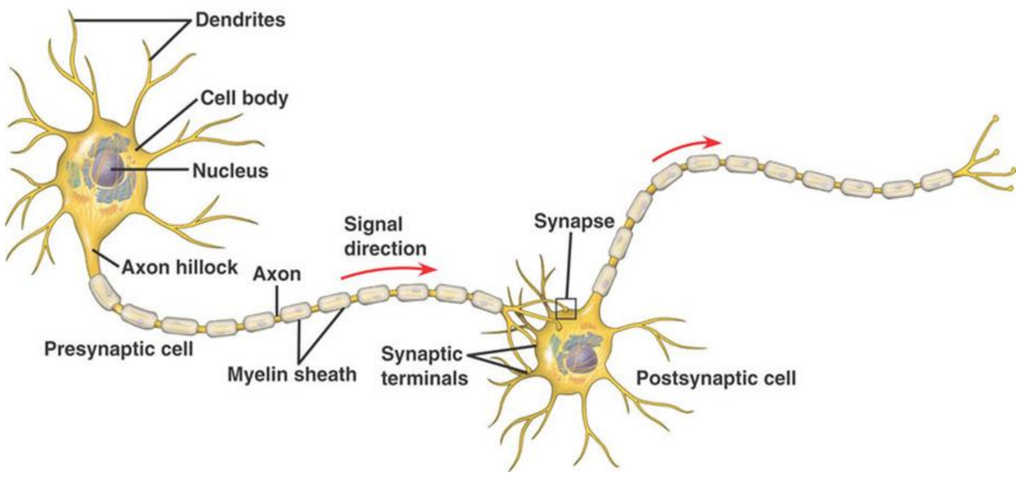


Figure 2.1 Biological neuron structure

According to studies our nervous system is made up of 86 billion neurons and all these neurons are connected with almost 10^{14} - 10^{15} **synapses in** a nervous system. Each neuron receives input signals from its **dendrites** and produces output signals along its **axon** [15] The purpose of axon is to transmit the information to the next neuron. The axon are output of neurons which ultimately branches out. Axons are connected with to the dendrites of other neurons with the help of synapses. In the computational model of a neuron, the signals that travel along the axons (e.g., x0x0) interact multiplicatively (e.g. w0x0w0x0) with the dendrites of the other neuron. This multiplication at synapse is based on the synaptic strength (e.g. w0w0) [16]. One of the important things in the human nervous system is synaptic strengths. The Synaptic strengths (the weights wo) can control the direction or a neuron positive weight or its negative weight. In the elementary model, the dendrites carry the signal to the cell body where they all get added. So a neuron check its final sum if the value of this sum is more than the selected threshold value then it will send a fire or spike on axons and if the value is less than threshold then obviously the fire will not be send

2.3 Artificial Neural Network

The human brain has strong decision-making features that have helped it to solve very complex problems such as mathematical problems, object and motion detection and recognition, and many other complicated problems. Scientists had been trying to mimic the human brain function and structure into the computer. Through Taking inspiration from the biological neural networks in the brain to process information the scientist has created a computational model Artificial Neural Network ANN. The node accepts input from other units or some other external node. All inputs have some weights attached with them; these **weights (w)** are being assigned to the inputs on the base of importance compared to other inputs. There is another input **1** in addition to inputs which is **b** bias as shown in the figure 2.2 .

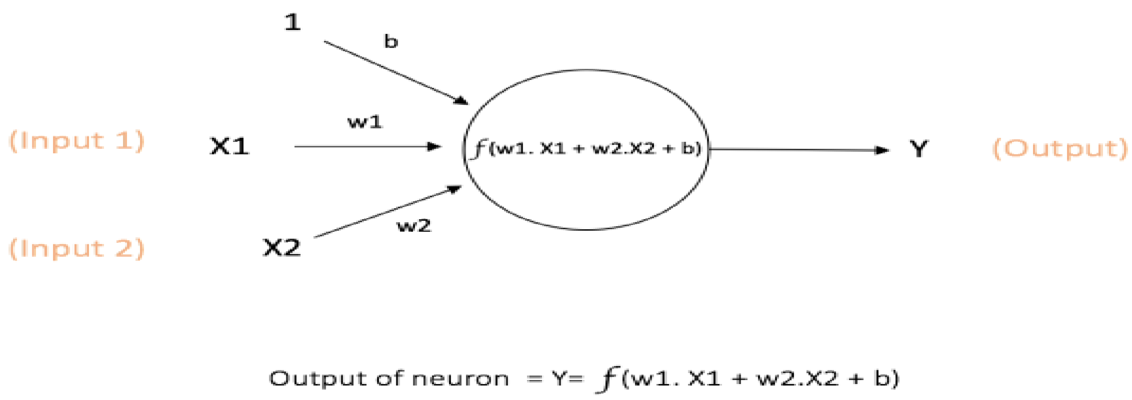


Figure 2.2 A single neuron structure

Bias has the main function of providing a trainable constant value to every node in addition to normal inputs to the nodes. The node applies function **f** to the input weighted sum. The function f is called the **Activation Function**. The most modest and first artificial neural network is the feed-forward neural network. They consist of three main parts

(1) input layer

(2) middle or hidden layers

(3) output layer.

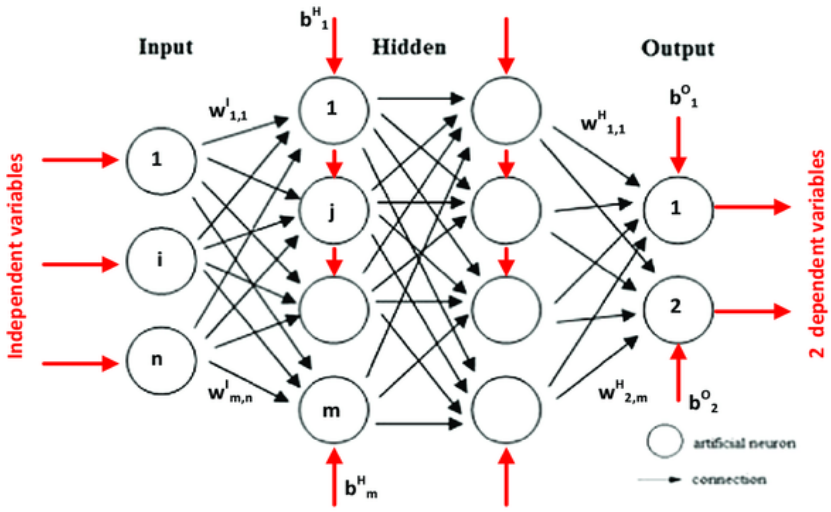


Figure 2.3 Structure of Neural Network

2.4 Input Layer

The input layer which consists of input neuron takes information from an external source and simply transfer it to network Generally, the input layer which is also referred as a **non-processing layer** the reason for calling them non-processing layer is because no computation is performed and the information is just transferred to the network without any change. It is simply a vector that handles the inputs of the network preparing to send them to the hidden layers. As it can be seen in Figure 2.3 first level is the input layer from where the neural network is receiving information

2.5 Hidden Layers

After the input layer, there are hidden layers in the neural network. The hidden layer has no connection with the external source hence called hidden. The hidden layers are the core of an artificial neural network and are composed of many hidden nodes. The hidden nodes play a chief role in neural network structure as they hold most of the processing operations of the neural network. They perform the main mathematical computation and then information is transferred to the outside layer. They all process the different inputs and provide the different outputs. Each neuron in the hidden layers is responsible to receive and send information from the other neurons. It must also be noted that a feedforward can have a single input layer and single output layer but it may have at a time no or maybe many hidden layers

2.6 Output Layer

In the end, all output nodes are called Output Layer. The output layer is the last layer of the neural network that produces the output of the whole network based on the processed information. It plays a very significant role during network training as it is responsible for checking and comparing processed data with the desired outputs. In the output, layer error is being calculated through different ways and is sent back to previous nodes for the process of learning. The weight values are being found and adjusted through the mean square error MSE approach as a minimization approach. A mean square error is calculated by taking the difference between expected results and actual results. A network has a different model and there is a different learning algorithm as well. But the basic model of all networks remains the same e.g., one input and one output. It must be noted that complex models might have many layers, inputs, and outputs.

2.7 Activation Functions

The node applies a function f on input weighted sum as shown in Figure 2.2. This function is called the Activation or Transfer Function. AS the real-world problems are non-linear so activation function is applied to introduce non-linearity into the output of a neuron. An activation function can be on-off e.g., either generate output or cancel it. There are several types of activation functions

2.7.1 Softmax

The softmax function is typically used in the multilayer neural networks output layer. The softmax function is basically used to do the job of classification. A Softmax function computes each class probability. AS we already know about probabilities, the sum of all softmax results in an output layer equals 1. Instead of making a strict choice between classes, using softmax function a input can be estimate that to which class it is more likely belonging. For M classes

$$softmax(x_k) = \frac{e^{x_k}}{\sum_{i=1}^{M} e^{x_i}}$$ (Equation 2.1)

2.7.2 Sigmoid

The sigmoid function taker real-valued numbers and squeezes them between 0 or 1. Basically large positive numbers become 1 while large negative numbers become 0. In Sigmoid Function the training gets slower as the values move away from 0 and saturate[17]. The Sigmoid Function mathematical form and can be seen in below Figure

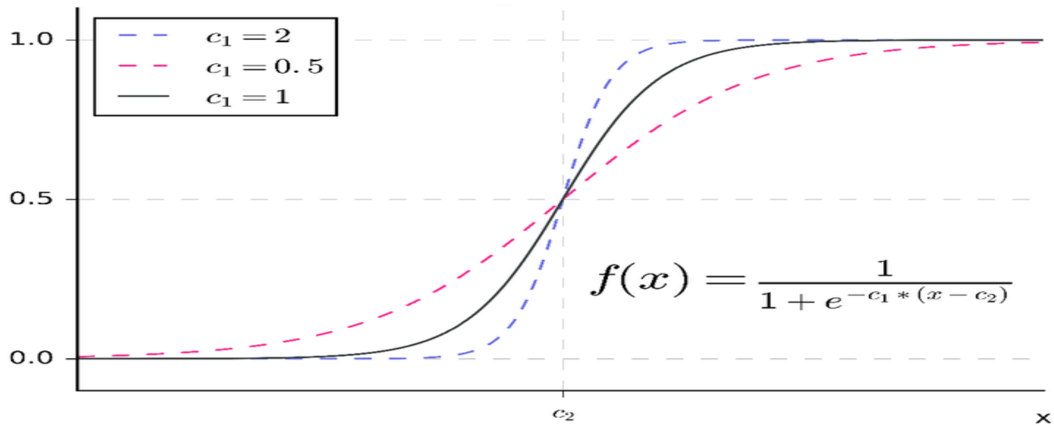


Figure 2.4 Basic sigmoid function

$$f(x) = \frac{1}{(1 + \exp(-x))}$$ (Equation 2.2)

Sigmoid use to be used frequently as it has a nice interpretation of neuron firing rate e.g., at zero frequency no firing and at maximum assumed frequency (1) fully saturated firing. Nowadays in practice, it is rarely used.

2.7.3 Tanh

The Tanh function takes the real-valued number and squeezes it between [-1,1]. Tanh function output is zero-centered, unlike the sigmoid function. Because of this reason tanh is preferred over sigmoid function. Tanh Function has mathematical form and can be seen in Figure 2.5

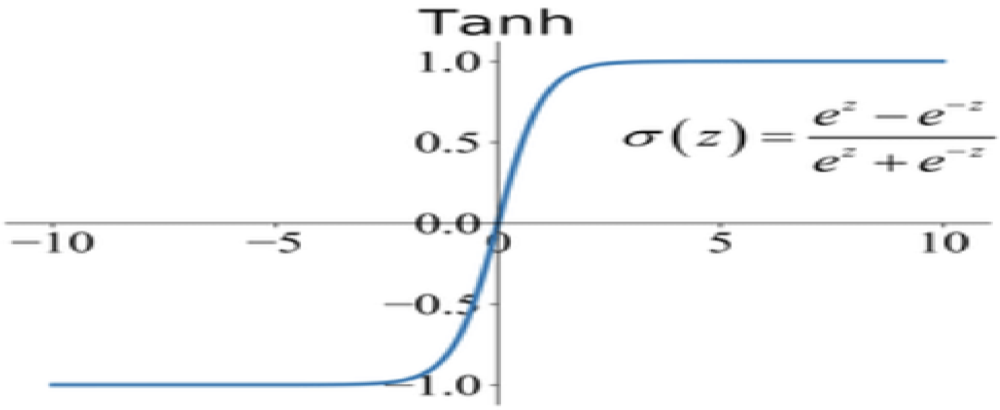


Figure 2.5 Basic Tanh Function

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$ (Equation 2.3)

2.7.4 ReLU

ReLU stands for Rectified Linear Unit. It has become very famous past few years. ReLU is just a Threshold at zero as it can be seen from its mathematical form **f(x)=max (0, x)** and can be seen in Figure 2.6

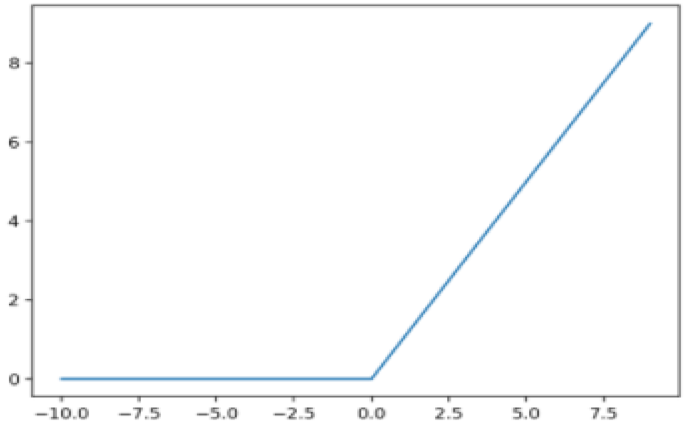


Figure 2.6 Basic ReLU Function

Rectified Linear Function is zero for x<0 and it is linear with slope one for x>0[18]. Furthermore, ReLU has an advantage as the function does not saturate, the gradients do not vanish for very large values[19]

2.7.5 Leaky ReLU

Leaky ReLU is the solution to the dying ReLU problem. Instead of being zero for x<0 in ReLU. Leaky ReLU has a small negative slope of 0.01 for x<0.Leaky ReLU computes **f(x)=1(x<0) (ax)+1(x>=0) (x)** where a is a small constant. Leaky ReLU can be seen in Figure 2.6 below

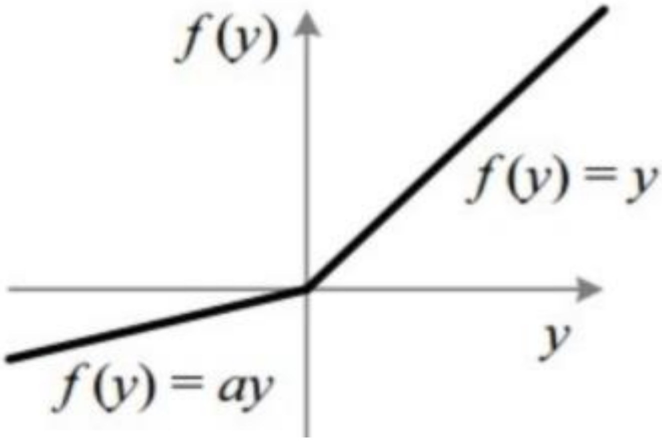


Figure 2.7 Basic ReLU Function

It must be noted that if α is a trainable value, the function is called parametric rectified linear unit PReLU

2.7.6 Maxout

Maxout neuron is the generalization of ReLU and Leaky ReLU. It must be noted that ReLU and Leaky ReLU are a special case of this form as fr ReLU we have w1, b1 =0. Maxout neuron function calculates **max** $(w_1^T x + b_1,\ w_2^T x + b_2)$. Thus using Maxout has all advantages of the ReLU unit .e.g., no saturation and it does not have to face the drawback of ReLU .e.g., dying ReLU. But using ReLU leads to a high total number of parameters as for single neuron parameters doubles.

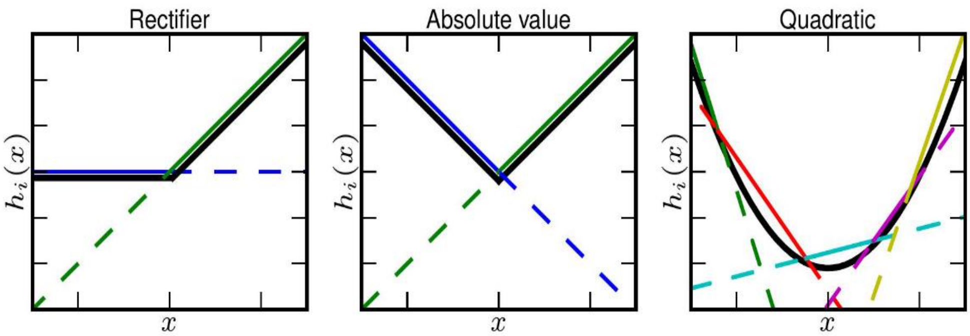


Figure 2.8 Basic Maxout Functi

CHAPTER 03
CONVOLUTIONAL NEURAL NETWORKS

3.1 Brief History of Convolutional Neural Networks

Convolutional neural networks are also termed ConvNets or CNN.CNN was first presented in the 1980s by post-doc computer science researcher Yann LeCun. LeCun had constructed on a Japanese scientist Kunihiko Fukushima work. Kunihiko Fukushima had designed a very elementary image recognition neural network neocognitron a few years back. The early version of CNN's called LeNet named after LeCun, LeNet could recognize handwritten digits. As we know Artificial neural networks got their motivation from Human brain neural networks. Similarly, convolutional neural networks got motivation from animal visual systems Though CNNs have been used in the literature for a time , they were not used broadly until the success of because of two main reasons. The first reason was the lack of data to train the network [20]. The second was that their training time was so time taking because of the lack of parallelization process. Nowadays with the advent of technology numerous photographs are being uploaded every day on the Internet, So the first problem resolved and researchers have overcome the first problem. The second problem was also solved by the advance in the *Graphics Processing Unit* (GPU) Technology.

According to studies, an average Human brain has **8.6 ∗ 1010** number of neuron cells on average and it is being powered by from being immensely parallel. Being such parallelizable, the GPUS architecture is more fitting to the human brain than that of CPUs architecture. AlexNet presented that possibly it is time to go back to deep learning in 2012, the subdivision of Artificial Intelligence which uses multi-layered artificial neural networks.

The accessibility to large data sets, i.e., the ImageNet dataset which has masses of labeled images, and massive compute resources that are available have made it possible for researchers to generate composite Convolutional neural networks that could do computer vision tasks that were impossible to do in past.

3.2 Convolutional Neural Network

Convolutional Neural Network is a subdivision of deep learning which is designed specifically to process pixel data. Convolutional Neural Network is a class of deep feedforward artificial neural networks CNN algorithm is similar to a multilayer perceptron that has been designed for the reduction of processing necessities [21]. The core strengths of CNNs are to provide an efficient dense network that proficiently does the task of prediction or identification etc.

Huge datasets are applied to CNNs, it is even well-thought-out that the larger the data will be, the larger the accuracy will be, and if large data is not available then other techniques such as transfer learning can also be applied to enlarge the data. The main strength of this architecture is that it is able to detect distinct features from images all by itself, without the need for any actual human involvement.

3.3 Convolutional Neural Network Hyperparameters

There are some factors that affect efficiency and must take into consideration

3.3.1 Filters

These filters or kernel or 'feature detector is used to detect the significant exclusive features from the input images and then these features are multiplied with random weights specified These are mainly trained on labeled images and with help of training their efficiency to improve. It must be noted that filters acts lie feature detectors from the same input image.

The different values of the filter matrix will produce different Feature Maps for the same input original image We can create many feature maps to obtain the first convolutional layer of CNN [25]. In Figure 3.1 below are few examples that result from show image after applying different types of filters (Kernels) on the same image

By capable enough to understand and learning various values of filters, CNNs is found to be much more efficient for human images and human calculated filters might not be able to detect.

Frequently, we see these filters in a convolutional layer learn to detect intellectual ideas, like the border of a person facial expression or the person other features. Layers of convolutions in a stack on top of each other, we can get more comprehensive and much more deep knowledge about CNN

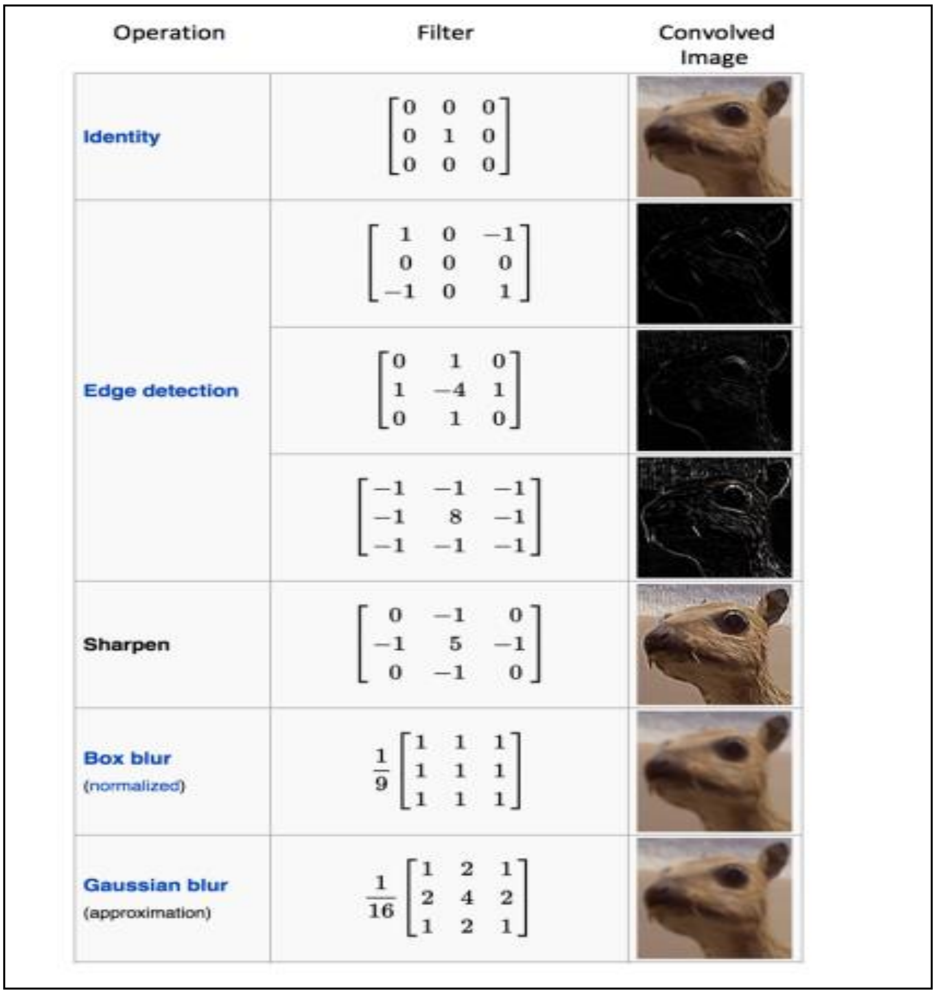


Figure 3.1 image after applying different types of filters

3.3.2 Kernel Size

The parameter that first affects the effectiveness and performance of the working of a CNN model is Kernel size. The kernel size of CNN is usually kept at 3x3 or 5x5. It must be noted for dimensionality reduction that aims to reduce the number of channels 1x1 kernel size is used. If we Increase the kernel size then this will result in an increase in the total number of parameters. We can calculate kernel size by the formula

$$Kernel\ size = n\_inputs * n\_outputs \quad \text{(Equation 3.1)}$$

Considering where c_out is the bias size, the kernel size of the convolutional layer can be found out through

$$k\_w * k\_h * c\_in * c\_out \quad \text{(Equation 3.2)}$$

3.3.3 Stride

The parameter that first affects the efficiency of the working of a CNN model is stride. Stride is the number of pixels through which the filter matrix shift over the input image matrix[22]. It can be said it is the distance to move a filter over the image and it moves faster with larger values. There can be different values for Stride. The most common one for stride is 1[26]. So, when the stride will be 1 then the filters will move 1 pixel at a time the purpose of adding a stride in the model is to decreases the overall size of the feature map and thus reducing the amount of information. When the stride value is 2 this means moving the filters to 2 pixels at a time. Below is the image with a stride value is 2.

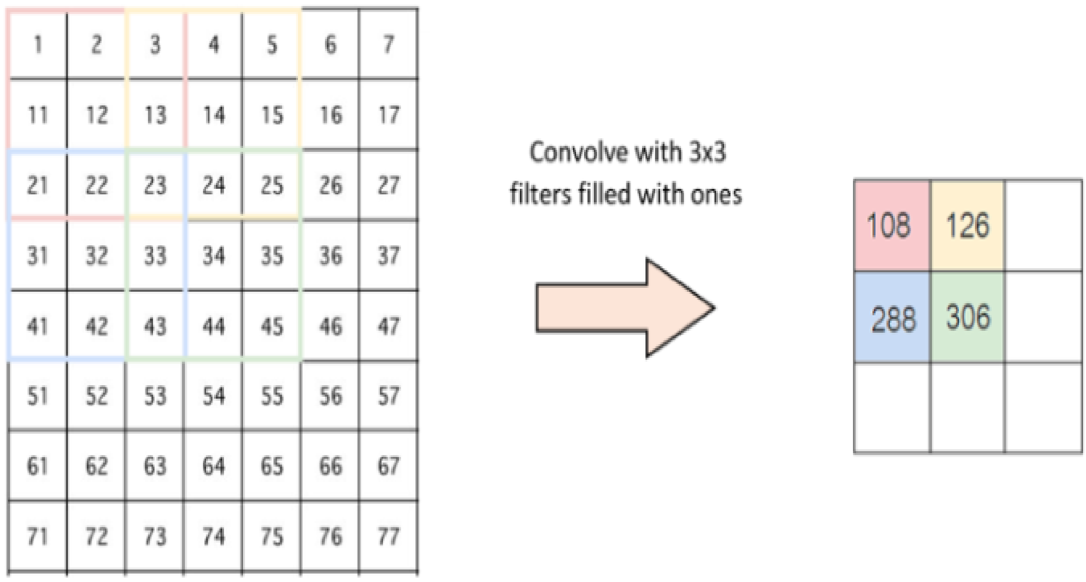


Figure 3.2 Stride of 2 pixels

3.3.4 Padding

Padding is needed when from time to time filter does not fit flawlessly into the input image. There are two types of padding.

(1) Zero-Padding
(2) Valid-Padding

3.3.4.1 Zero-Padding

Zero-padding means to pad zeros around the border of the input matrix through it we will be able to apply the filter to bordering elements of our input image matrix. The size of the feature maps is control through zero paddings. If we are Adding zero-padding to the input image then it is called ***wide convolution,*** and if we do not use zero-padding then it would call ***narrow convolution***.

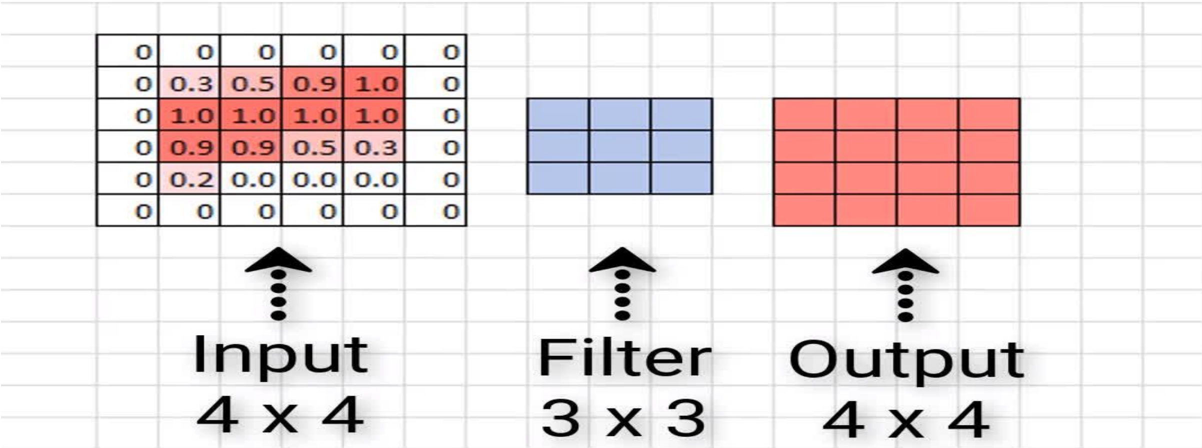


Figure 3.3 Zero Padding

3.3.4.2 Valid-Padding

Valid Padding means that if the filter does not fit then we drop the part of the image to make the filter fit. So, valid padding means that will keep only the valid part of the input image. In simple words, 'valid' padding means there is no padding at all. Here the convolutional layer output size shrinks that depends upon the input size and kernel size[27].

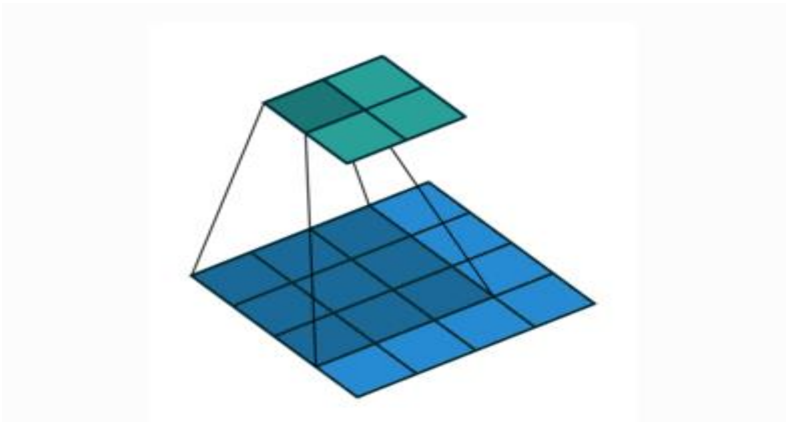


Figure 3.4 Zero Padding.

3.3.4.2 Special Cases of Zero-Padding

In reality, two specific cases of zero-padding are used quite broadly because of their respective properties. These two special cases are as follow

(1) Half padding/ same padding
(2) Full padding

3.3.4.2.1 Half Padding

In this padding type, the output size will be equal to the input size. It is sometimes can be a desirable property. It can be seen in figure

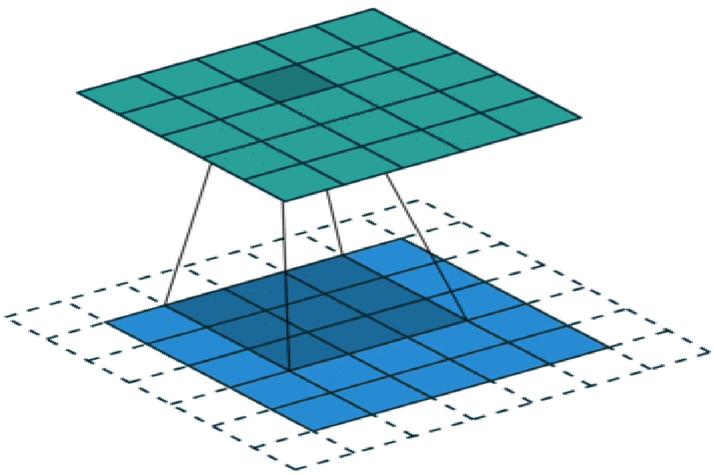


Figure 3.5 Half Padding

3.3.4.2.2 Full Padding

Sometimes the opposite is required i.e., the output size is less with reference to the input size. While convolving a kernel normally decreases the output size with regard to the input size. This can be accomplished with the help of proper zero paddings[28].

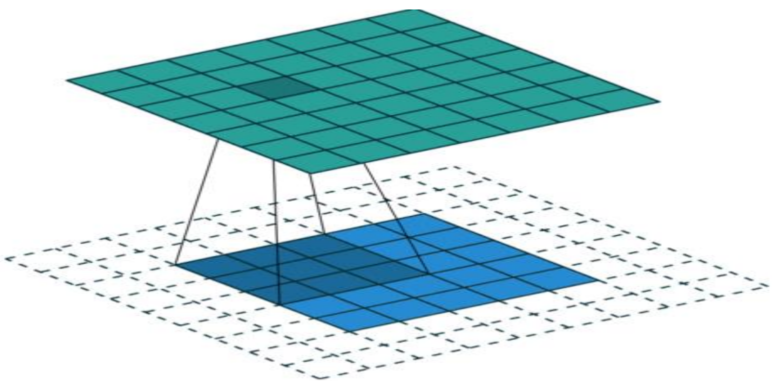


Figure 3.6 Full Padding

3.4 Layers of Convolutional Neural Network

There are some common characteristics between CNN and neural networks, for instance, optimization methods, loss functions, etc. But there exist some non-general characteristics of CNNs as well. The architecture of CNN consists of three types of the layer which are as follow:

(1) Convolution Layer
(2) Pooling Layer
(3) ReLU
(4) Fully connected Layer

3.4.1 Convolution Layers

The convolutional layer is the pillar of any Convolutional Neural Network model. ConvNets took their name because of the convolution operator The chief resolution of Convolution in the first layer to extract features from an input image In this layer pixel by pixel skimming takes place of the input images and creates a convolve feature. Convolution basically conserves the connection between pixels by learning features from the image using input data small squares box. It is such a mathematical operation that has two inputs such as a filter or kernel [23] and an image matrix. Convolution preserves the spatial relationship between pixels by learning image features using small squares of input data. To understand in a better way let's take an input image of dimension 5*5*1. Here the first 5 represents Height, the second 5 represents breath and the last 1 represents Channel. There is three-channel i.e., Red, green, and blue (RGB). The component involved responsible for the implementation of the convolution operation in the Convolutional Layer is called the Kernel/Filter[28], we have selected dimension for this K to be a 3x3x1 matrix, it is being represented in the color yellow in the figure 3.7.

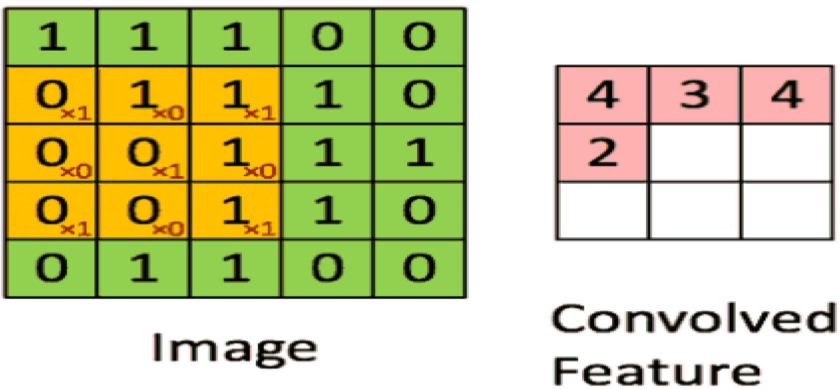


Figure 3.7 Convolution of 5x5x1 input image with a 3x3x1 filter

3.4.2 Pooling Layers

Alike to the Convolutional Layer, the Pooling layer is also accountable for reducing the spatial size of the Convolved Feature. The reason for dimensionality reduction is to decrease the computational power required for data processing. Moreover, it is beneficial for effective training of the model to extract foremost features which are positional invariant and rotational. When the images are too large the pooling layers in CNN architecture would reduce the number of parameters. Spatial pooling is also known as subsampling or down-sampling which reduces the dimensionality of each map but holds imperative information [29]. Spatial pooling can further be of different types mentioned below:

(1) Max Pooling

(2) Average Pooling

(3) Sum Pooling

3.4.2.1 Max Pooling

Max pooling as the name depicts takes the largest value from the rectified feature map. Max Pooling yields the maximum value from the part of the image under the Kernel. Max Pooling is also very useful as it can be used as Noise Suppressant. Its throw-outs the noisy activations all in all. Max pooling also performs de-noising along with the task of dimensionality reduction.

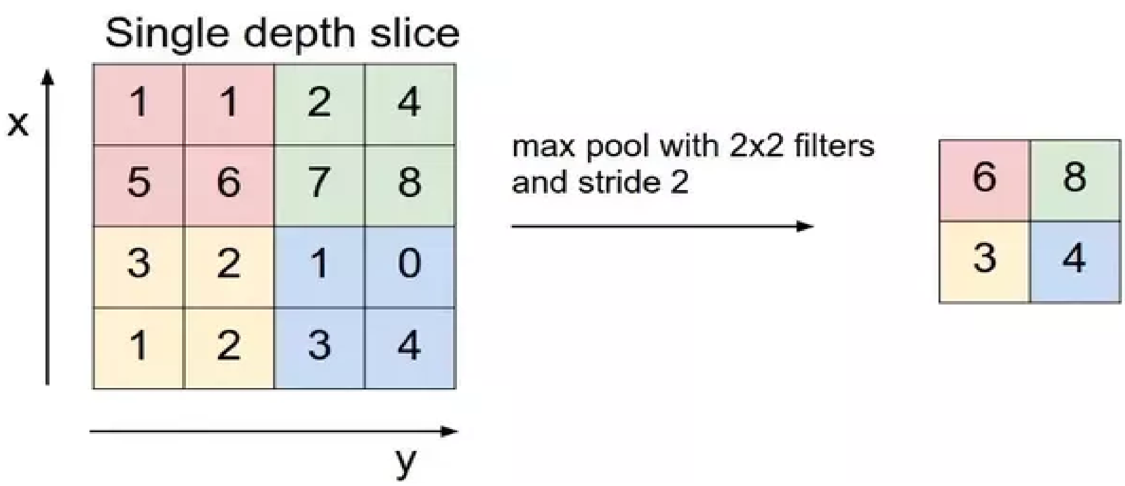


Figure 3.8 Max Pooling

3.4.2.2 Average Pooling

In Average Pooling yields the average of all the values from the slice of the image that is being covered by the filter. It must be noted that Average Pooling merely accomplishes dimensionality reduction as a noise suppressant. Thus, it can be said that Max Pooling performs a lot more in a better way than Average Pooling

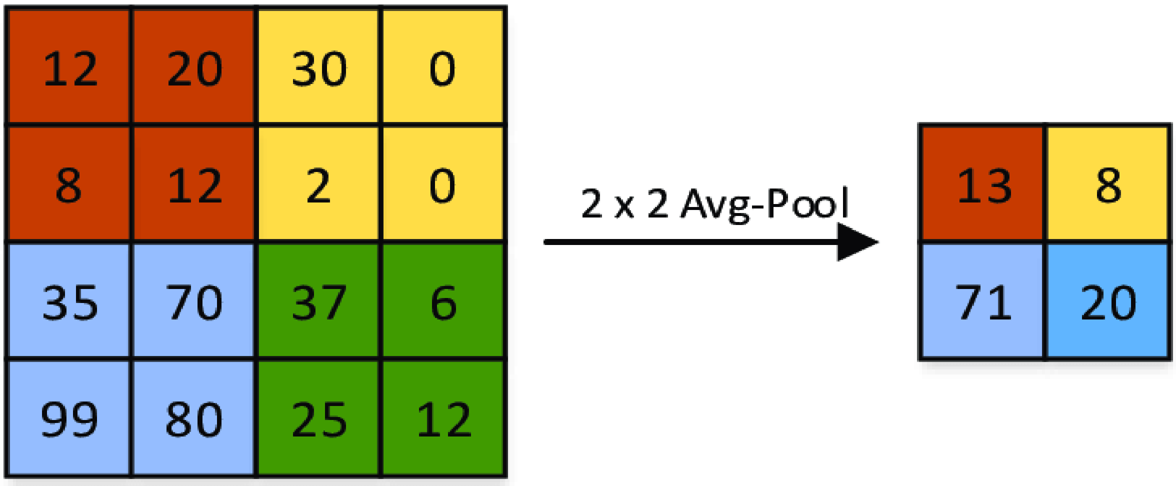


Figure 3.9 Average Pooling

3.4.2.3 Sum Pooling

As the name depicts sum pooling can be done by taking Sum of all elements in the convolved feature. The image below shows sum pooling to get a better understanding

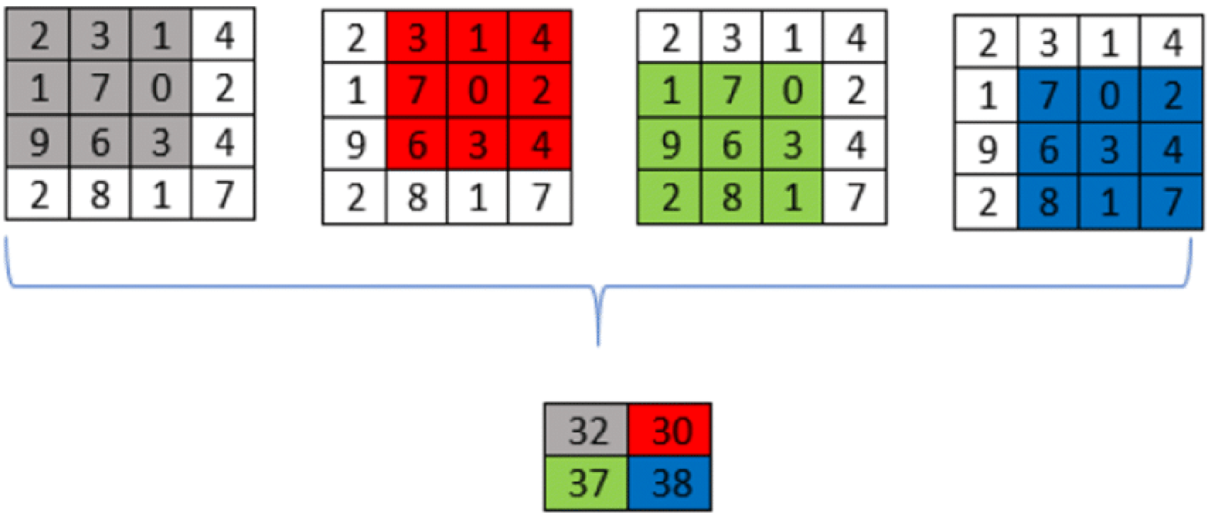


Figure 3.10 Sum Pooling

3.4.3 ReLU

ReLU stands for Rectified Linear Unit. It is a non-linear function. The formula for ReLU is

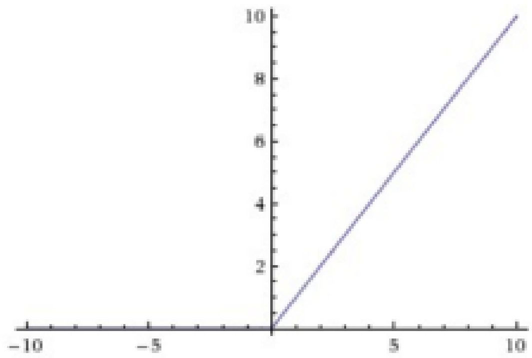


Figure 3.11 ReLU Function

$$f(x) = max\,(0, x) \quad \text{(Equation 3.3)}$$

The reason for introducing ReLU's is to introduce non-linearity in our CNN. As real-world problems are non-linear So this non-linear data would want our CNN to pick up non-negative linear values. ReLU is a pixel-wise operation. A ReLU function replaces all negative pixels to zero.

3.4.4 Fully Connected Layers

When Convolutional Layer and the Pooling Layer combined, they collectively form the i-th layer of a Convolutional Neural Network. Depending on the complexities in the images. We can also increase the number of layers if we want to capture low-levels information in more detail, but clearly, this will cost us more computational power. At this point, we are having empowered our model to comprehend the features. Now for the task of classification, we will flatten the final output so it can be provided to a regular Neural Network. The purpose of adding a Fully-Connected layer is an economical way of learning high-level features non-linear combinations of the output of the convolutional layer. We will flatten the image into a column vector i.e., an appropriate form for our Multi-Level Perceptron. Then finally the flattened output image is sent to a feed-forward neural network. Through the backpropagation method which is applied to every training epoch. Over a series of iteration, the model can now discriminate among low-level features in images and with help of Softmax [30] or sigmoid Classification technique model can classify them.

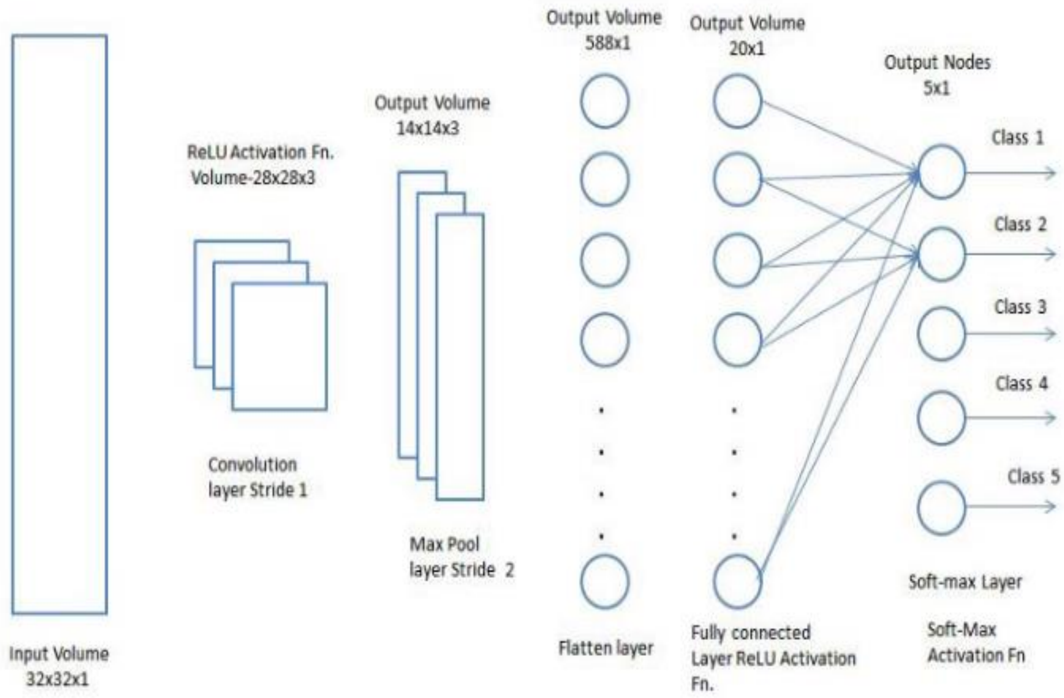


Figure 3.12 After pooling layer, flattened as Fully Connected Layer

3.4.4.1 The fully connected input layer

This fully connected input layer is otherwise known as the flattening of the images. In this layer the outputs obtained from the last layer, the feature map matrix will be flattened and converted into a single vector so that it can be used as the input data for the forthcoming layer.

3.4.4.2 Fully connected layer

After the feature analysis has been done in previous layers and it's time for calculation, this layer finally allocates random weights to the inputs and make a prediction for a suitable label.

3.4.4.3 Fully connected Output layer:

This is the final layer of the CNN model which contains the results of the labels determined for the classification and assigns a class to the images [24].

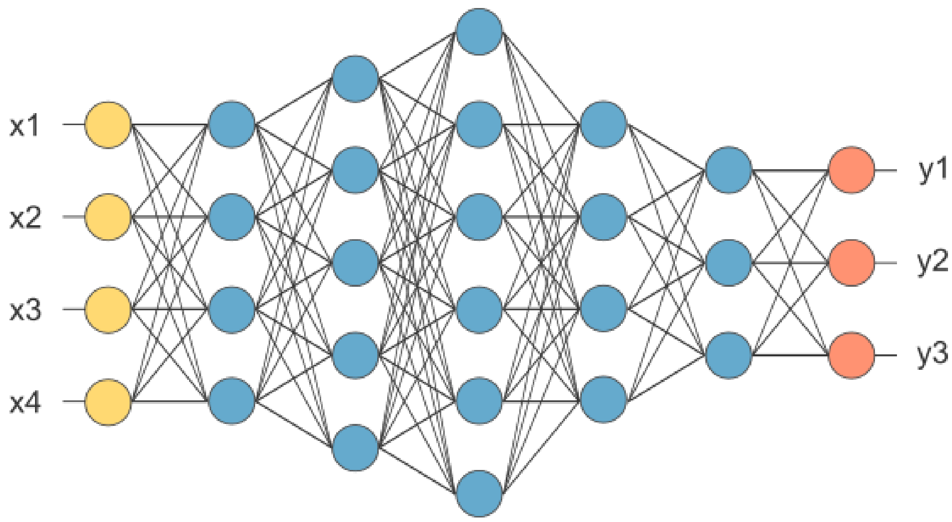


Figure 3.13 Fully Connected Layers

3.5 Complete CNN Architecture

The basic role of the Convolutional Neural Network model is to reduce the images into a form which is easier to process while maintaining dominant feature which is very crucial for deciding a good right class prediction. Together these layers extract the useful features from the images, introduce non-linearity in our network and reduce feature dimension while aiming to make the features somewhat equivariant to scale and translation. A complete Convolutional Neural Network will be this way

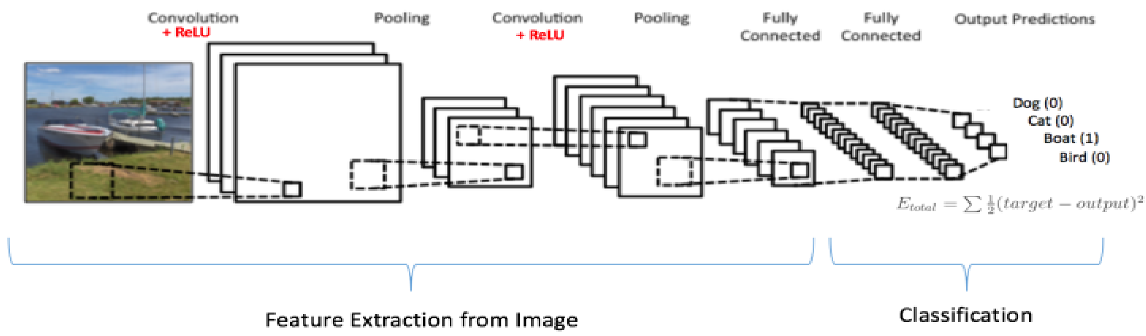


Figure 3.14 Complete CNN architecture

3.6 Dropout

Dropout is a technique used to prevent the problem of overfitting. In this technique during the training process arbitrarily selected neurons are ignored, thus "dropped-out" randomly. This means that the contribution of these arbitrarily selected neurons to the activation of downstream neurons is temporally detached on the forward pass and also on backward pass any updating of weight is not applied on the neuron[31]. Dropout is training a neural network by ignoring some of the neurons at certain iterations Figure 3.14. Thus, with the help of this technique, non-ignored neurons can make learning in better way.

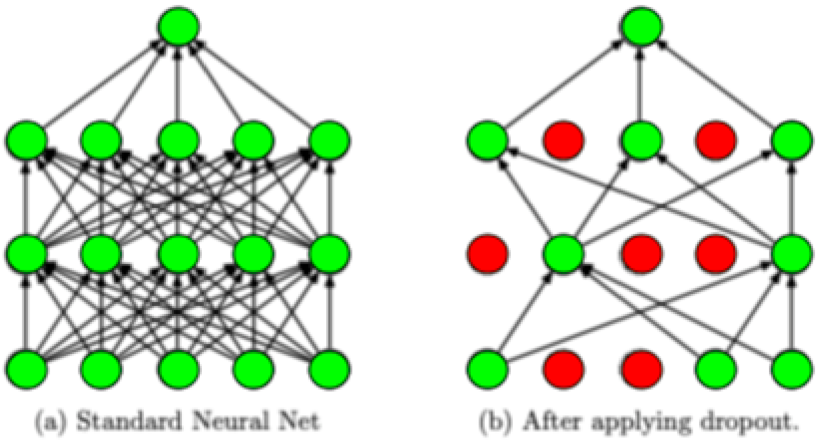


Figure 3.15 Depiction of Dropout Technique

3.7 Batch Normalization

The purpose behind applying this technique is to increase learning speed in classification tasks as we know already the datasets used in deep learning are massive, the input of the network fluctuates meaningfully. So as the name proposes, Batch normalization rather than normalizing the whole dataset as a pre-processing method it normalizes the data in the small-batch and making the normalization part of the network. Furthermore, using Batch Normalization[32] there will be no longer a need for dropout layers since batch normalization can be used as a regularizer.

CHAPTER 04

BRAIN TUMOR SEGMENTATION

4.1 Image Segmentation

To properly localize the image features at pixel level and partition the image into segments based on same pixel intensities assigned one class label and other pixel are assigned another class label. Pixel level segmentation helps to localize the interested part more clearly. Segmentation task is more difficult than the classification and detection. In detection, only the bounding box represents class. Bounding boxes can have different pixels other than the interested class but in case of segmentation the outline around the segmented part contains only the interested region as shown in Figure 4.1.

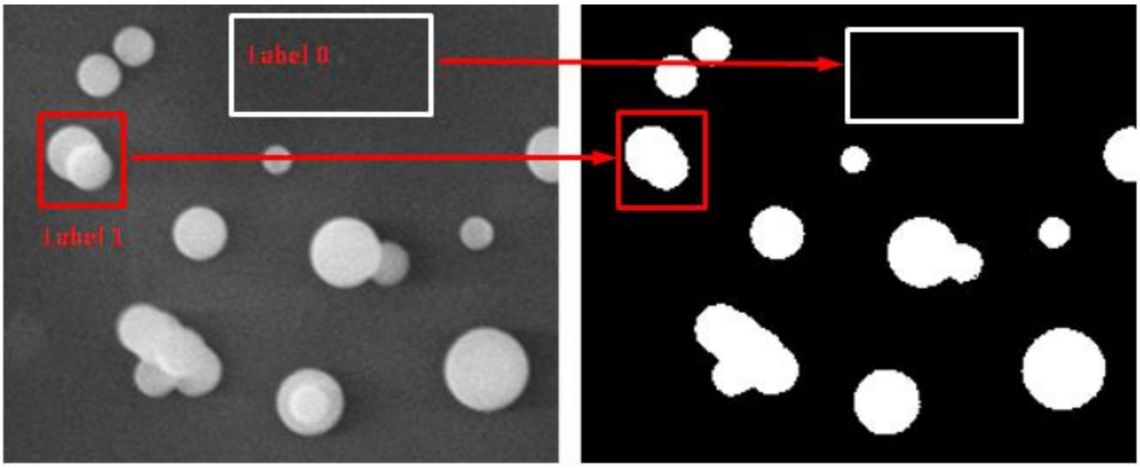


Figure 4.1 In this binary image segmentation [38], Label 1 is mapped to interested region and the label 0 is as a background. This is an easy segmentation task as compared to multiclass segmentation where more classes are classified

4.2 Importance of Brain Tumor Segmentation on detection

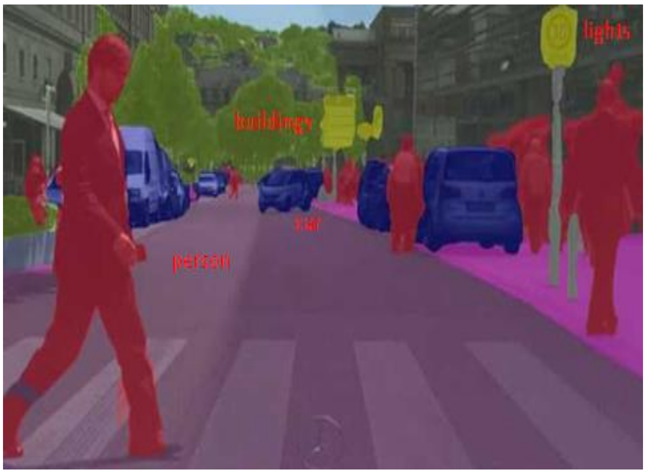
In our work, we have not only detected the brain tumor disease but we also segmented that where is brain tumor. This will help to doctors to completely analyze the brain region where there is tumor. Brain tissues are very sensitive to radiations. If non tumorous region is affected because of false positive segmentation of brain tumor, this can cause a lot of problem to the patient. Brain Tumor segmentation. Every region with its unique color represents the type of tumor. Labeling with numbers 1,2, and 3 is also assigned to each class of tumor. Label 4 is assigned as background with non-tumorous region. This image helps doctors to analyze the image more deeply and focused on more dangerous parts as shown in Figure 4.2 (b).

4.3 Types of Segmentation

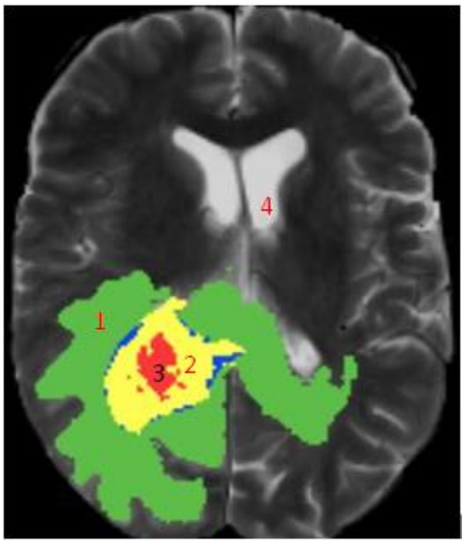
There are two types of segmentation which are mostly used. Segmentation of image is done by different architectures. We will discuss shortly two types of segmentation, semantic segmentation and instance segmentation.

4.3.1 Semantic Segmentation

In semantic segmentation, same type of objects assigned one class label. In this segmentation task, Higher level of image understanding is required. We need such algorithms which performs well at pixel level and can create boundary on the bases of different intensities, context, texture etc. We have done semantic segmentation. We have four different classes of tumor to segment as shown in Figure 4.2(a).



(a)



(b)

Figure 4.2: In (b) Brain Tumor segmentation. Every region with its unique color represents the type of tumor. Labeling with numbers 1,2, and 3 is also assigned to each class of tumor. Label 4 is assigned as background with non-tumorous region.

This image helps doctors to analyze the image more deeply and focused on more dangerous parts. In (a), same class of objects are segmented with same color. Person is segmented by the red color, car is segmented by the blue color, buildings are segmented by the yellowish color, lights are segmented by another color as mentioned with its name [39].

4.3.2 Instance Segmentation

Instance segmentation is same as the semantic segmentation. But in this segmentation, the architecture will find out the instance with same class labels and separate them with different colors. This will help to analyze the image more concretely and accurately. To perform such tasks in real applications, the dataset with lot of variations of same label are required. This will also lead to more training time, More, deeper network, the more time it will take to learn the features and can lead to over fitting problem. We have not used this technique because our problem is restricted to semantic segmentation. Instance segmentation can only be done through large amount of data and we have not enough data to do.

4.4 2D vs 3D Brain tumor segmentation

We have briefly explained 2D and 3D brain tumor segmentation and why we used 2D approach to detect brain tumors and what are the issues while using 3D segmentation.

4.4.1 2D Brain tumor segmentation

In 2D brain tumor segmentation, each slice of brain is segmented separately. This will separate out the tumorous region from the non-tumorous region. 2D segmentation takes less time for training and results are very efficient in this case but this is not effective for real time application. Because this is laborious task to analyze each and every slice of brain image separately and then analyze and combine the whole volume. To find the tumorous part of the brain becomes impossible to analyze. Any mistake can lead to the most serious issues. Doctors will use their domain knowledge to analyze the brain image. In this case, 2D image segmentation is not good for practical applications as it can lead to many errors. But 2D approach is easy to implement and one can combine different axial views to see the whole brain tumorous region.

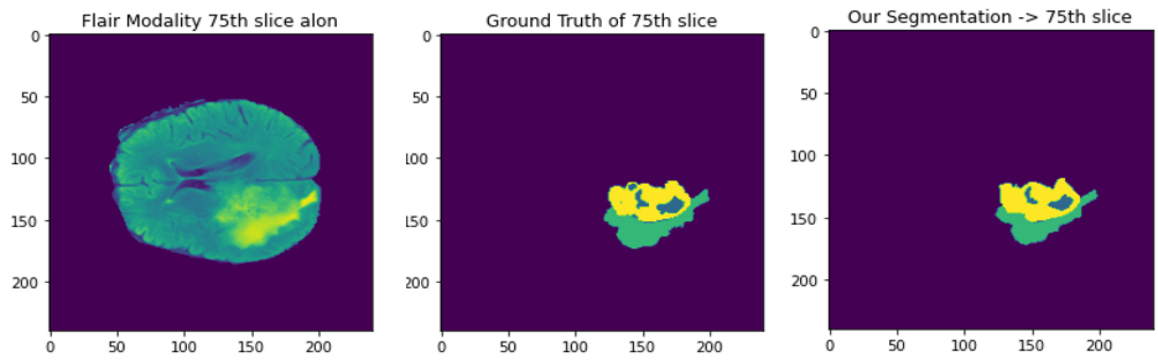


Figure 4.3: Flair modality 75^{th} 2D slice is used here to segment

In Figure 4.3 , On the very right side, there is segmented region which our architecture has done. In the middle, ground truth of the slice is given. Results are much comparable

4.4.2 3D Brain tumor segmentation

In 3D brain tumor segmentation, not each slice is segmented separately as in case of 2D segmentation. Here the whole volume of brain is used as input to the model and model give the whole segmented volume as the output. This will help in dealing with the issues as we have faced in the above segmentation techniques. The whole volume can be observed simultaneously and correctly identified the image.

4.5 Segmentation Architectures

There are multiple segmentation architectures which are used in deep learning. Most useful and recent architectures which works best for segmentation are described briefly. We have explained, why some architectures are superior than others and also explained that why we choose U-NET architecture for brain tumor segmentation.

4.5.1 FCN

Fully Connected Convolutional Neural networks are powerful. We show that convolutional networks by themselves, trained end-to-end, pixels-to-pixels, improve on the previous best result in semantic segmentation [35]. Fully connected convolutional neural networks works best for semantic segmentation. This architecture has convolutional layers with 3*3 filters, followed by a max pooling layer to reduce the size of the image and then further followed by an activation function.

Down-sampling extracts the low and high level features of the image. After extracting the low and high level features of an image. Next task is to segment the image pixel-wise which is done by the expanding path where image is up-sampled with transpose of convolutional layers. This architecture is simple as compared to other architectures as it has no skip connection concatenation like in other architectures.

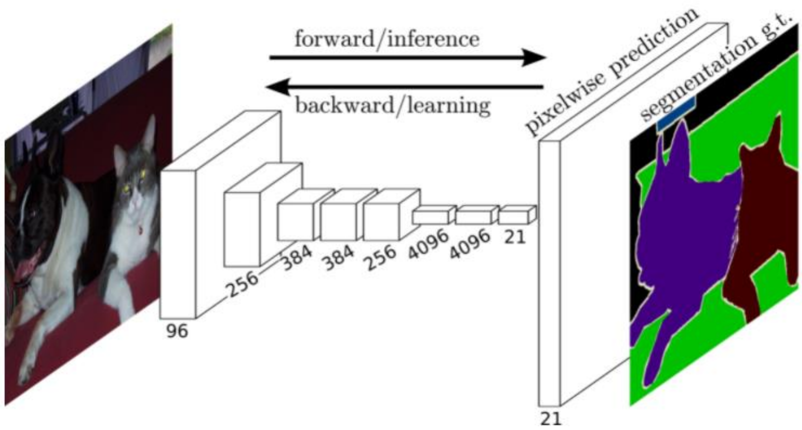


Figure 4.4: 2D RGB input image results a semantic segmented classes. Architecture have a forward path to learn the features of an image and then back propagate to adjust the parameters of convolutional filter. Only convolutional layers are used with no flatten layers [35].

4.5.2 U-NET

UNET architecture is named after its U like shape. This architecture [33] is inspired to deal with specifically biomedical image segmentation. Because in medical field, Large amount of data is not available easily. In case of large amount of data, neural networks can be trained for months to extract the features.

Working with small dataset is not as easy as in large amount of data. After many researches, they designed an architecture which helps in working with small amount of data. U-NET Architecture is also encoding decoding path architecture like FCN. It has a contracting path which down-samples the image and extract the low- and high-level features of an image.

Contracting path [52] consists of convolutional layers with 3*3 filters, max-pooling layer and then non-linear activation function. Contracting path is all about to find the number of features. This grouped all features in one bottleneck. After this, expanding path with transpose convolution is used to locate the features with concatenation of same shape contracting layers with some crop. At the last layer, 1*1 convolutions are used to segment the class.

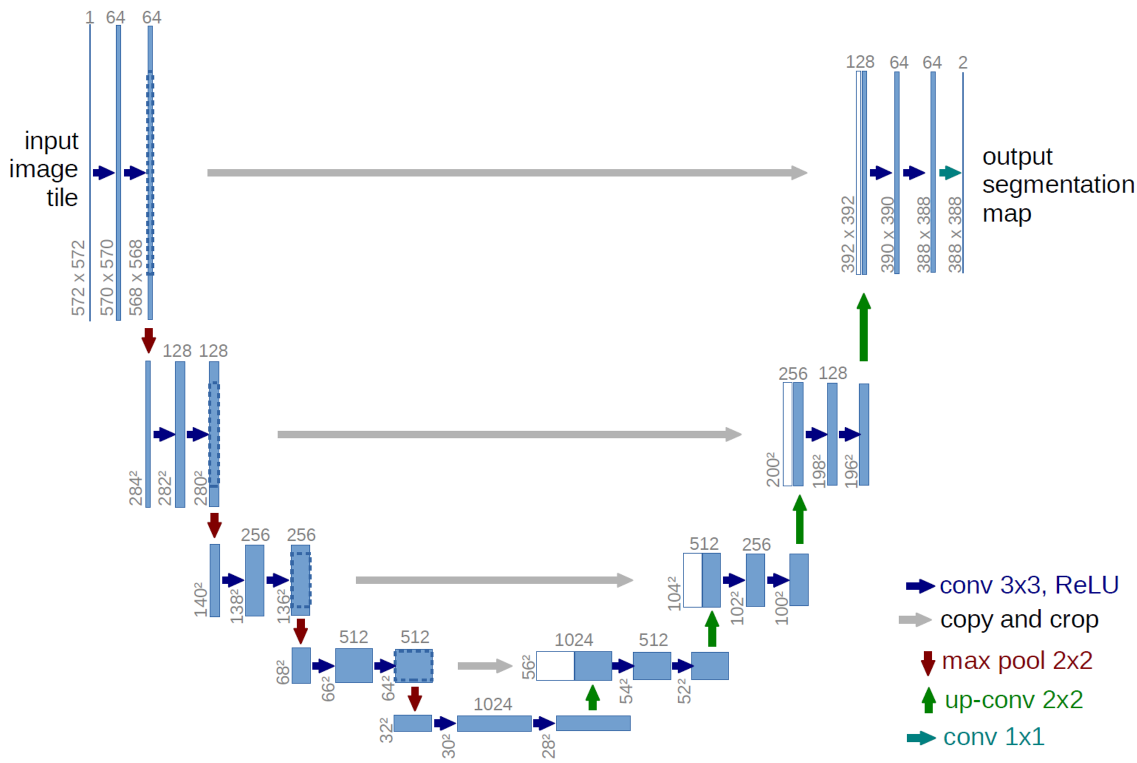


Figure 4.5: input image is gray-scale followed by 2 consecutive convolutions with 64 filters each. This then max-pooled to reduce the image size. Bottleneck with 1024 filters have all combined features. This is 'what' part of the image. After bottleneck, it has 'where' part of the image which uses skip connections and concatenate to locate interested region [33].

4.5.3 Faster R-CNN

Faster R-CNN [34]is an extension to fast R-CNN with an addition of region proposal network to propose regions of interest in the region proposal feature map. Like other architectures, this also uses the fully convolutional layers not dense layers. This architecture has convolutional layers with 3*3 filters, followed by a max pooling layer to reduce the size of the image and then further followed by an activation function. Down-sampling [49] extracts the low and high level features of the image. After extracting the low and high level features of an image. Next task is to segment the image pixel-wise which is done by the expanding path where image is up-sampled with transpose of convolutional layers.

The RPN is designed to be trained end to end to generate not only the class but also predict the object scores for each pixel. In faster R-CNN, image is passed into the convnet. This will find the areas of interest and so called region proposals to create anchor boxes on the image. The center of the anchor box comes from the coordinate of the sliding window and the boundaries of the box comes from the RPN, giving us a score that the boundaries of the box better fit the object.

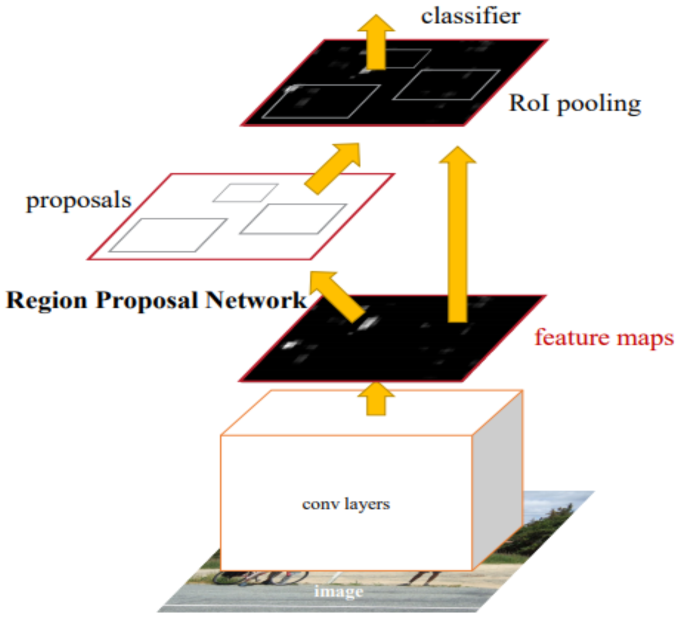


Figure 4.6: Faster R-CNN [34] is a single, unified network for object detection. The RPN module serves as the 'attention' of this unified network. The center of the anchor box comes from the coordinate of the sliding window and the boundaries of the box comes from the RPN, giving us a score that the boundaries of the box better fit the object.

4.6 Segmentation Losses

Loss function [48] plays an important role in any machine learning model. Loss function tells how much the model is fitted to training data. We always prefer to have very lower value of loss. Zero value of loss indicates 100 % model accuracy on training data. There are different types of loss functions depend upon the model. Model parameters changes after every iteration depend upon the loss value. Segmentation losses are different than classification losses. Major semantic segmentation losses which helps us in our model good or bad are given below:

4.6.1 Dice Score Loss

The Dice coefficient [50] is widely used metric in computer vision community to calculate the similarity between two images. $\boldsymbol{DSC} = \frac{2TP}{2TP+FN+FP}$

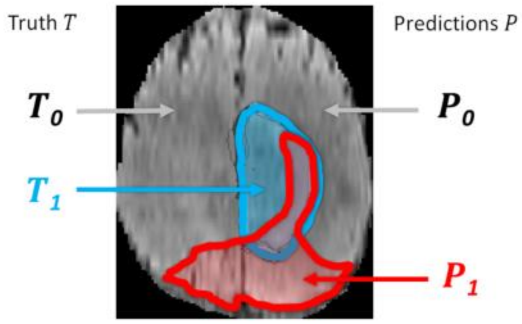


Figure 4.7: T*o* is truth region [45] where no tumor, T1 shows the ground truth region where tumor exists shown by blue color, Po is prediction where tumor not exists, P1 is the predicted tumor region shown by red color

4.6.2 Sensitivity Specificity Loss

Similar to Dice Coefficient, [40] Sensitivity and Specificity are widely used metrics to evaluate the segmentation predictions. In our application, dataset is very imbalanced. To overcome that issue, w is used to adjust loss to understand the model and make decisions accurately. TP is true positive and TN is true negative. The loss [40] is defined as:

$$\mathbf{SSL = w \times sensitivity + (1 - w) \times specificity}$$

Where,

$$\boldsymbol{Sensitivity = \frac{TP}{TP + TN}}$$

And

$$\boldsymbol{Specificity = \frac{TN}{TN + TP}}$$

4.7 Proposed U-NET Model Architecture

We have used U-NET architecture for practical implementation. This architecture works best for the biomedical image segmentation. The whole implementation of Up-sampling and down-sampling layers are explained in section 4.6.1 and 4.6.2 below:

4.7.1 Down-Sampling

3D input image [45] of axial view is passing through 2 convolution layers with 16 filters in first convolution layer and 32 filters in second convolution layer, batch normalization is also used after each convolution for generalizing the pixel values. One max-pooling layer is used to reduce the image size double.

4.7.1.1 Layer 1 and Layer 2 followed by Max-pooling

3D input image of axial view is passing through 2 convolution layers [49] with 16 filters in first convolution layer and 32 filters in second convolution layer, batch normalization is also used after each convolution for generalizing the pixel values as shown in Figure 4.8. One max-pooling layer is used to reduce the image size double.

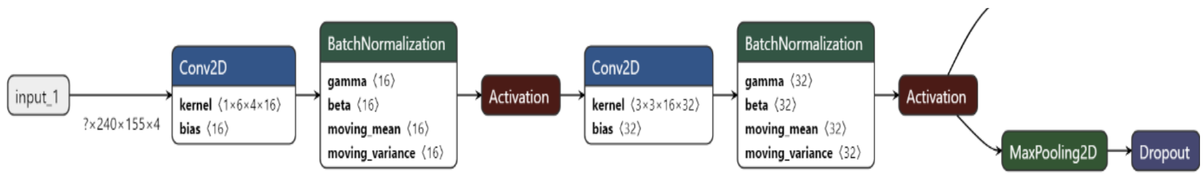


Figure 4.8: Layer 1 and Layer 2 followed by Max-pooling

4.7.1.2 Layer 3 and Layer 4 followed by Max-pooling

Layer 2 input image of axial view is passing through 2 convolution layers with 64*64 filters in first convolution layer and 64*128 in second convolution layer, batch normalization is also used after each convolution for generalizing the pixel values as shown in Figure 4.9. One max-pooling layer is used to reduce the image size double

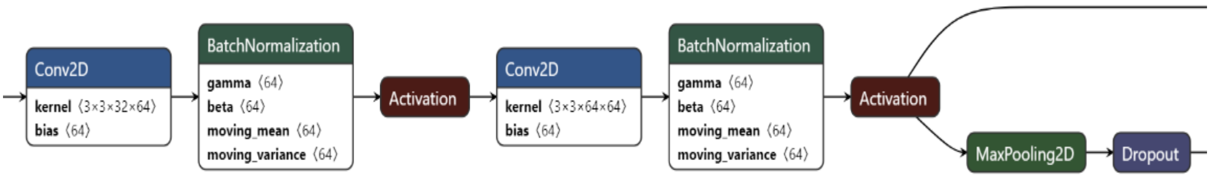


Figure 4.9: Layer 3 and Layer 4 followed by Max-pooling

4.7.1.3 Layer 5 and Layer 6 followed by Max-pooling

Layer 4 input image of axial view is passing through 2 convolution layers with 64*128 filters in first convolution layer and 128*128 in second convolution layer, batch normalization is also used after each convolution for generalizing the pixel values as shown in Figure 4.10. One max-pooling layer is used to reduce the image size double.

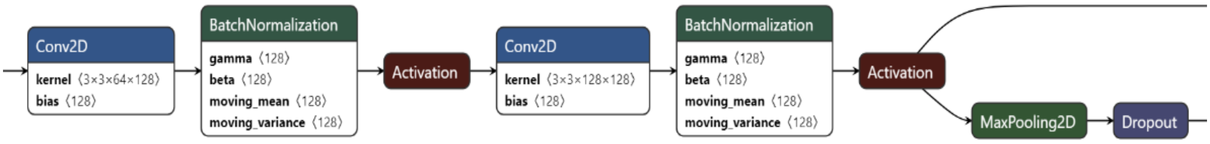


Figure 4.10: Layer 5 and Layer 6 followed by Max-pooling

4.7.1.4 Layer 7 and Layer 8 followed by Max-pooling

layer 6 input image of axial view is passing through 2 convolution layers with 128*256 filters in first convolution layer and 256*256 in second convolution layer, batch normalization is also used after each convolution for generalizing the pixel values. One max-pooling layer is used to reduce the image size double. Line above shows the skip connections which will concatenate in the up-sampling layers as shown in Figure 4.11.

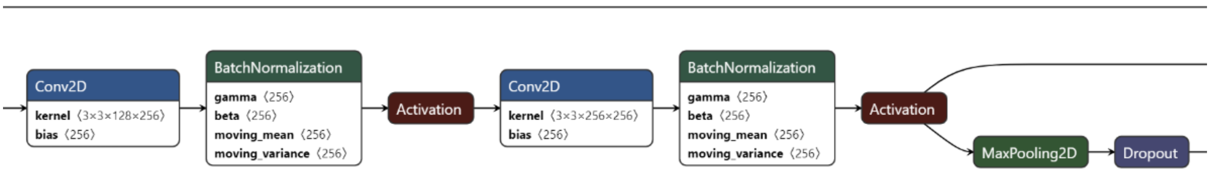


Figure 4.11: Layer 7 and Layer 8 followed by Max-pooling

4.7.2 Bottleneck

After number of down-sampling convolution layers. Bottleneck has all the features of the image. Convolution filters have extracted the 'what' part of the input image which means that it has all the information about the image but it has no information regarding location of all features. Features will group in next section of up-sampling and grouped them at their interested locations. Intensity imbalance is the major issue in MRI images. Next section of Up-sampling will deal this issue with concatenation of skip layers from down-sampling images. After this, up-sampling operation will start. Bottleneck portion is shown in Figure 4.12.

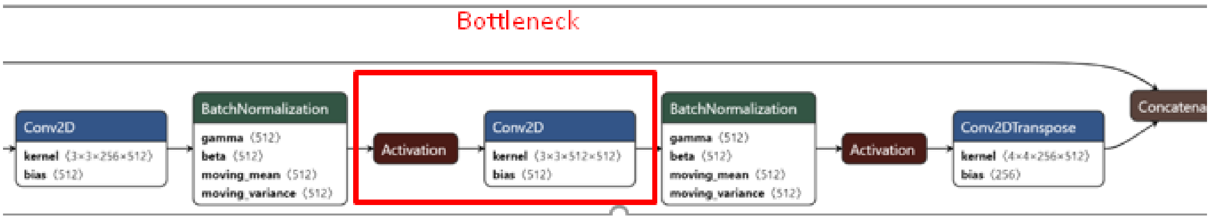


Figure 4.12: Bottleneck layer

4.7.3 Up-Sampling

In Figure 4.10 conv2D transpose is used which is totally opposite to the max-pooling layer. In this operation, one dimensional feature vector is converted into 2 dimensional matrices, skip connections from the down-sampling layers are now concatenated and features are further explored two convolutional layers are used same as in down-sampling with some drop-out. The cropping is necessary due to the loss of border pixels in every convolution. At the final layer a 1x1 convolution is used to map each 64- component feature vector to the desired number of classes [36]. To minimize the overhead and make maximum use of the GPU memory, we favor large input tiles over a large batch size and hence reduce the batch to a single image [35].

4.7.3.1 conv2D transpose operation after bottleneck

conv2D transpose is used which is totally opposite to the max-pooling layer. In this operation, one dimensional feature vector is converted into 2 dimensional matrices, skip connections from the down-sampling layers are now concatenated and features are further explored two convolutional layers are used same as in down-sampling with some drop-out as shown in Figure 4.13.

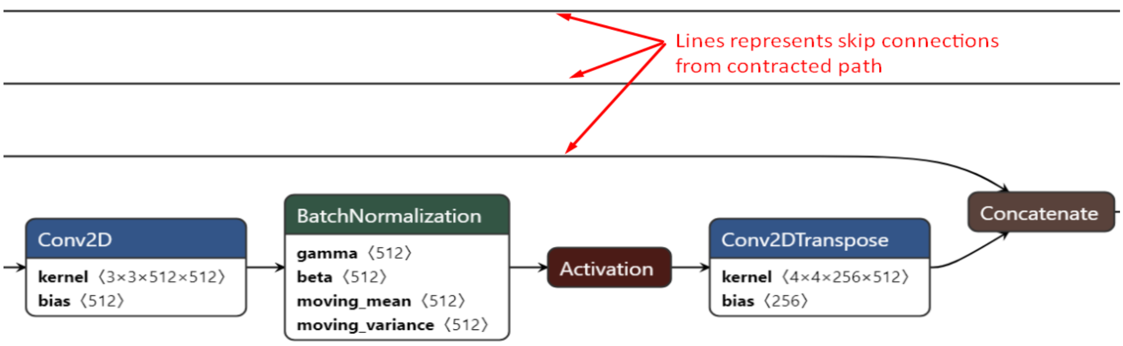


Figure 4.13 conv2D transpose operation after bottleneck

4.7.3.2 conv2D skip connections and concatenation

Skip connections from the down-sampling layers are now concatenated and features are further explored two convolutional layers are used same as in down-sampling with some drop-out. Features are now localized. Concatenation layers and conv2D transpose layers are shown in Figure 4.14.

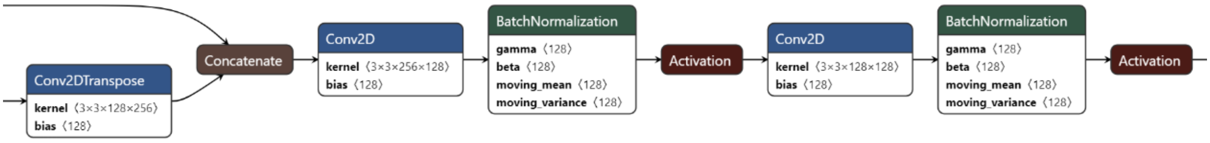


Figure 4.14: conv2D skip connections and concatenation layers with ReLu activations

4.7.3.3 Segmentation Layer

After last skip connection is concatenated then comes the part of segmentation. 1*1 convolution layers are used to segment the brain tumorous region and separating the non-tumorous part as shown in Figure 4.15.

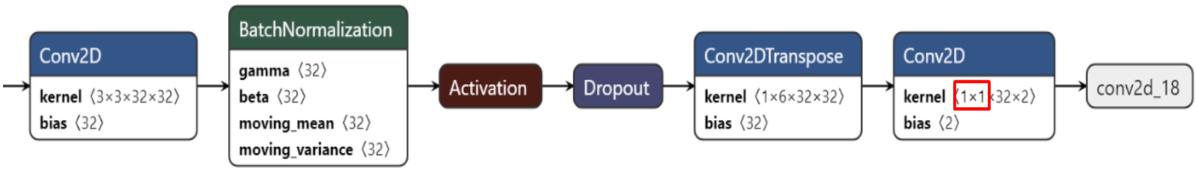


Figure 4.15: segmentation layer with 1*1 convolutions

CHAPTER 05

5.1 RESULTS

5.1.1 Classification using CNN:

As classification is important part as we want to distinguish if an image has a tumor or not. So, we can say if the brain is healthy or not. We used CNN for classification as given in Topic 3.1.

We used Kaggle dataset but out final dataset as explained in chapter 1. We used 10 epochs for training our dataset and with 11000 Training images and 200 Testing images. The accuracy achieved for both Testing and Training is given below:

TABLE 1: CLASSIFICATION OF BRAIN TUMOR

Training and Testing	**Original Dataset**	**Cross-validation Dataset**
Training accuracy	0.981	0.9828
Testing accuracy	0.9900	0.9360
Training loss	0.0551	0.0515
Validation loss	0.0600	0.3913

Graph and results for classification:

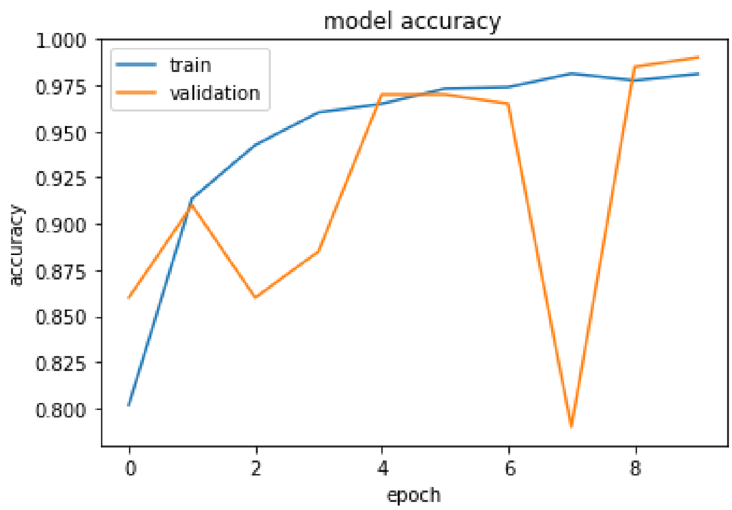


Figure 5.1 Testing and Training accuracy

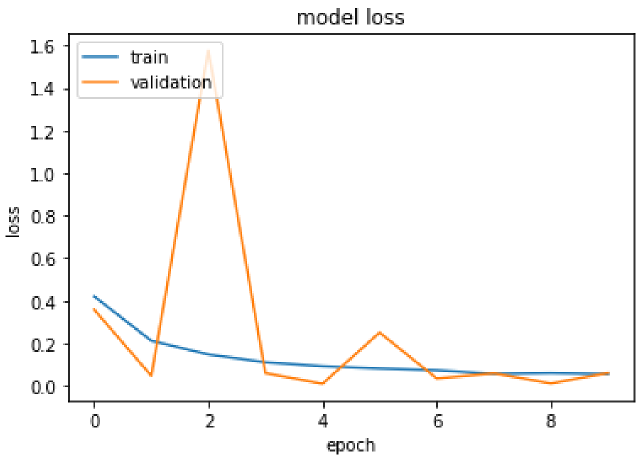


Figure 5.2: Training and Testing loss

Few Testing images:

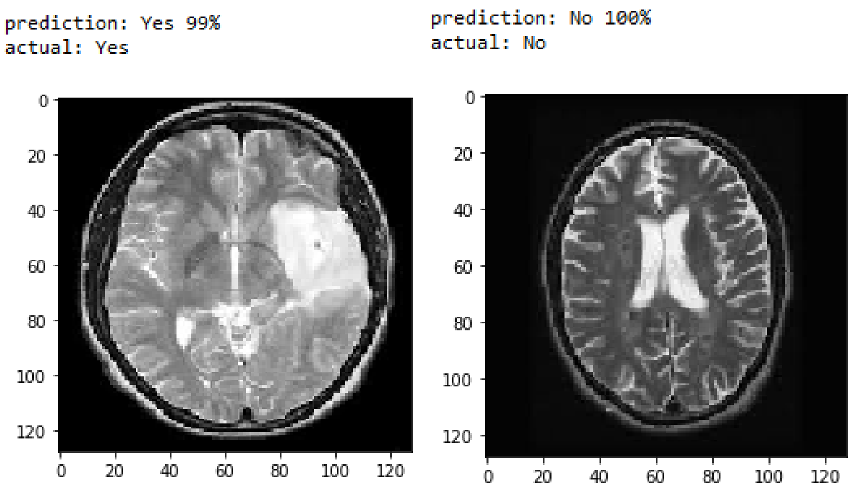
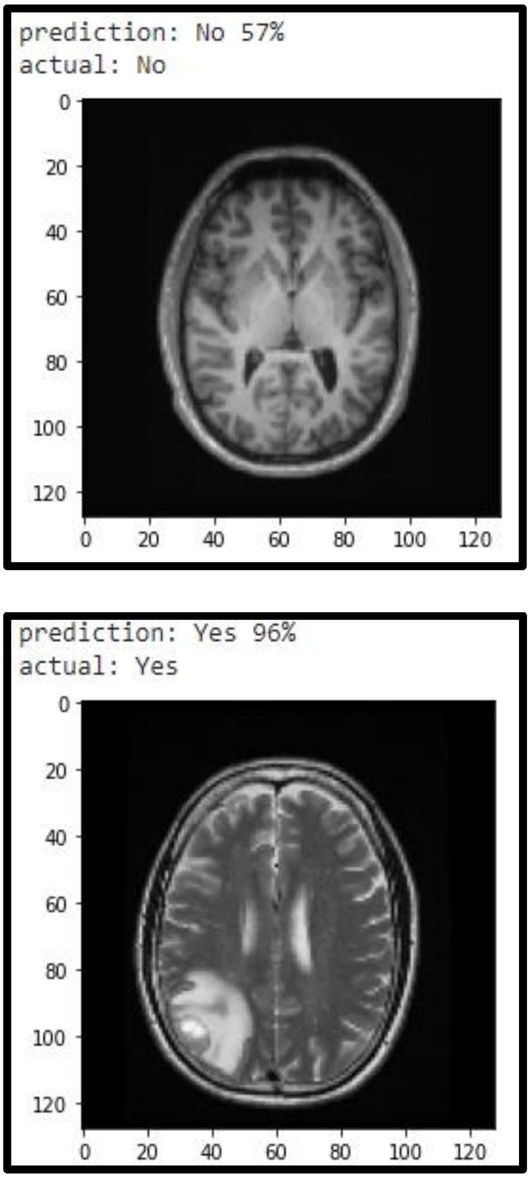


Figure 5.3 Classification Results

We cross-validated out CNN classification model on another Kaggle dataset of 160 Test images for achieving the result. As we can see from the table that the CNN model we used and the number of layers are giving us the best result in after cross validation proving the great efficiency and predictability of our model.

5.1.2 Classification outputs:

The classification results we have gotten are as follows



(a)

Figure 5.4 Classification Results

5.1.3 Segmentation Results:

We achieved segmentation using UNET model as explained in chapter 4. We used Brats 2018 dataset consisting of 210 3D images. As we will slice, the 3D images and take images from 3 axis so the total number of 2D images come out to be 151200.The parameters and model for all them were kept the same due to its suitability with the dataset and we trained the data for a little bit more and achieved the following result:

TABLE 2: SEGMENTATION OF BRAIN TUMOR

Training and Testing	**Two Class Segmentation**	**Four Class Segmentation**	**Four Class Segmentation Improved**
Epochs	10	10	45
Training accuracy	0.9726	0.7526	0.7526
Training loss	0.0274	0.2474	0.2474
Testing accuracy	0.9222452600797018	0.6383521348237992	0.7266646613677342
Testing loss	0.07775473992029826	0.3616478661696116	0.27333533962567647
Mean Sensitivity for class 0	0.9987968804288612	0.9973130890962196	0.9994418608006598
Mean Specificity for class 0	0.8157543780263949	0.7399030542635091	0.7855767936044754
Mean Sensitivity for class 1	0.8157543780263949	0.3213817273098429	0.4228020048290975
Mean Specificity for class 1	0.9987968804288612	0.9994933975562182	0.9997933902524028
Mean Sensitivity for class 2	NULL	0.41349304918469726	0.6064759350159152
Mean Specificity for class 2	NULL	0.9996057752590369	0.9993690548406183
Mean Sensitivity for class 3	NULL	0.8404884744002085	0.826595230858187
Mean Specificity for class 3	NULL	0.9955309186827245	0.9985983158385012

5.1.4 Segmentation Outputs of two classes:

Graph Output:

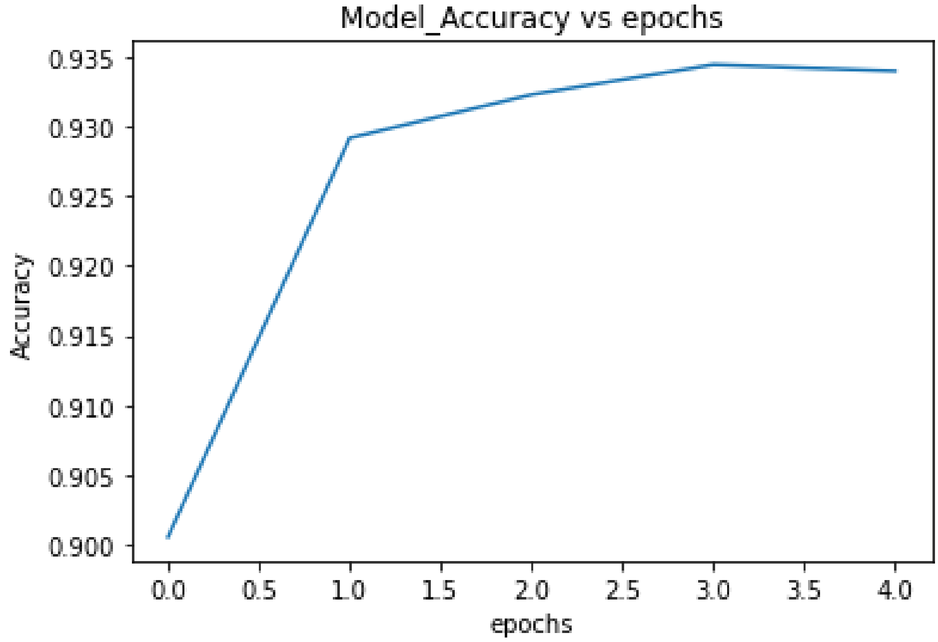


Figure 5.5 Model accuracy vs epochs for 2 classes

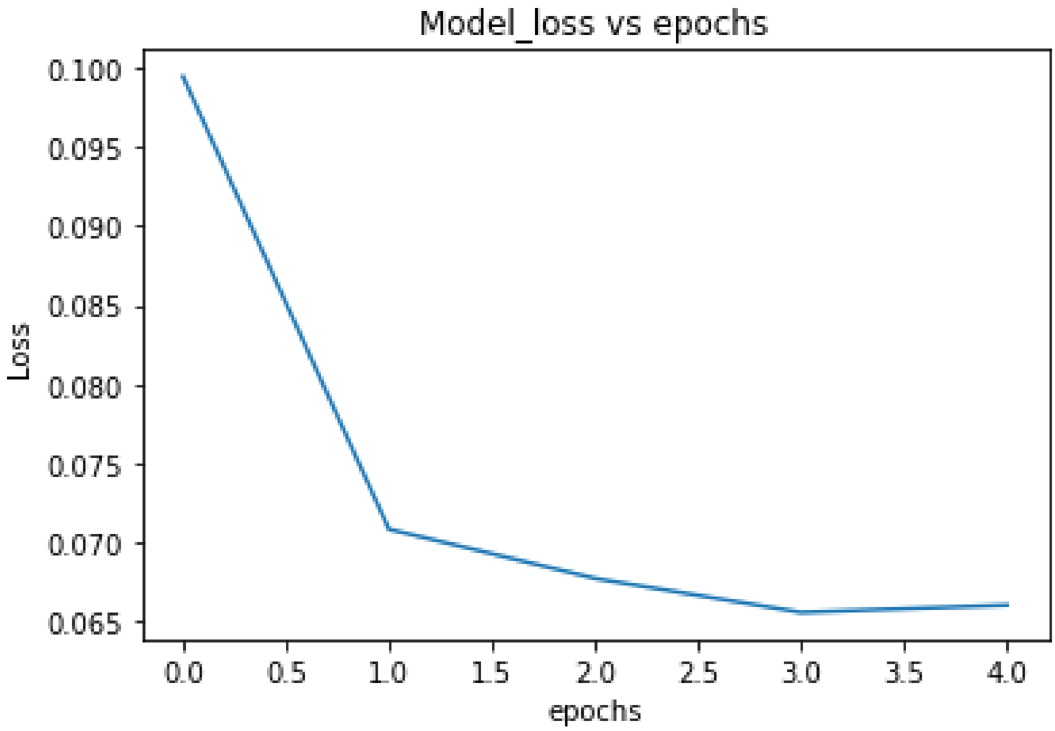
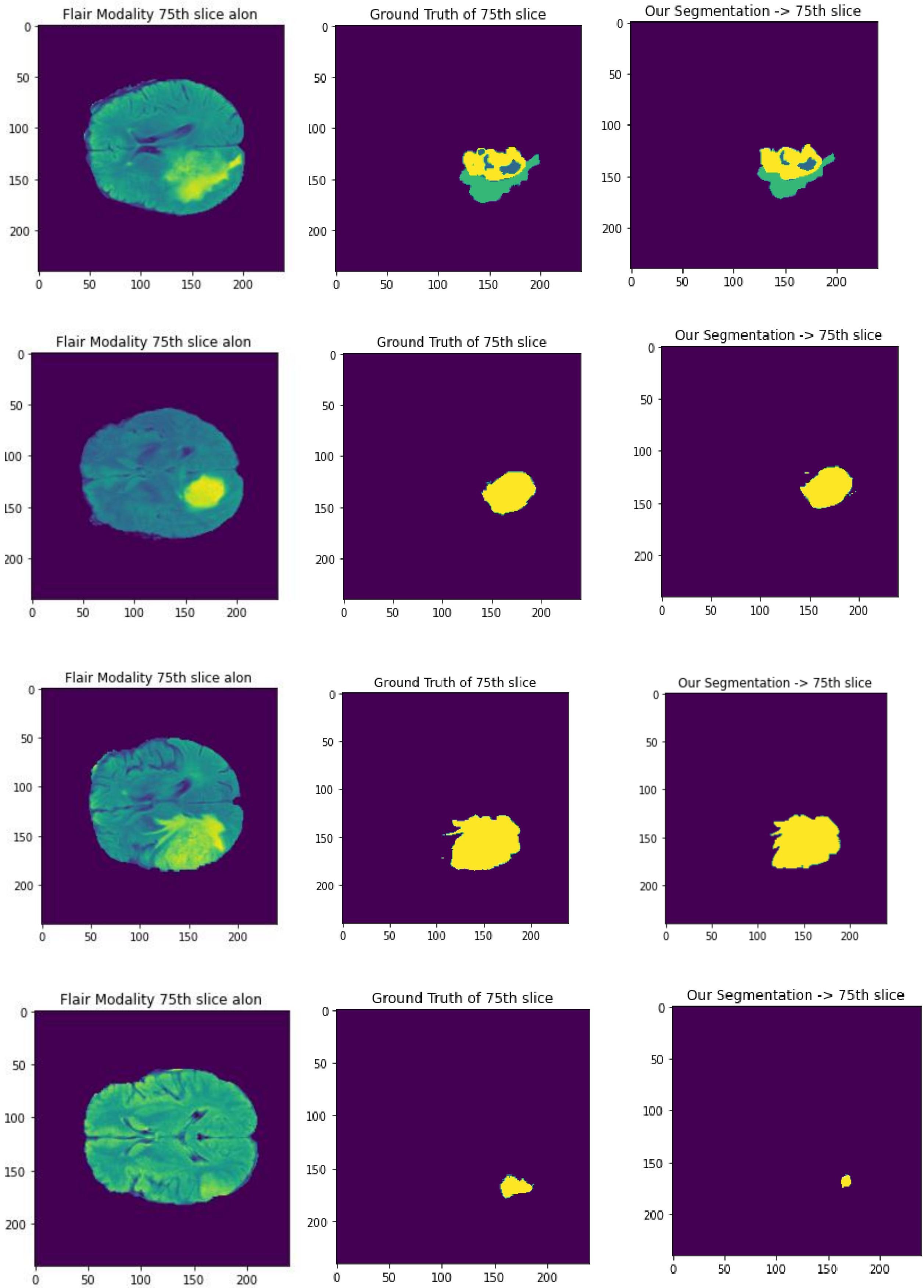


Figure 5.6 Model loss vs epochs for 2 classes

Segmentation Outputs:



5.1.5 Segmentation Outputs For four classes:

Graph Output:

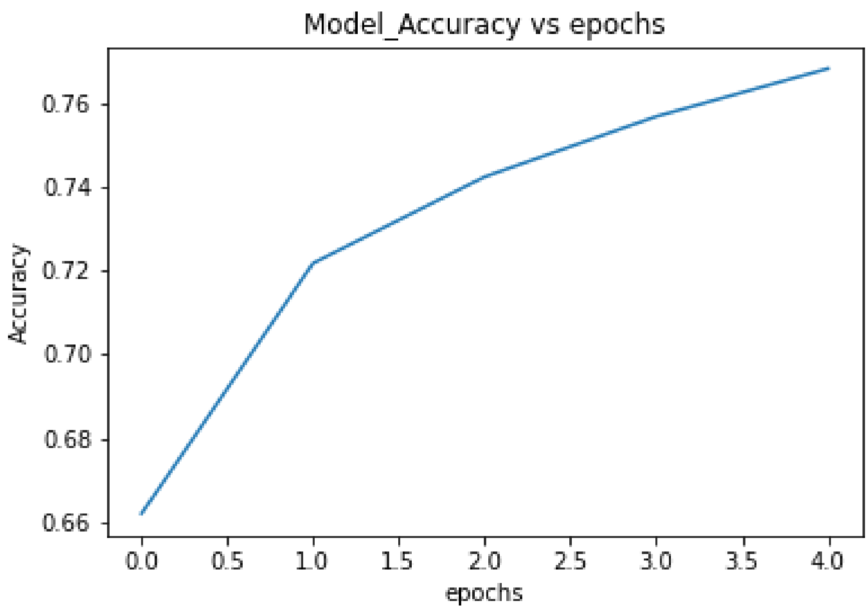


Figure 5.7 Model accuracy vs epochs for 4 classes

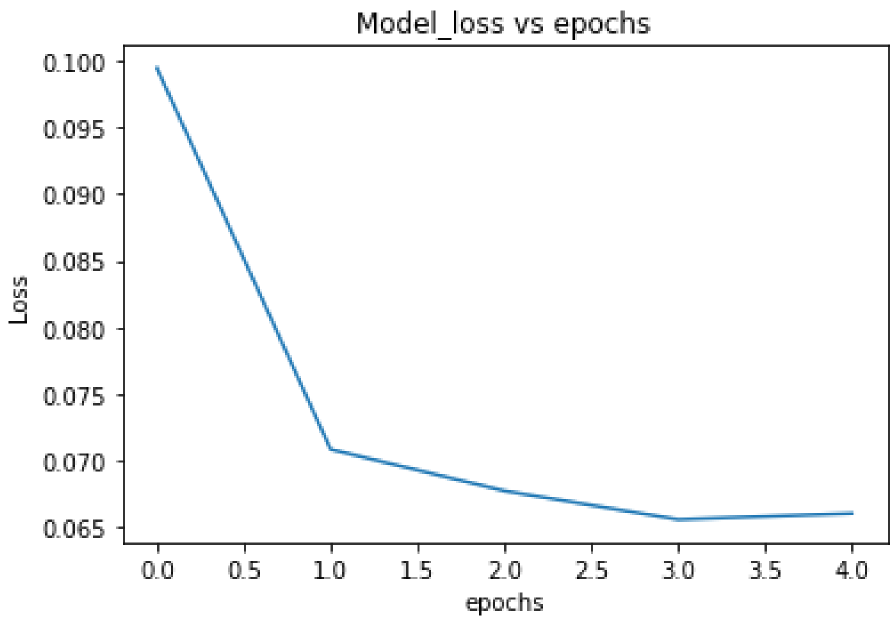
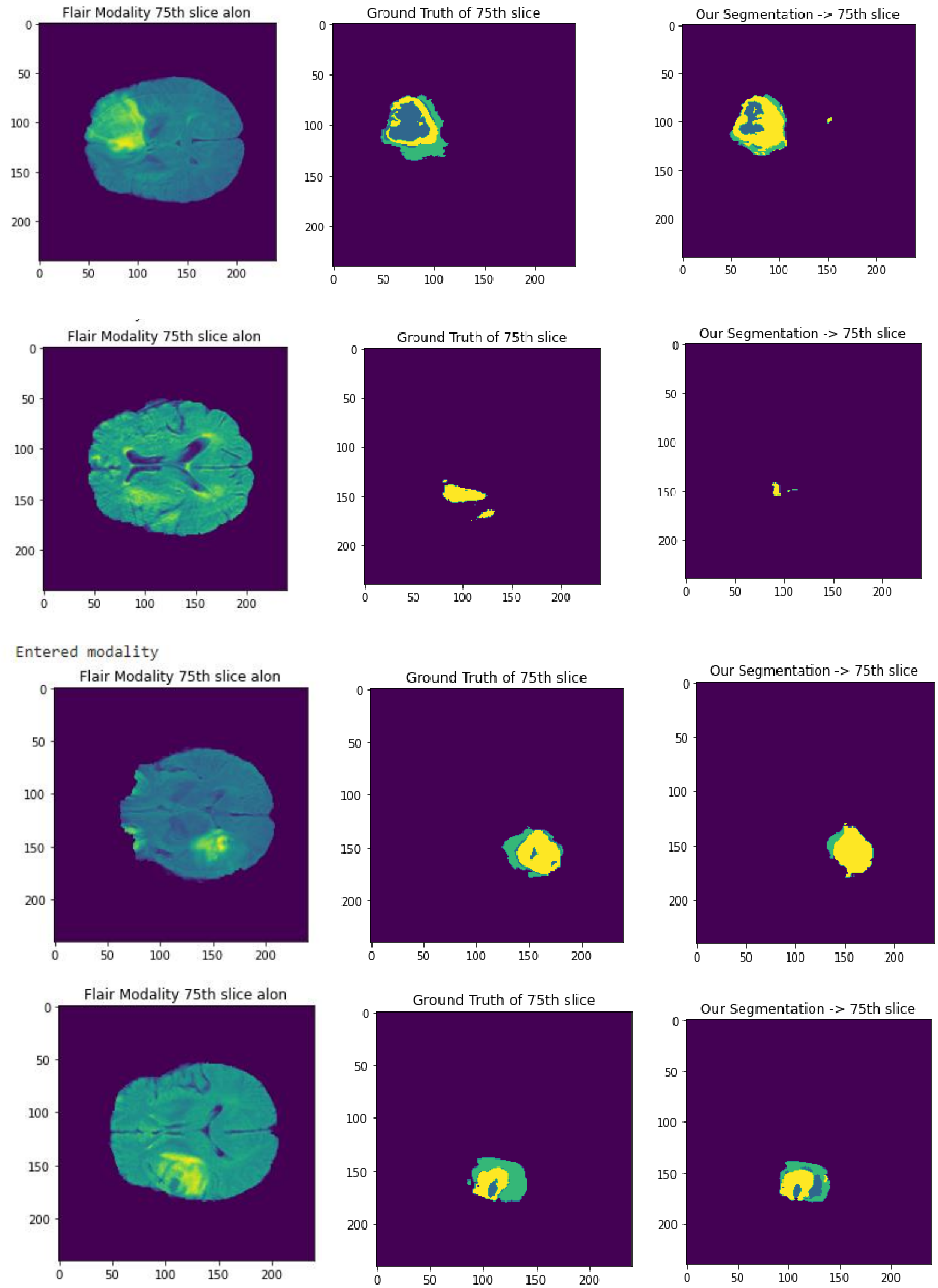


Figure 5.8 Model loss vs epochs for 4 classes

Segmentation Outputs:



5.1.6 Segmentation Outputs for Four classes (Improved):

Graph Output:

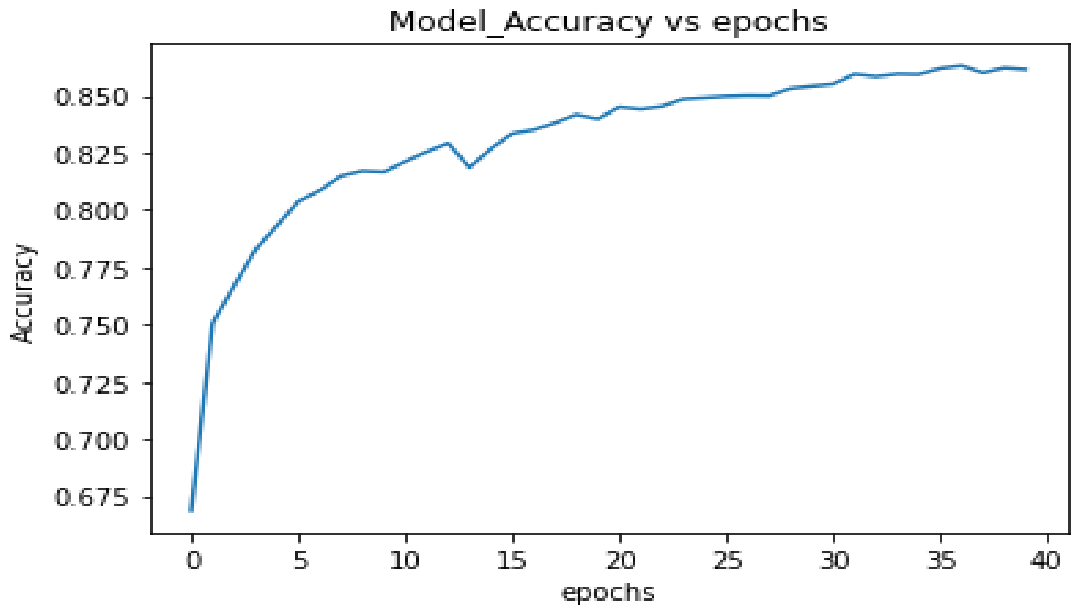


Figure 5.9 Model accuracy vs epochs for 4 classes

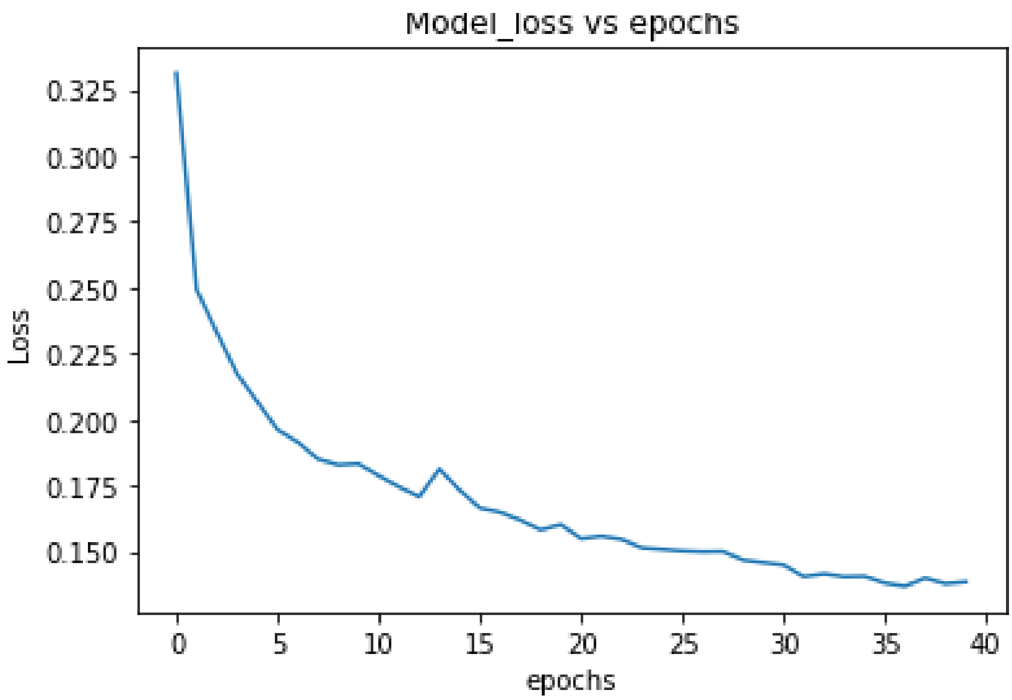
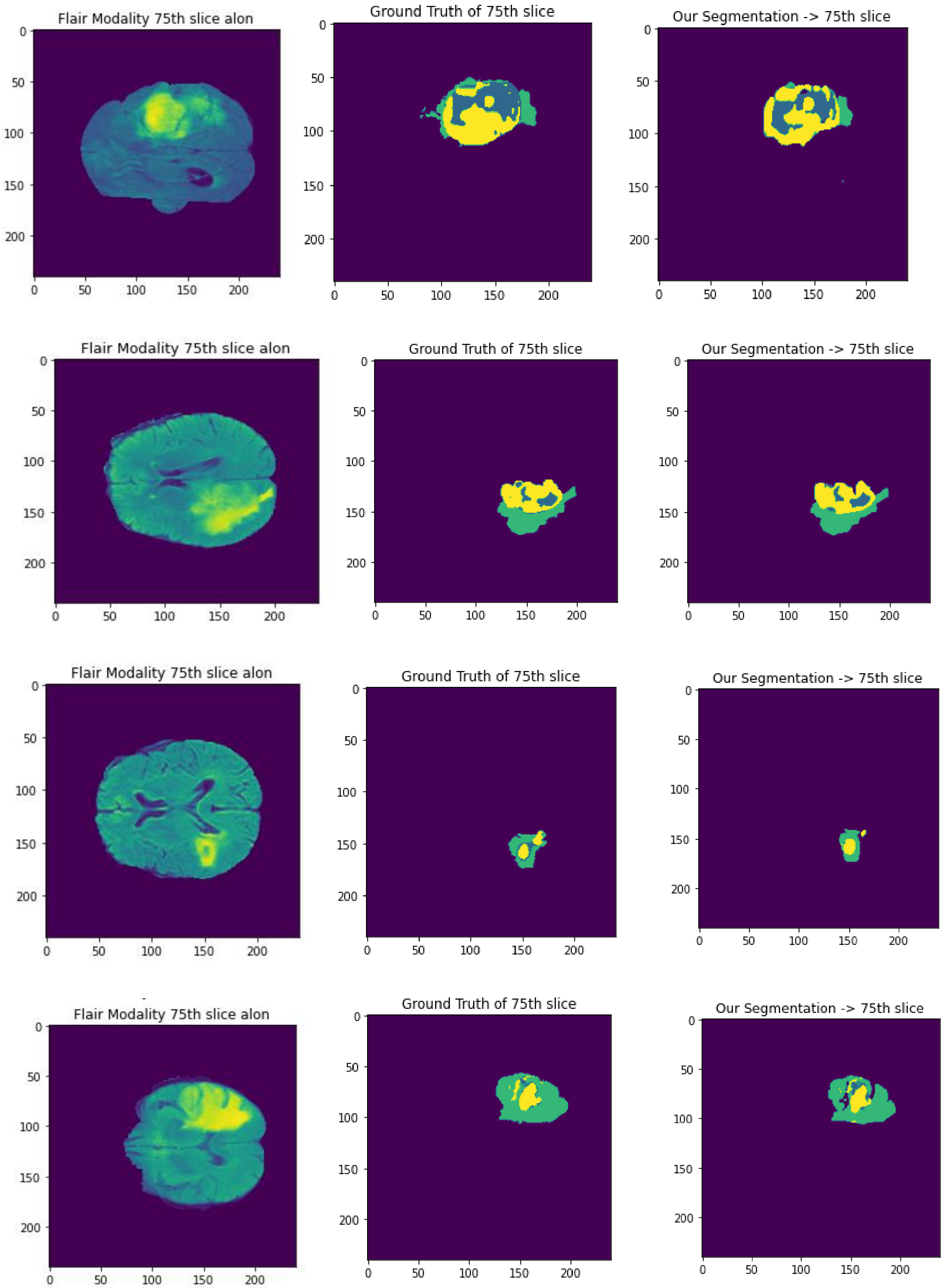


Figure 5.10 Model loss vs epochs for 4 classes

Segmentation Outputs:



5.3 Comparison:

TABLE 3: COMPARISON

Author	Method	Level of user interaction	Performance (Dice Scores)			Mean Dice accuracy
			Whole Tumor	Core Tumor	Active Tumor	
Human Rater	Medical training experience	Manual	0.88	0.93	0.74	Not Reported
Pereira et al	CNN with small (3x3) Filters for Deeper architecture	Fully automatic	0.88	0.83	0.77	Not Reported
Kwon et al	Generative model that performs joint segmentation and registration	semi-automatic	0.88	0.83	0.77	Not Reported
Havaei et al	Cascaded Two-pathway CNNs for simultaneous local and global processing	Fully automatic	0.88	0.79	0.73	Not Reported
Tustison et al	Concatenated RFs, trained using asymmetry and first order statistical features	Fully automatic	0.87	0.78	0.74	Not Reported
Urban et al	3D CNN architecture using 3D convolutional filters	Fully automatic	0.87	0.77	0.73	Not Reported
Havaei et al	Uses SVM; training and segmentation implemented within the same brain	Semi-automatic	0.86	0.77	0.73	Not Reported
Dvorak and Menze	Local structured prediction with CNN and k-means	Fully automatic	0.83	0.75	0.77	Not Reported
Davy et al	Two-pathway CNN for simultaneous local and global processing	Fully automatic	0.85	0.74	0.68	Not Reported

Zikic et al	3D input patches are interpreted into 2D input patches to train a CNN	Fully automatic	0.837	0.736	0.69	Not Reported
Hamamci et al	Generative model, uses cellular automata to obtain tumor probability map	Semi-automatic	0.72	0.57	0.59	Not Reported
Our Work	Unet architecture	Fully automatic	0.9222452600797018	Not calculated	Not calculated	0.7266646613677

As we compare our results from the table 3, our results are a lot better or at least comparable and it can be further improved if necessary, equipment are made available.

CHAPTER 6

6.1 Conclusion

In this thesis we will discuss how we can detect different diseases especially brain tumor using machine learning or deep learning using neural network.

As we can see that many disease detection and diagnosis take, a lot of time due to human limitation but if, we get a little bit of help of the machine we can speed up the detection process and in turn, influence the diagnosis of the disease and it will speed up the process. Early detection can help reduce the chance of getting the disease to a severe stage.

As brain disease is one of those whose detection takes, time and so if not diagnosed in time might cause permanent damage or death. The number of neurologists are very low in Pakistan as well as around the world. Therefore, we decided to work on detection and diagnosis of brain tumor. As it is the most dangerous and common brain disease.

Therefore, we first worked on classification of the brain tumor and studying about different techniques upon which we used CNN one of the best method and achieved the results as discussed above. The other techniques like ANN, FCN, and Inception were giving less accuracy then CNN and were very time consuming.

After the detection knowing the place of the tumor was our priority, so after doing some research on different segmentation techniques such as CNN, ANN, UNET, Fuzzy C Means Algorithm, K means Clustering and Thresholding we decided to go with UNET which was designed specifically for segmentation purposes in biomedical. Moreover, we were successful in achieving our result of brain tumor segmentation of two and four classes. As other techniques do not have that, much space of improvement but UNET has space of improvement so went with it while other techniques can do so much so we cannot improve them. The process and the results are mentioned in the previous chapters.

As machine learning and deep learning techniques are becoming more common most of the work will be done by the machines, as they will help minimize the error that we might have and is already become part of our lives giving suggestion of what we might like and want and so much more. We can train models and then different application for mobile phones or computers can be developed to help the patient or doctor to check. Making it easier and more feasible.

However, keep in mind machine is there to help people not to replace them.

REFERENCES

[1] https://aapm.onlinelibrary.wiley.com/doi/pdf/10.1002/mp.13288

[2] https://www.cancer.net/cancer-types/brain-tumor/statistics

[3] N. Noreen, S. Palaniappan, A. Qayyum, I. Ahmad, M. Imran and M. Shoaib, "A Deep Learning Model Based on Concatenation Approach for the Diagnosis of Brain Tumor," in IEEE Access, vol. 8, pp. 55135-55144, 2020, doi: 10.1109/ACCESS.2020.2978629.

[4] K. Hu et al., "Brain Tumor Segmentation Using Multi-Cascaded Convolutional Neural Networks and Conditional Random Field," in IEEE Access, vol. 7, pp. 92615-92629, 2019, doi: 10.1109/ACCESS.2019.2927433.

[5] A. Gumaei, M. M. Hassan, M. R. Hassan, A. Alelaiwi and G. Fortino, "A Hybrid Feature Extraction Method With Regularized Extreme Learning Machine for Brain Tumor Classification," in IEEE Access, vol. 7, pp. 36266-36273, 2019, doi: 10.1109/ACCESS.2019.2904145.

[6] C. Zhou, C. Ding, X. Wang, Z. Lu and D. Tao, "One-Pass Multi-Task Networks With Cross-Task Guided Attention for Brain Tumor Segmentation," in IEEE Transactions on Image Processing, vol. 29, pp. 4516-4529, 2020, doi: 10.1109/TIP.2020.2973510.

[7] K. Pathak, M. Pavthawala, N. Patel, D. Malek, V. Shah and B. Vaidya, "Classification of Brain Tumor Using Convolutional Neural Network," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), 2019, pp. 128-132, doi: 10.1109/ICECA.2019.8821931.

[8] M. S. Islam, N. Kaabouch and W. C. Hu, "A survey of medical imaging techniques used for breast cancer detection," IEEE International Conference on Electro-Information Technology , EIT 2013, 2013, pp. 1-5, doi: 10.1109/EIT.2013.6632694.

[9] M. Aljaleeli, A. Nahar, M. Mahmood and O. Bayat, "Magnetic Resonance Imaging (MRI) For Brain Tumor And Seizures Classification Using Recurrent Neural Network," 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), 2020, pp. 1-7, doi: 10.1109/ISMSIT50672.2020.9254648.

[10] A. Letyagin et al., "Multi-class Brain Tumor Segmentation via Multi-sequences MRI Mixture Data Preprocessing," 2020 Cognitive Sciences, Genomics and Bioinformatics (CSGB), 2020, pp. 185-189, doi: 10.1109/CSGB51356.2020.9214645.

[11] L. Zhou, R. Chav, T. Cresson, G. Chartrand and J. de Guise, "3D knee segmentation based on three MRI sequences from different planes," 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2016, pp. 1042-1045, doi: 10.1109/EMBC.2016.7590881.

[12]https://www.mayoclinic.org/tests-procedures/mri/about/pac-20384768#:~:text=Magnetic%20resonance%20imaging%20(MRI)%20is,large%2C%20tube%2Dshaped%20magnets.

[13]Vishal K. Waghmare, Maheshkumar H. Kolekar. "Chapter 8 Brain Tumor Classification Using Deep Learning" , Springer Science and Business Media LLC, 2021

[14] Vishal K. Waghmare, Maheshkumar H. Kolekar. "Chapter 8 Brain Tumor Classification Using Deep Learning" , Springer Science and Business Media LLC, 2021

[15]] W. McCulloch and W. Pitts, "A logical calculus of ideas immanent in nervous

activity," Bulletin of Mathematical Biophysics, vol. 5, pp. 115–133, 1943

[16] Qi Chang, Leon Axel, Dimitris N. Metaxas. "An overview of artificial intelligence: basics and state-of-the-art algorithms" , Elsevier BV, 2021

[17] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," Psychological Review, vol. 65, no. 6, pp. 386– 408, 1958

[18] H. Robbins and S. Monro, "A stochastic approximation method," Ann. Math. Statist., vol. 22, pp. 400–407, 09 1951.

[19] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function," Ann. Math. Statist., vol. 23, pp. 462–466, 09 1952.

[20] R. S. Sutton, "Two problems with backpropagation and other steepest-descent learning procedures for networks," in Proceedings of the Eighth Annual Conference of the Cognitive Science Society, Hillsdale, NJ: Erlbaum, 1986.

[21] K. J. Somaiya College of Engineering Vidyavihar, Mumbai

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[23] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," Biological Cybernetics, vol. 36, pp. 193–202, 1980

[24] D. Hubel and T. Wiesel, "Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex," Journal of Physiology, vol. 160, pp. 106– 154, 196 Submitted to Indian Institute of Technology Roorkee Student Paper

[25] Guinan, E.M., Lowy, C., Stanhope, N., et al.: 'Cognitive effects of pituitary tumours and their treatments: two case studies and an investigation of 90 patients', J. Neurology Neurosurgery Psychiatry, 1998, 65, (1), pp. 870–876

[26] Padlia, M., Sharma, J.: 'Fractional Sobel filter based brain tumor detection and segmentation using statistical features and svm', in Nath, V., Mandal, J. (Eds): 'Nanoelectronics, circuits and communication systems, lecture notes in electrical engineering' (Springer, Singapore, 2019), pp. 161–175

[27] Havaei, M., Davy, A., Warde-Farley, D., et al.: 'Brain tumor segmentation with deep neural networks', Med. Image Anal., 2017, 35, (1), pp. 18–31

[28] Kharrat, A., Gasmi, K., Messaoud, M.B., et al.: 'A hybrid approach for automatic classification of brain MRI using genetic algorithm and support vector machine', Leonardo J. Sci., 2010, 17, (1), pp. 71–82

[29] Paul, J.S., Plassard, A.J., Landman, B.A., et al.: 'Deep learning for brain tumor classification'. Proc. SPIE Orlando, USA, May 2017

[30] Reddi, S.J., Kale, S., Kumar, S.: 'On the convergence of Adam and Beyond'. Proc. of the 6th Int. Conf. on Learning Representations (ICLR 2018), Vancouver, Canada, May 2018

[31] Qiao, K., Zhang, C., Wang, L., et al.: 'Accurate reconstruction of image stimuli from human fMRI based on the decoding model with capsule network architecture', arXiv:1801.00602, 2018

[32] Yuan, Y., He, C.: 'Variational level set methods for image segmentation based on both l2 and Sobolev gradients', Int. J. Nonlinear Anal. Real World Appl., 2012

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation " (2018)

[34] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks" (2017)

[35] Jonathan Long, Evan Shelhamer, Trevor Darrell UC Berkeley, "Fully Convolutional Networks for Semantic Segmentation" (2018)

[36] Wang, P., et al.: Understanding convolution for semantic segmentation. In: 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Lake Tahoe, NV, USA, 12–15 March 2018, pp. 1451–1460 (2018)

[37] Sengupta Debleena , "Deep Learning Architectures for Automated Image Segmentation" (2015)

[38] Srivastava, N., Salakhutdinov, R.: Multimodal learning with deep boltzmann machines. J. Mach. Learn. Res. 15(1), 2949–2980 (2014)

[39] Zhao, X., Wu, Y., Song, G., Li, Z., Zhang, Y., Fan, Y.: A deep learning model integrating FCNNs and CRFs for brain tumor segmentation. CoRR abs/1702.04528 (2017)

[40] Leon Weninger(B), Oliver Rippel, Simon Koppers, and Dorit Merhof,"Segmentation of Brain Tumors and Patient Survival Prediction: Methods for the BraTS 2018 Challenge" (2016)

[41] Bakas, S., et al.: Segmentation labels and radiomic features for the pre-operative scans of the TCGA-LGG collection. The Cancer Imaging Archive (2017)

[42] Bakas, S., et al.: Segmentation labels and radiomic features for the pre-operative scans of the TCGA-GBM collection. The Cancer Imaging Archive (2017)

[43] Bakas, S., et al.: Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features. Scientific Data 4, 170117 EP, September 2017. https://doi.org/10.1038/sdata.2017.117

[44] Bakas, S., et al.: Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BRATS challenge. arXiv preprint arXiv:1811.02629

[45] C¸i¸cek, ¨O., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-Net: learning dense volumetric segmentation from sparse annotation. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9901, pp. 424–432. Springer, Cham (2016). https://doi.org/10.1007/978-3-31946723-8 49

[46] Isensee, F., Kickingereder, P., Wick, W., Bendszus, M., Maier-Hein, K.H.: Brain tumor segmentation and radiomics survival prediction: contribution to the BRATS 2017 challenge. CoRR abs/1802.10508 (2018). http://arxiv.org/abs/1802.10508

[47] Isensee, F., Kickingereder, P., Wick, W., Bendszus, M., Maier-Hein, K.H.: No newnet. CoRR abs/1809.10483 (2018). http://arxiv.org/abs/1809.10483

[48] Isensee, F., et al.: nnU-Net: self-adapting framework for U-Net-based medical image segmentation. CoRR abs/1809.10486 (2018). http://arxiv.org/abs/1809.10486

[49] Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. CoRR abs/1505.04597 (2015)

[50] Taha, A.A., Hanbury, A.: Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. BMC Med. Imaging 15(1), 29 (2015)

[51] Taherdangkoo, M., Bagheri, M.H., Yazdi, M., Andriole, K.P.: An effective method for segmentation of MR brain images using the ant colony optimization algorithm. J. Digit. Imaging 26(6), 1116–1123 (2013)

[52] Wu, W., Chen, A.Y.C., Zhao, L., Corso, J.J.: Brain tumor detection and segmentation in a CRF (conditional random fields) framework with pixel-pairwise affinity and superpixel-level features. Int. J. of Comp. Assist. Radiol. Surg. 9(2), 241–253 (2014)

Comparison Table References:

Menze B, et al. The Multimodal brain tumor image segmentation benchmark (brats). IEEE Trans Med Imaging 2015; 34(10):1993-2024.

Pereira S, Pinto A, Alves V, Silva C A. Brain tumor segmentation using convolutional neural networks in MRI images. IEEE Trans Med Imaging 2016;35(5):1240–1251.

D. Kwon et al. Combining generative models for multifocal glioma segmentation and registration. Medical Image Computing and Computer-Assisted Intervention–MICCAI 2014. Springer, 2014:763–770.

Havaei M, Davy A, Farley W D, Biard A, Courville A, Bengio Y, Pal C, Jadoin P M, Larochelle H. Brain tumor segmentation with deep neural networks. Medical Image Analysis 2016, doi: 10.1016/j.media.2016.05.004.

Tustison N. et al. Optimal symmetric multimodal templates and concatenated random forests for supervised brain tumor segmentation (simplified) with antsr. Neuroinformatics 2015;13(2): 209–225.

Urban G. et al. Multi-modal brain tumor segmentation using deep convolutional neural networks. MICCAI Multimodal Brain Tumor Segmentation Challenge (BraTS) 2014:31–35.

Havaei M, Larochelle H, Poulin P, Jadoin P M. Within-brain classification for brain tumor segmentation. Int J Cars 2016; 11:777-788.

Dvorak P, Menze B. Structured prediction with convolutional neural networks for multimodal brain tumor segmentation. MICCAI Multimodal Brain Tumor Segmentation Challenge (BraTS) 2015:13–24.

Davy A. et al. Brain tumor segmentation with deep neural networks. MICCAI Multimodal Brain Tumor Segmentation Challenge (BraTS) 2014:1–5.

Zikic D. et al. Segmentation of brain tumor tissues with convolutional neural networks. MICCAI Multimodal Brain Tumor Segmentation Challenge (BraTS) 2014:36–39

Hamamci A, et al. Tumor-Cut: segmentation of brain tumors on contrast enhanced MR images for radiosurgery applications. IEEE Trans Med Imaging 2012; 31(3):790–804