



**Ain Shams University**  
**Faculty of Computer & Information Sciences**  
**Computer Science Department**

# Diagnosing Cancer Using Deep-Learning

**August 2020**



**Ain Shams University**  
**Faculty of Computer & Information Sciences**  
**Computer Science Department**

# Diagnosing Cancer Using Deep-Learning

**By:**

Islam Mohamed Zakaria [CS]  
Islam Hesham Hussein [CS]  
Islam Yehia Abd El-Aziz [CS]  
El-Bara Mohamed Mahmoud [CS]  
Ahmed Magdy Ali [CS]

## **Under Supervision of:**

Associate Professor. Manal Mohsen Tantawi,  
Scientific Computing Department,  
Faculty of Computer and Information Sciences,  
Ain Shams University.

TA. Hader El-Saadawy,  
Scientific Computing Department,  
Faculty of Computer and Information Sciences,  
Ain Shams University.

# Acknowledgement

We would like to thank our supervisor, Prof. Manal Mohsen Tantawi, for guiding us to explore the ramification of our work and we express our sincere gratitude towards her for leading us through the completion of Project.

We would also like to thank T.A. Hadeer Elsaadawy, who helped us in our supervisor's absence. your suggestions and the constant guidance in every stage of the project helped us a lot.

Finally, we would like to thank our family members for their moral support and encouragement to achieve goals.

## **Abstract**

Recently deep learning has been playing a major role in the field of computer vision. One of its applications is the reduction of human judgment in the diagnosis of diseases. Especially, brain tumor diagnosis requires high accuracy, where minute errors in judgment may lead to disaster. For this reason, we are using a deep learning model to reduce the diagnosing error.

Here we present a solution for brain tumor classifying by using deep learning. In this work, we studied three types of brain tumors (i.e., meningioma, glioma, and pituitary tumor) in T1-weighted contrast-enhanced MRI (CE-MRI) images and applied different models for classification.

In this paper, we propose a method to enhance the classification performance. we classify the angle of the input MR image (i.e., axial, sagittal, and coronal), classify The MR image angle (i.e., normal, meningioma, glioma, and pituitary tumor).

We evaluate the efficacy of the proposed method on a dataset we collected from multiple online sources for abnormal MR images and the normal dataset is acquired from hospitals. This dataset contains 12 class. We use LINK-NET, VOLUME-NET, RES-NET, VGG, and SVM. The best accuracy we achieved is 97% from RES-NET.

# Table of Contents

Acknowledgement .....	1
Abstract.....	2
List of Figures .....	5
List Of table .....	6
List of Abbreviations .....	7
1- Introduction .....	9
1.1 Motivation.....	9
1.2 Problem Definition .....	10
1.3 Objective .....	11
.....	11
1.4 Time Plan.....	12
1.5 Document Organization .....	13
2- Background .....	15
2.1 Intro to Deep Learning .....	16
2.2 Convolution Neural Network .....	22
3- Related Work .....	24
3.1 Survey.....	24
3.2 Survey Analysis.....	33
4- Analysis and Design.....	34
4.1 System Overview.....	34
4.1.1 System Architecture.....	34
4.1.2 System Users .....	34
4.2 System Analysis & Design .....	35
4.2.1 Use Case Diagram .....	35
5- Implementation and Testing.....	36
5.1 Dataset .....	36
5.1.1 Abnormal .....	36
5.1.2 Abnormal Coronal .....	36

5.1.3 Abnormal Axial .....	37
5.1.4 Abnormal Sagittal.....	37
5.1.5 Normal .....	38
5.1.6 MATLAB work in image .....	39
5.2 Models .....	40
5.2.1 VGG-16.....	40
5.2.2 VGG-19.....	42
5.2.3 Res-Net 50.....	46
5.2.4 SVM .....	47
5.3 Experimental Results for All Architecture .....	49
5.3.1 Optimizers .....	49
5.3.2 Activation Function .....	50
5.3.3 Batch Size .....	50
5.3.4 Number of Epochs.....	51
5.3.5 Loss Function.....	51
5.3.6 Learning Rate .....	52
5.3.7 Table of Accuracy .....	52
6- User Manual.....	54
7- Conclusion and Future Work .....	57
7.1 Conclusion.....	57
7.2 Future Work.....	58
8- References .....	59

## List of Figures

Figure 1.1 Example of Brain Tumor .....	11
Figure 1.2 Project Time Plan .....	12
Figure 2.1 Neuron Architecture .....	16
Figure 2.2 Artificial Intelligence Hierarchy.....	17
Figure 2.3 Example for Fully Connected Layers.....	21
Figure 2.4 CNN Architecture.....	22
Figure 4.1System Architecture .....	34
Figure 4.2 Use Case Diagram .....	35
Figure 5.1 Abnormal Coronal Example.....	36
Figure 5.2 Abnormal Axial Example.....	37
Figure 5.3 Abnormal Sagittal Example .....	37
Figure 5.4 Normal Brain MR Images Example .....	38
Figure 5.5 VGG-16 Architecture .....	40
Figure 5.6 VGG-19 Architecture .....	42
Figure 5.7 ResNet-50 Architecture .....	46
Figure 5.8 SVM Architecture .....	48
Figure 5.9 Res-Net Model Accuracy .....	52
Figure 5.10 Overall Accuracy .....	53
Figure 5.11 Training Accuracy .....	53

## List Of table

Table 3-1 Survey Table1 .....	24
Table 3-2 Survey Table2.....	26
Table 3-3 Survey Table3.....	28
Table 3-4 Survey Table4.....	30
Table 3-5 Survey Table5.....	32
Table 5-1 Accuracy Table.....	52



## List of Abbreviations

ANN	Artificial Neural Network
AWS	Amazon Web Services
BPNN	Back Propagation Network
CAD	Computer-Aided Design
CNN	Convolution Neural Network
CNS	Central Nervous System
CT	Computed Tomography
DNN	Deep Neural Networks
ELM	Extreme Learning Machine

FCNN	Fourier Convolutional Neural Networks
GLCM	Gray Level Co-occurrence Matrix
KNN	K Nearest Neighbor
MLP	Multi-Layer Perceptron
MRI	Magnetic Resonance Imaging
NN	Neural Networks
Res-Net	Residual Neural Network
SRC	Short Range Correlated
SVM	Support Vector Machine
UI	User Interface
VGG	Visual Geometry Group

# 1- Introduction

## 1.1 Motivation

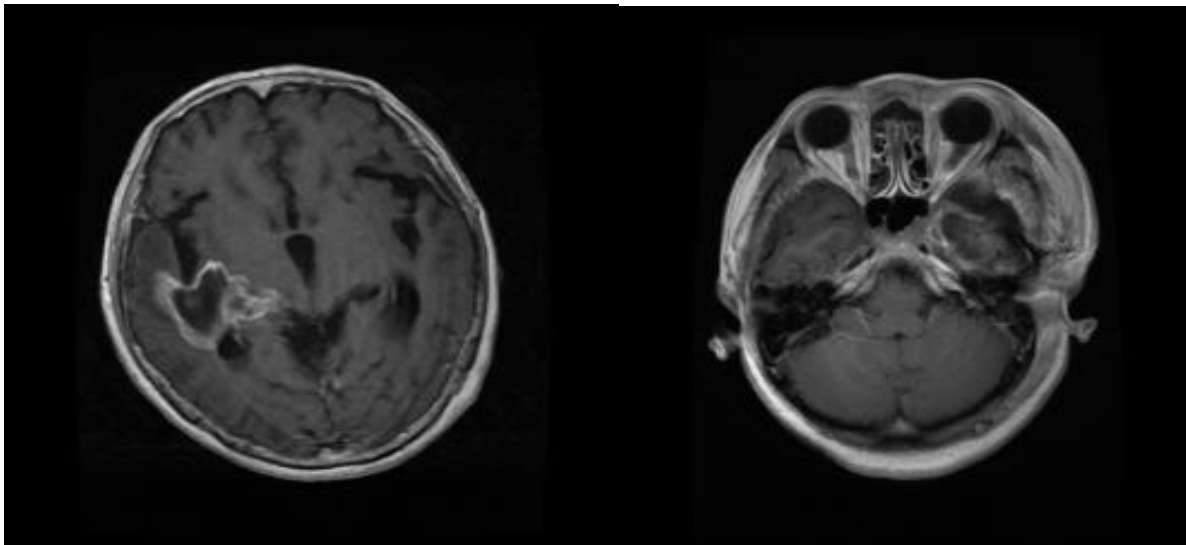
- Brain tumor detection is an important task in medical image processing. Early diagnosis of brain tumors plays an important role in improving treatment possibilities and increases the survival rate of the patients.
- Due to the level of complexity in both the Human brain and the MRI Scanners there are many issues that may occur that increases the misdiagnoses of tumor.
- Misdiagnoses occur due to technical reasons related to image quality resulting in human error.
- (CAD) systems are developed to overcome these restrictions as it improves radiologists' performance in discriminating between normal and abnormal tissues.
- Brain tumors are about four times as common as primary brain tumors, with about half of metastases coming from lung cancer. Primary brain tumors occur in around 250,000 people a year globally, making up less than 2% of cancers. In children younger than 15, brain tumors are second only to acute lymphoblastic leukemia as the most common form of cancer. In Australia, the average lifetime economic cost of a case of brain cancer is \$1.9 million, the greatest of any type of cancer.

## 1.2 Problem Definition

- In the past there was no way to diagnose tumors resulting in many deaths.
- As technology advances, X-rays have emerged, allowing scientists to speculate the presence of the tumor and act accordingly.
- Neuroimaging: In the late 1940s, which outlined tumors at the time of surgery, and then tagged the fluorescein with radioactive iodine, which allowed visualization of the tumors before surgery
- Until the emergence of MRI, which in turn led to the development of methods of treatment of patients and contribute to the medical field through computer-aided diagnosis (CAD).
- Magnetic resonance imaging is a medical imaging technique that uses a magnetic field and computer-generated radio waves to create detailed images of the organs and tissues in your body.
- MRI is the most frequently used imaging test of the brain often performed to help diagnose tumors, stroke, and brain injury from trauma.

### 1.3 Objective

- The Project aims to Implement the techniques of Deep learning for diagnosis tumors.
- Deep learning implementation aims to provide an accurate diagnosis for brain tumor by processing (MRI) Scans
- The Project is designed to be scalable and to be used for diagnosis of other malignant tumors.



*Figure 0.1 Example of Brain Tumor*

## 1.4 Time Plan

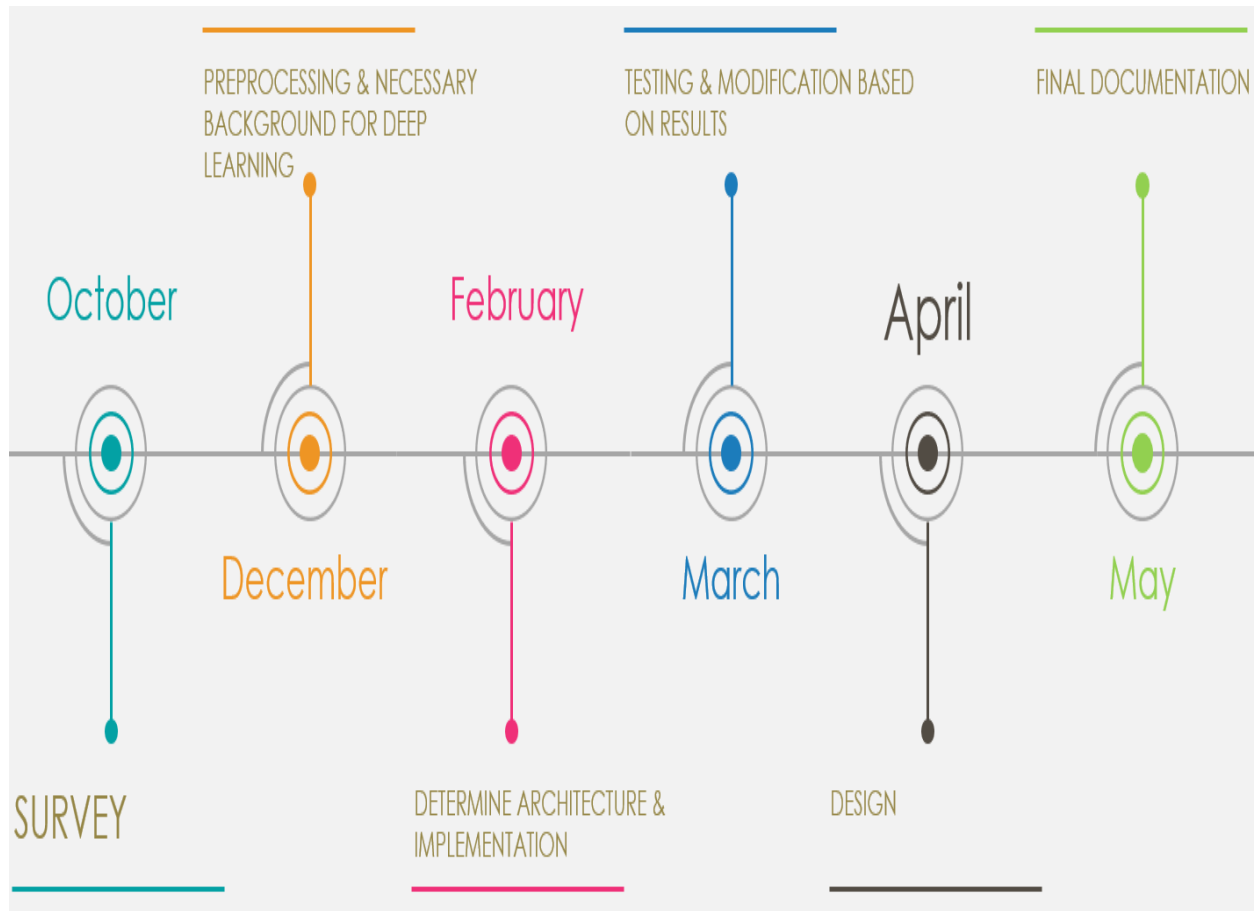


Figure 0.2 Project Time Plan

## 1.5 Document Organization

Here we would include a paragraph for each chapter describing what was discussed in this chapter.

- **Background:**

This chapter would give a detailed description of the project, all the scientific background related to our project, table the show an explanatory survey of the work done in diagnose brain cancer using deep learning, and finally a description of any existing similar systems.

- **Analysis and Design:**

This chapter would contain the following content

- **System Overview:**

we would give a detailed description of system architecture, and system users.

- **System Analysis and Design:**

we would include the use case diagram, class diagram, and sequence diagram.

- **Implementation and Testing**

This chapter would give a detailed description of all the functions in our system, a detailed description of all implemented techniques and algorithms, a description of any new technologies used in the implementation, UI design, and testing procedures and levels used.

- **User Manual**

This chapter would give a detailed description of how to operate the project along with screenshots of the project representing all steps, and also there would be an "Installation Guide" that would describe how to install the program, and all required third-party tools that need to be available for the project to run.

- **Conclusion and Future Work**

In this chapter, there would be a summary of the whole project along with the results obtained, and also our thoughts of what can be done in the future to improve the performance of the project and what additional functions could be added.



## 2- Background

In general, diagnosing a brain tumor usually begins with magnetic resonance imaging (MRI). Once MRI shows that there is a tumor in the brain, the most common way to determine the type of brain tumor is to look at the results from a sample of tissue after a biopsy or surgery. These tests and procedures are described below in more detail.

The brain is the body organ composed of nerve cells and supportive tissues like glial cells and meninges – there are three major parts – they control your activity like breathing (brain stem), an activity like moving muscles to walk (cerebellum) and your senses like sight and our memory, emotions, thinking and personality (cerebrum). Primary brain tumors can be either malignant (contain cancer cells) or benign (do not contain cancer cells). A primary brain tumor is a tumor that begins in the brain tissue. If a cancerous tumor starts elsewhere in the body, it can spread cancer cells, which grow in the brain. These types of tumors are called secondary or metastatic brain tumors.

An MRI uses magnetic fields, not x-rays, to produce detailed images of the body. MRI can be used to measure the tumor's size. A special dye called a contrast medium is given before the scan to create a clearer picture. This dye can be injected into a patient's vein or given as a pill or liquid to swallow.

MRIs create more detailed pictures than CT scans (see below) and are the preferred way to diagnose a brain tumor.

The MRI may be of the brain, spinal cord, or both, depending on the type of tumor suspected and the likelihood that it will spread in the CNS.

Glioma is a tumor occurs in the lowest part of the brain. It can be a low-grade or high-grade tumor.

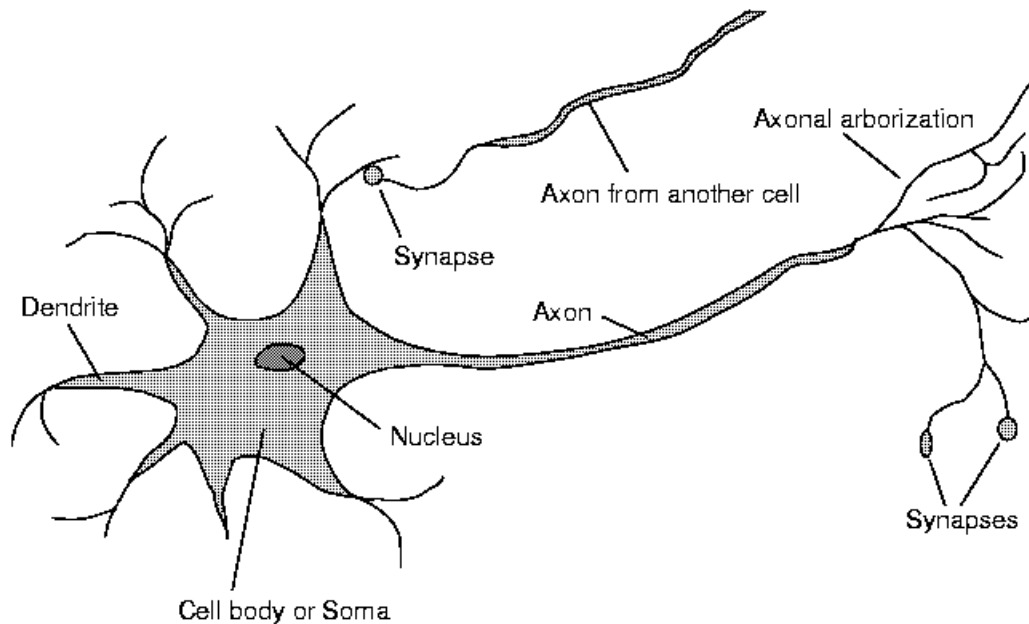
The most common type is a diffuse intrinsic pontine glioma. A meningioma can be difficult to diagnose because the tumor is often slow growing.

Symptoms of a meningioma may also be subtle and mistaken for other health conditions or written off as normal signs of aging.

The pituitary is a small gland attached to the base of the brain (behind the nose) in an area called the pituitary fossa or Sella turcica. The pituitary is often called the "master gland" because it controls the secretion of most of the hormones in the body.

A normal pituitary gland weighs less than 1 gram and is about the size and shape of a kidney bean.

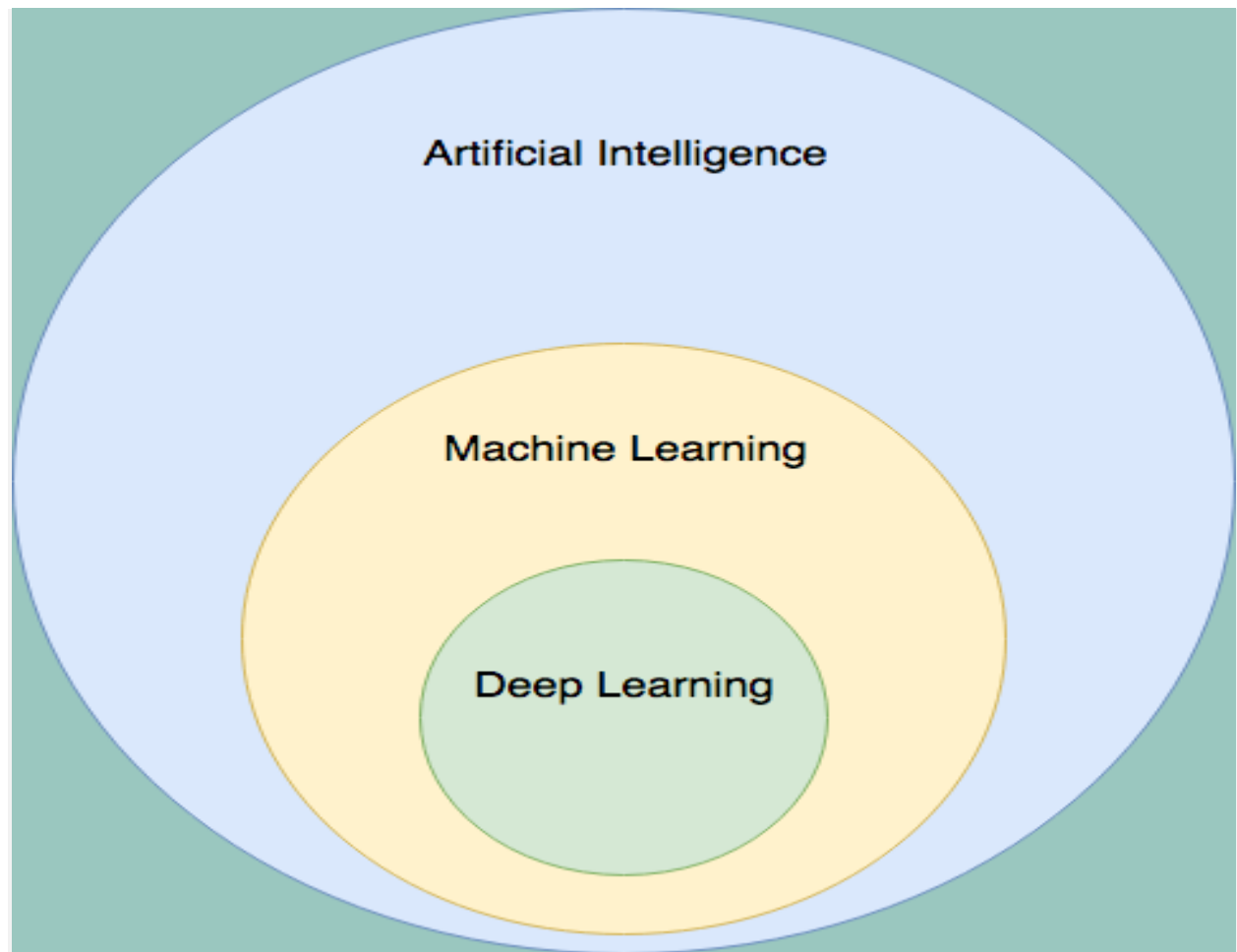
## 2.1 Intro to Deep Learning



*Figure 2.1 Neuron Architecture*

- A neuron contains a cell body for signal processing,
- many dendrites to receive signals (Inputs receives through dendrite)
- an axon for outputting the result; and
- a synapse between the axon and each dendrite

Deep learning is a subset of machine learning. Machine learning is a subset of artificial intelligence. Said another way — all deep learning algorithms are machine learning algorithms, but many machine learning algorithms do not use deep learning. As a Venn Diagram, it looks like this:



*Figure 2.2 Artificial Intelligence Hierarchy*

Deep learning refers specifically to a class of algorithm called a neural network, and technically only to “deep” neural networks (more on that in a second). This first neural network was invented in 1949, but back then they weren’t very useful. In fact, from the 1970’s to the 2010’s traditional forms of AI would consistently outperform neural network-based models.

These non-learning types of AI include rule based algorithms (imagine an extremely complex series of if/else blocks); heuristic based AIs such as A\* search; constraint satisfaction algorithms like Arc Consistency; tree search

algorithms such as minimax (used by the famous Deep Blue chess AI); and more.

There were two things preventing machine learning, and especially deep learning, from being successful. Lack of availability of large datasets and lack of availability of computational power. In 2018 we have exabytes of data, and anyone with an AWS account and a credit card has access to a distributed supercomputer. Because of the new availability of data and computing power, Machine learning — and especially deep learning — has taken the AI world by storm.

You should know that there are other categories of machine learning such as unsupervised learning and reinforcement learning but for the rest of this article, I will be talking about a subset of machine learning called supervised learning.

Supervised learning algorithms work by forcing the machine to repeatedly make predictions. Specifically, we ask it to make predictions about data that we (the humans) already know the correct answer for. This is called “labeled data” — the label is whatever we want the machine to predict.

Here is an example let us say we wanted to build an algorithm to predict if someone will default on their mortgage. We would need a bunch of examples of people who did and did not default on their mortgages. We will take the relevant data about these people; feed them into the machine learning algorithm; ask it to make a prediction about each person; and after it guesses we tell the machine what the right answer actually was. Based

on how right or wrong it was the machine learning algorithm *changes how it makes predictions*.

We repeat this process *many* times, and through the miracle of mathematics, our machine's predictions get better. The predictions get better relatively slowly though, which is why we need so much data to train these algorithms.

Machine learning algorithms such as linear regression, support vector machines, and decision trees all “learn” in different ways, but fundamentally they all apply this same process: make a prediction, receive a correction, and adjust the prediction mechanism based on the correction. At a high level, it is quite similar to how a human learns.

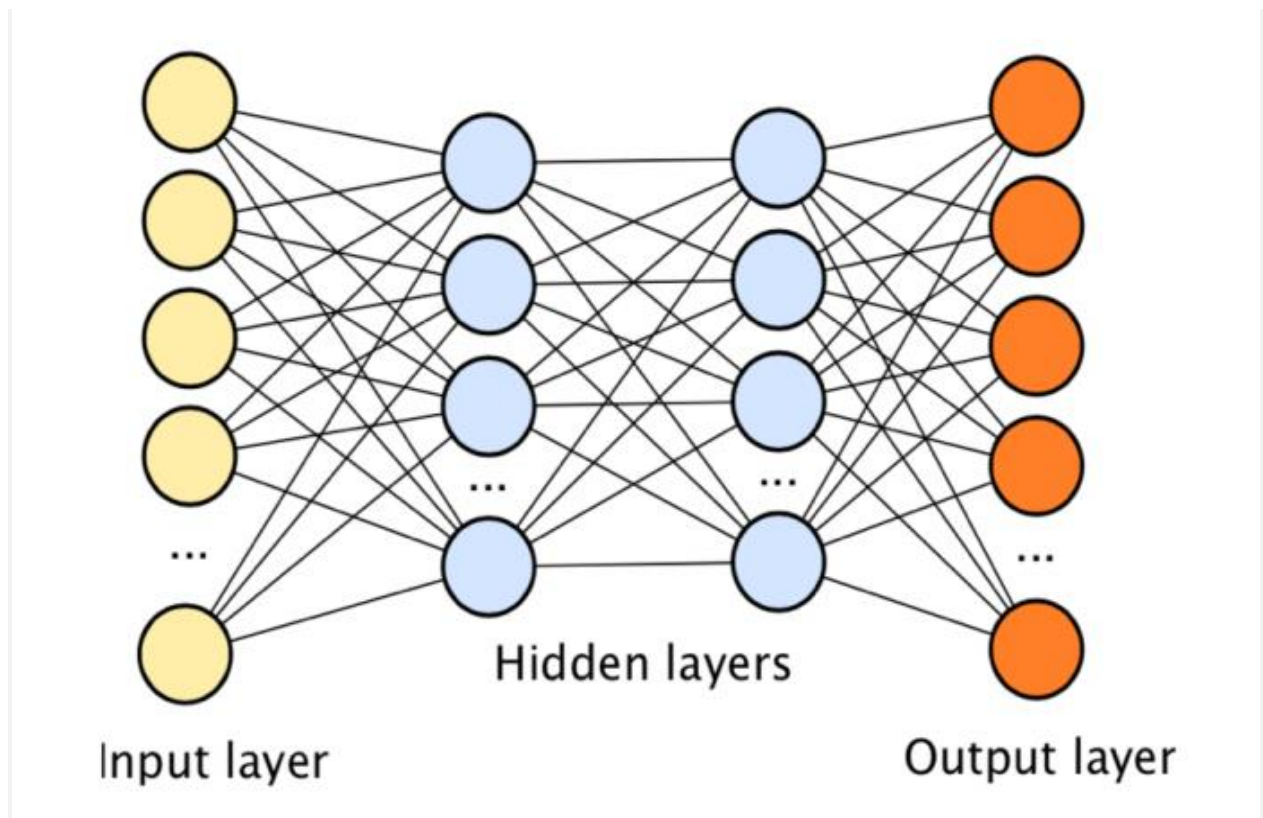
Recall that deep learning is a subset of machine learning which focuses on a specific category of machine learning algorithms called neural networks. Neural networks were originally inspired by the way human brains work — individual “neurons” receive “signals” from other neurons and in turn send “signals” to other “neurons”. Each neuron transforms the incoming “signals” in some way, and eventually an output signal is produced. If everything went well that signal represents a correct prediction!

This is a helpful mental model, but computers are not biological brains. They do not have neurons, or synapses, or any of the other biological mechanisms that make brains work. Because the biological model breaks down, researchers and scientists instead use graph theory to model neural

networks — instead of describing neural networks as “artificial brains”, they describe them as complex graphs with powerful properties.

Viewed through the lens of graph theory a neural network is a series of layers of connected nodes; each node represents a “neuron” and each connection represents a “synapse”.

Different kinds of nets have different kinds of connections. The simplest form of deep learning is a deep neural network. A deep neural network is a graph with a series of fully connected layers. Every node in a layer has an edge to every node in the next layer; each of these edges is given a different weight. The whole series of layers is the “brain”. It turns out, if the weights on all these edges are set *just right* these graphs can do some incredible “thinking”.

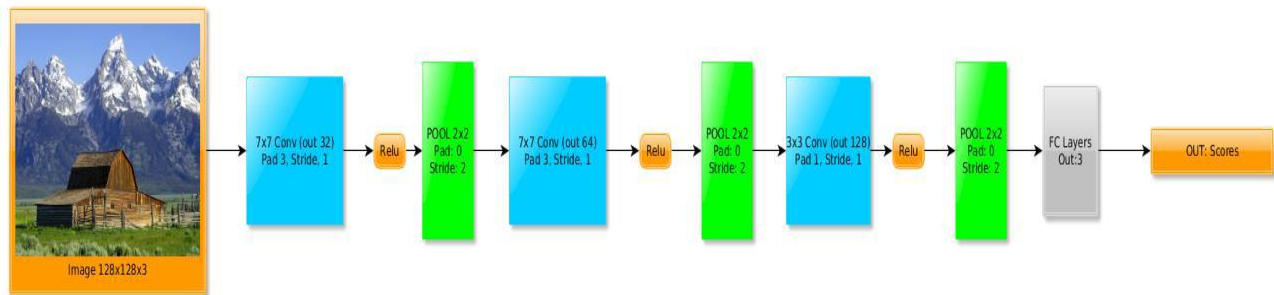


*Figure 2.3 Example for Fully Connected Layers*

Ultimately, the Deep Learning Course will be about how to construct different versions of these graphs; tune the connection weights until the system works; and try to make sure our machine does what we *think* it is doing. The mechanics that make Deep Learning work, such as gradient descent and backpropagation, combine a lot of ideas from different mathematical disciplines. In order to really understand neural networks, we need some math background.

## 2.2 Convolution Neural Network

A CNN is composed of layers that filters(convolve) the inputs to get useful information.



*Figure 2.4 CNN Architecture*

These have two kinds of layers: convolution layers and pooling layers

convolutional layer contains a set of filters whose parameters need to be learned. The height and weight of the filters are smaller than those of the input volume. Each filter is convolved with the input volume to compute an activation map made of neurons. In other words, the filter is slid across the width and height of the input and the dot products between the input and filter are computed at every spatial position. The output volume of the convolutional layer is obtained by stacking the activation maps of all filters along the depth dimension. Since the width and height of each filter is designed to be smaller than the input, each neuron in the activation map is only connected to a small local region of the input volume. In other words, the receptive field size of each neuron is small, and is equal to the filter size. The local connectivity is motivated by the architecture of the animal visual cortex where the receptive fields of the cells are small. The local connectivity of the convolutional layer allows the network to learn filters which maximally respond to a local region of the input, thus exploiting the spatial local correlation of the input (for an input image, a pixel is more correlated to the nearby pixels than to the distant pixels). In addition, as the activation map is obtained by performing convolution between the filter and the input, the filter parameters are shared for all local positions. The weight sharing reduces the number of parameters for efficiency of expression, efficiency of learning, and good generalization pooling layer is usually incorporated between two successive convolutional layers.



The pooling layer reduces the number of parameters and computation by down sampling the representation. The pooling function can be max or average. Max pooling is commonly used as it works better

Fully connected layers are an essential component of Convolutional Neural Networks (CNNs), which have been proven very successful in recognizing and classifying images for computer vision. The CNN process begins with convolution and pooling, breaking down the image into features, and analyzing them independently. The result of this process feeds into a fully connected neural network structure that drives the final classification decision.

## 3- Related Work

### 3.1 Survey

We have made a survey before work to get the best ways for beginning in our project and in the following table we would show what we have got.

*Table 3-1 Survey Table1*

References	Dataset	Preprocessing	Features & Classifiers	Accuracy
<ul style="list-style-type: none"> <li>Isin et al. [1] (2016)</li> </ul>	BRATS 2013 <ul style="list-style-type: none"> <li>274 mri scan of patients with gliomas (both high and low grades)</li> <li>110 scans are available with unknown grades and unknown ground truths for testing.</li> </ul>	<ul style="list-style-type: none"> <li>Noise removal</li> <li>Skull-stripping</li> <li>Intensity base correction</li> </ul>	Features: - <ul style="list-style-type: none"> <li>Discrete wavelet transforms (DWT)</li> <li>Textons</li> <li>Multifractal Brownian motion</li> <li>First order statistical</li> <li>Raw intensities</li> <li>Local image textures</li> <li>Intensity gradients</li> <li>Edge based</li> </ul> Classifiers: - <ul style="list-style-type: none"> <li>NN</li> <li>SVM</li> <li>AdaBoost</li> <li>KNN</li> <li>SOM</li> <li>RFs</li> </ul>	67%

			<ul style="list-style-type: none"> <li>• CRF</li> <li>• CC</li> </ul>	
<ul style="list-style-type: none"> <li>• Zacharaki et al. [2] (2017)</li> </ul>	<ul style="list-style-type: none"> <li>• 98 patients (52 women - 46 men) age (17 - 83) 4 had multiple tumors</li> </ul>	<ul style="list-style-type: none"> <li>○ Noise Reduction</li> <li>○ Inhomogeneity correction</li> <li>○ rigid intra subject registration using the public software (FSL)</li> </ul>	Features: - <ul style="list-style-type: none"> <li>○ forward selection method</li> <li>○ subset selection method</li> </ul> Classifiers: - <ul style="list-style-type: none"> <li>• SVM</li> <li>• RFs</li> <li>• Leave-one-out</li> </ul>	98.2%
Sudha et al. [3] (2014)	<ul style="list-style-type: none"> <li>• 42 (25 abnormal 17 normal) testing data 30 (12 abnormal)</li> </ul>	<ul style="list-style-type: none"> <li>○ Noise removal</li> <li>○ Intensity base correction</li> </ul>	Features: - <ul style="list-style-type: none"> <li>○ FFNN</li> <li>○ MLP</li> <li>○ BPN</li> </ul> Classifiers: - <ul style="list-style-type: none"> <li>• GLCM &amp; GLRM AND Fuzzy Entropy Measure</li> </ul>	96%

Table 3-2 Survey Table2

References	Dataset	Preprocessing	Features & Classifiers	Accuracy
Roy et al. [4] (2019)	<ul style="list-style-type: none"> <li>Whole Brain Atlas: MR brain image</li> <li>The EASI MRI Home. MR brain image</li> </ul>	<ul style="list-style-type: none"> <li>Dimension construction method</li> <li>Transform MRI into grayscale using weighted summation of R, G, B components.</li> </ul>	Classifiers: - <ul style="list-style-type: none"> <li>K-means</li> </ul>	94.4%
<ul style="list-style-type: none"> <li>Akkus et al. [5] (2017)</li> </ul>	BRATS 2016 <ul style="list-style-type: none"> <li>220 subjects with high grade and 54 subjects with low-grade for training.</li> <li>53 subjects with mixed grades for testing</li> </ul>	<ul style="list-style-type: none"> <li>Registration</li> <li>skull-stripping</li> <li>intensity base correction</li> <li>Intensity Normalization</li> <li>noise removal</li> </ul>	Classifiers: - <ul style="list-style-type: none"> <li>DSC</li> <li>CNN</li> <li>SVM</li> <li>RFs</li> <li>CSF</li> <li>GM</li> <li>WM</li> </ul>	90%

<ul style="list-style-type: none"> <li>Cheng et al. [6] (2015)</li> </ul>	<p>Brain tumor dataset</p> <ul style="list-style-type: none"> <li>233 patients with 3064 images (708 meningioma's – 1426 gliomas – 930 pituitary tumor).</li> </ul>	<ul style="list-style-type: none"> <li>Intensity normalization.</li> </ul>	<p>Features: -</p> <ul style="list-style-type: none"> <li>intensity histogram</li> </ul> <p>Classifiers: -</p> <ul style="list-style-type: none"> <li>BoW</li> <li>GLCM</li> <li>SVM</li> <li>SRC</li> <li>(1-3-715-45) NN</li> </ul>	91.14%
<ul style="list-style-type: none"> <li>Thillaikkarasi et al. [7] (2018)</li> </ul>	<ul style="list-style-type: none"> <li>Small dataset</li> </ul>	<ul style="list-style-type: none"> <li>Smoothing using LoG</li> <li>CLAHE</li> </ul>	<p>Features: -</p> <ul style="list-style-type: none"> <li>SGLDM</li> </ul> <p>Classifiers: -</p> <ul style="list-style-type: none"> <li>SVM</li> </ul>	84%

Table 3-3 Survey Table3

References	Dataset	Preprocessing	Features & Classifiers	Accuracy
<ul style="list-style-type: none"> <li>Zhao et al. [8] (2016)</li> </ul>	BRATS 2013 <ul style="list-style-type: none"> <li>35 patients.</li> </ul>	<ul style="list-style-type: none"> <li>intensity base correction</li> <li>Intensity Normalization</li> </ul>	Classifiers: - <ul style="list-style-type: none"> <li>FCNN</li> <li>CRF</li> </ul>	82%
<ul style="list-style-type: none"> <li>Gubta et al. [9] (2015)</li> </ul>	<ul style="list-style-type: none"> <li>78 MRI brain tumor</li> </ul>	<ul style="list-style-type: none"> <li>Morphological Opening</li> <li>Noise Removal</li> <li>gradient magnitude</li> </ul>	Classifiers: - <ul style="list-style-type: none"> <li>CNN</li> <li>HSV</li> <li>SVM</li> <li>Naive Bayes</li> </ul>	91.49%
<ul style="list-style-type: none"> <li>Kaur et al. [10] (2014)</li> </ul>	<ul style="list-style-type: none"> <li>70 MRI from local govt. hospital</li> </ul>	<ul style="list-style-type: none"> <li>image resizing</li> <li>Noise removal</li> <li>morphological operations like dilation and erosion</li> </ul>	Classifiers: - <ul style="list-style-type: none"> <li>Thresholding based methods</li> <li>Region growing based methods</li> <li>Neural network methods</li> <li>Fuzzy methods</li> <li>FVT</li> </ul>	99.59%
<ul style="list-style-type: none"> <li>Seetha et al. [11] (2018)</li> </ul>	BRATS 2015 <ul style="list-style-type: none"> <li>80 MRI brain tumor</li> </ul>	<ul style="list-style-type: none"> <li>intensity base correction</li> <li>Intensity Normalization</li> </ul>	Classifiers: - <ul style="list-style-type: none"> <li>NN</li> <li>SVM</li> <li>KNN</li> <li>FCM</li> <li>DWT</li> </ul>	97.5%

			<ul style="list-style-type: none"> <li>• DNN</li> <li>• LDA</li> <li>• SMO</li> </ul>	
<ul style="list-style-type: none"> <li>▪ Kalapala et al. [12] (2015)</li> </ul>	<ul style="list-style-type: none"> <li>▪ 1100 image slices (833 abnormal slices and 267 normal slices) of 150 abnormal patient volumes were considered in this work.</li> </ul>	<ul style="list-style-type: none"> <li>○ Normalization</li> </ul>	Classifiers: - <ul style="list-style-type: none"> <li>• SVM</li> <li>• LS</li> </ul>	98.92%

Table 3-4 Survey Table4

References	Dataset	Preprocessing	Deep-Learning Models	Accuracy
<ul style="list-style-type: none"> <li>▪ Ari et al. [13] (2018)</li> </ul>	<ul style="list-style-type: none"> <li>▪ 16 patients: 9 patients used for training and 7 patients used for testing</li> </ul>	<ul style="list-style-type: none"> <li>○ Noise removal</li> </ul>	<ul style="list-style-type: none"> <li>○ CNN</li> <li>○ ELM</li> <li>○ ELM-LRF</li> <li>○ AlexNet</li> </ul>	97.18%
<ul style="list-style-type: none"> <li>• Banerjee et al. [14] (2018)</li> </ul>	<ul style="list-style-type: none"> <li>• 285 patients: 210 hgg and 75 lgg</li> </ul>	<ul style="list-style-type: none"> <li>○ Noise Reduction</li> </ul>	<ul style="list-style-type: none"> <li>○ PatchNet</li> <li>○ SliceNet</li> <li>○ VolumeNet</li> <li>○ VggNet</li> <li>○ ResNet</li> <li>○ Anfc-IH</li> <li>○ NB</li> <li>○ LR</li> <li>○ MIP</li> <li>○ SVM</li> <li>○ CART</li> <li>○ KNN</li> </ul>	97.18%
Sobhaninia et al. [15] (2018)	<ul style="list-style-type: none"> <li>▪ 233 patients with 3064 images (708 meningioma's –</li> </ul>	<ul style="list-style-type: none"> <li>○ Intensity normalization.</li> </ul>	<ul style="list-style-type: none"> <li>○ LinkNet</li> </ul>	79%



	1426 gliomas – 930 pituitary tumor). 900 for training and 200 for testing.			
<ul style="list-style-type: none"> <li>Lin et al. [16] (2016)</li> </ul>	BRATS 2015 <ul style="list-style-type: none"> <li>(213) patient 18 has been removed because of incomplete or inappropriate feature disclosure</li> </ul>	<ul style="list-style-type: none"> <li>Resize to 60*60</li> <li>Intensity normalization.</li> </ul>	<ul style="list-style-type: none"> <li>LeNet-5</li> </ul>	(74 – 85) %

Table 3-5 Survey Table5

References	Dataset	Preprocessing	Deep-Learning Models	Accuracy
▪ Iqbal et al. [17] (2017)	▪ 36 patients	○ Noise removal	○ SVM ○ FFNN ○ ELM ○ EC	91.17%
• Zhang et al. [18] (2017)	• 274 patients: 220 hgg and 54 lgg	○ Noise Reduction	○ VGG ○ U-Net ○ ResNet	87%
▪ Soltaninejad et al. [19] (2018)	BRATS 2017 ▪ 46 patient 220 hgg and 75 lgg	○ Intensity normalization.	○ VGG	86%

### 3.2 Survey Analysis

Most of these papers was using small dataset and on abnormal MR images only. We decided to collect dataset for normal and abnormal MR images. During work on these datasets we used Noise removal, resizing, and intensity normalization as pre-processing. Finally, ResNet is the architecture with the best results.

## 4- Analysis and Design

### 4.1 System Overview

#### 4.1.1 System Architecture

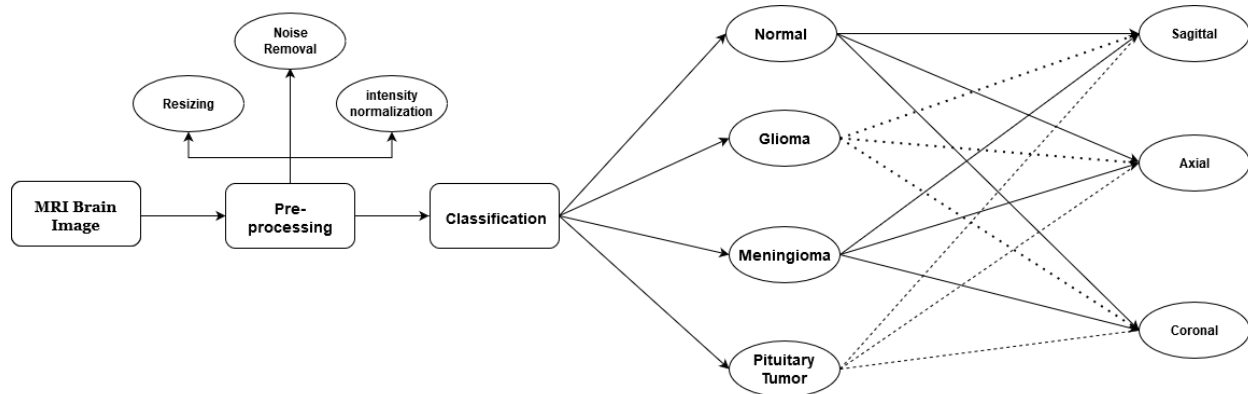


Figure 4.1 System Architecture

At classification we used Residual Network (ResNet) which is a Convolutional Neural Network (CNN) architecture which was designed to enable hundreds or thousands of convolutional layers. While previous CNN architectures had a drop off in the effectiveness of additional layers, ResNet can add many layers with strong performance.

#### 4.1.2 System Users

##### A. Intended Users:

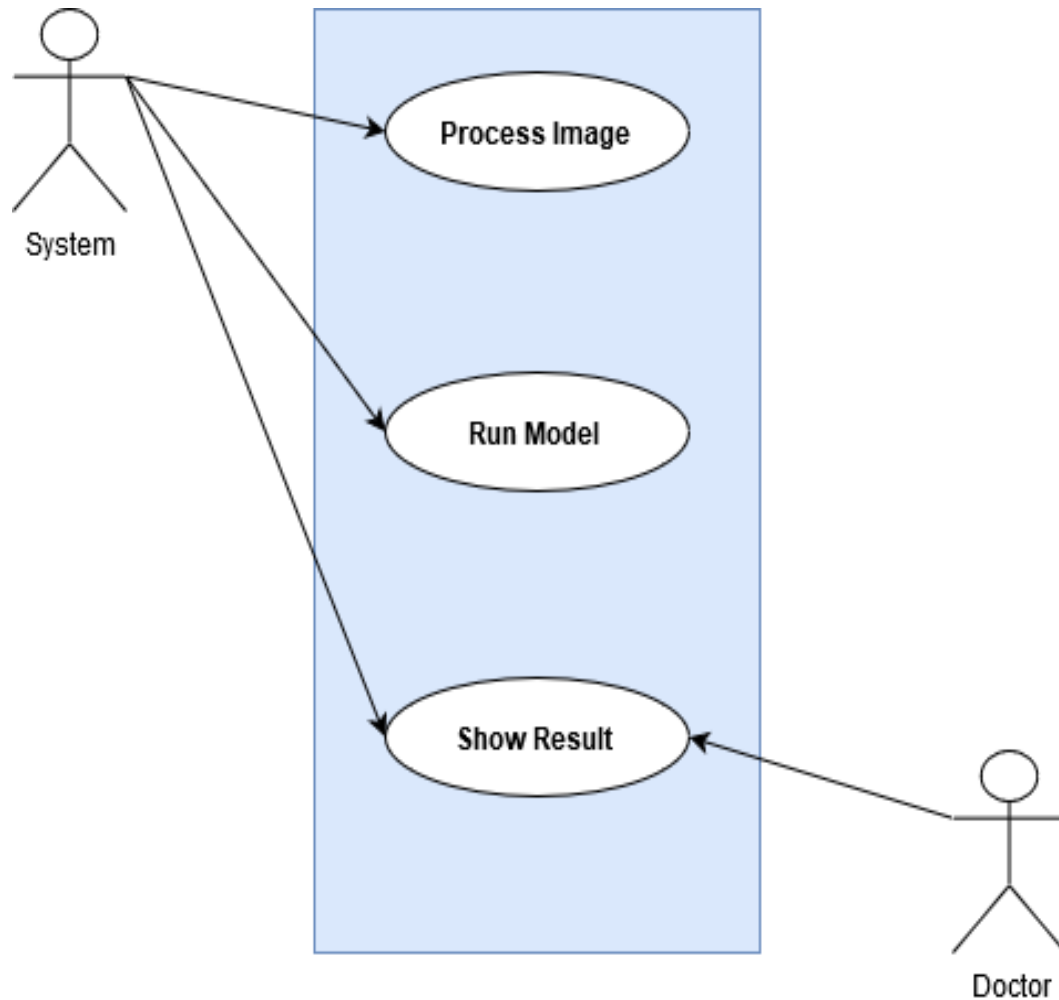
Doctors are working to learn more about brain tumors, ways to prevent them, how to best treat them, and how to provide the best care to people diagnosed with a brain tumor. Unfortunately, that takes time. So, we built this system to help them learn more about the characteristics of a brain tumor, as well as determine which areas of the brain are affected and which healthy tissue to avoid for the best and safest approach to treatment.

##### B. User Characteristics

There is no experience needed to use our system. Since doctors could easily upload MR Brain Image and it would be classified.

## 4.2 System Analysis & Design

### 4.2.1 Use Case Diagram



*Figure 4.2 Use Case Diagram*

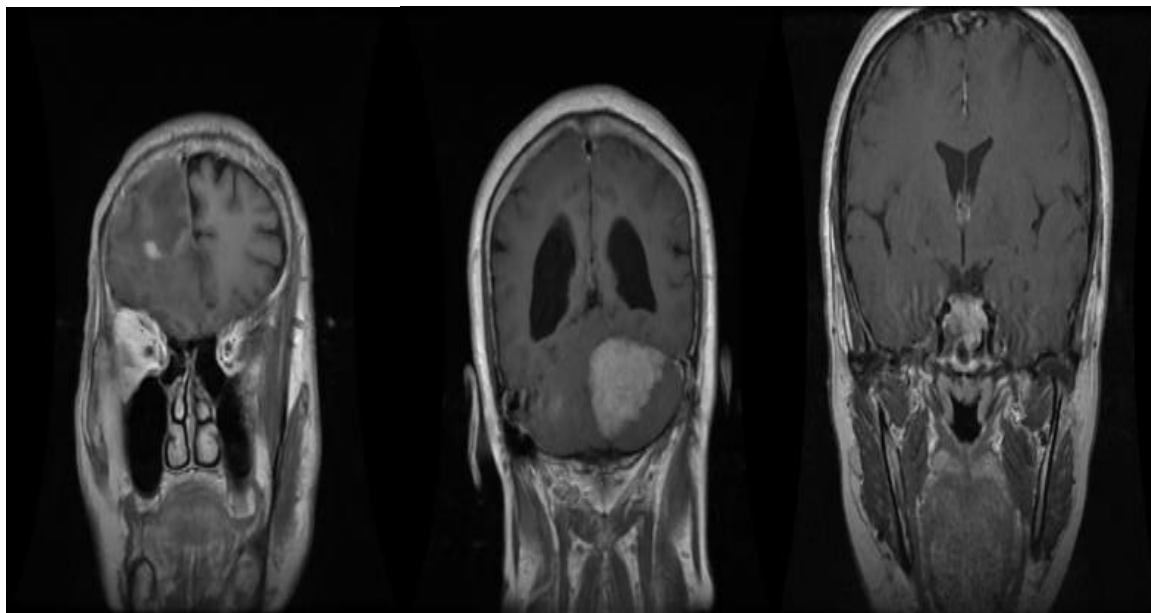
## 5- Implementation and Testing

### 5.1 Dataset

#### 5.1.1 Abnormal

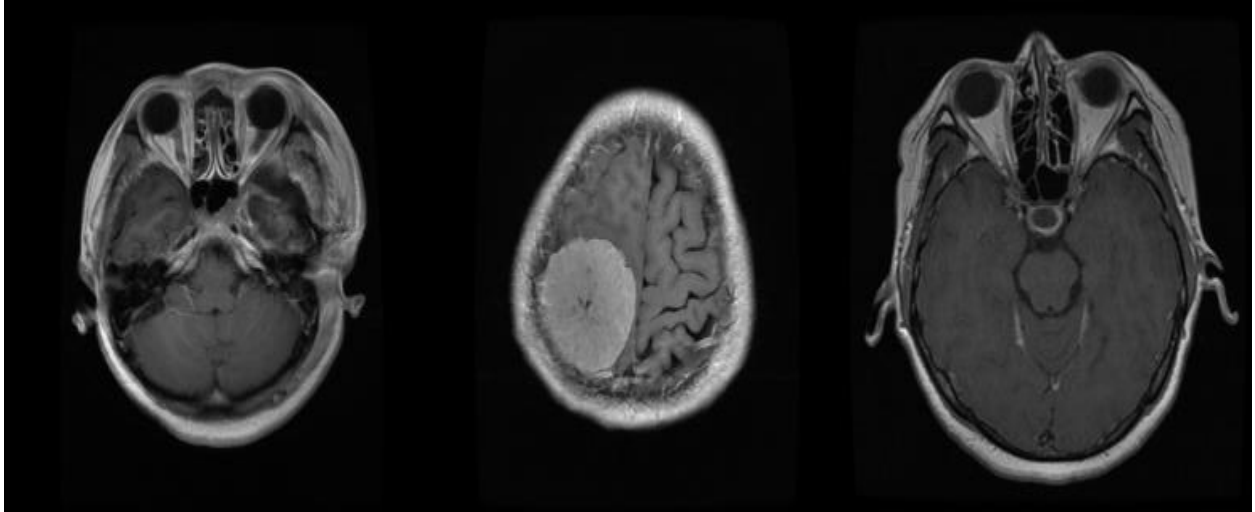
- ❑ To ensure accurate results, The Dataset is acquired from
  - Hospital
- ❑ The dataset for brain tumor comprises of:  
233 patients with 3064 images, classified as following
  - 708 meningioma's
  - 1426 gliomas
  - 930 pituitary tumors
- ❑ We are challenged to find MRI for normal brain with 3 views not only one view

#### 5.1.2 Abnormal Coronal



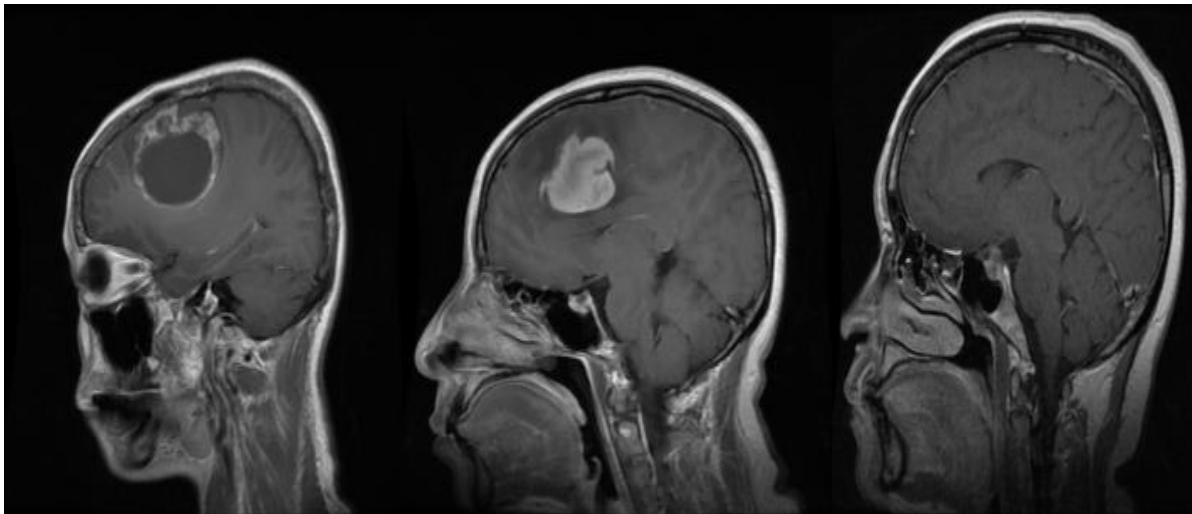
*Figure 5.1 Abnormal Coronal Example*

### 5.1.3 Abnormal Axial



*Figure 5.2 Abnormal Axial Example*

### 5.1.4 Abnormal Sagittal



*Figure 5.3 Abnormal Sagittal Example*

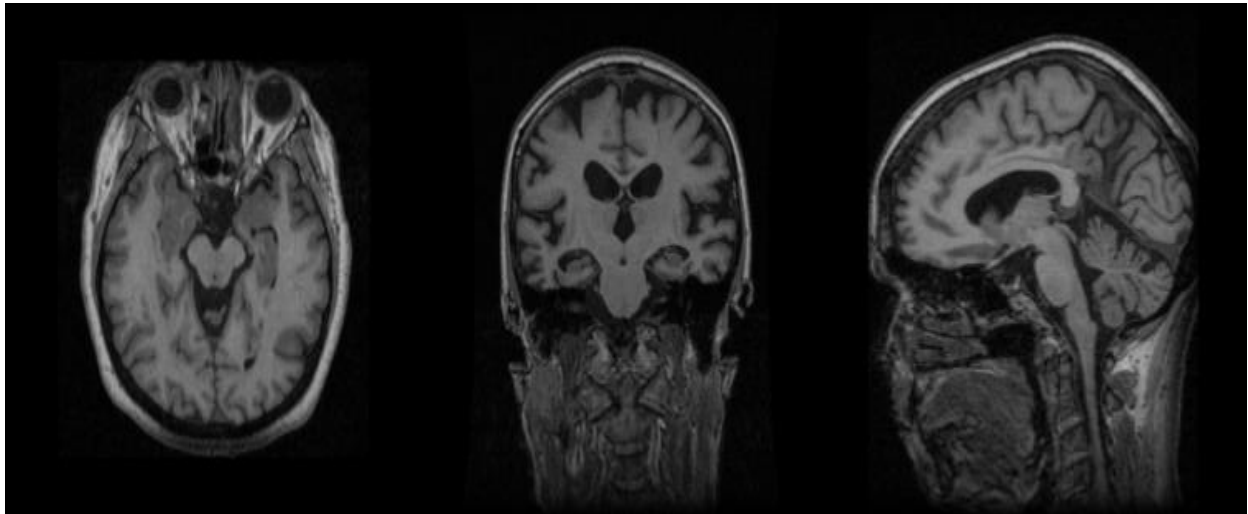
### 5.1.5 Normal

At the first we only have an abnormal dataset. the challenge we must get a normal dataset.

We start to search online in the first on Kaggle, google and a lot of searching sites but we did not find it.

After that we try to get it from an Egyptian hospital like (Bahia, 57357, alfa scan, techno scan) but for patient privacy they refused.

We are sending a lot of mail for an outside hospital until one of them respond with a dataset for a normal dataset and its software called mango



*Figure 5.4 Normal Brain MR Images Example*



### 5.1.6 MATLAB work in image

First, we start to load data and doing a pre-processing for it like Noise removal, Resizing, and Intensity normalization

At Noise removal we are using noise removal because our data is a sensitive for noise, we use a median filter to remove the noise

At Resizing dataset was 512x512 we trying to reduce the dataset size to 224x224

Intensity normalization is a process that changes the range of pixel intensity values. Applications include photographs with poor contrast due to glare, for example. Normalization is sometimes called contrast stretching or histogram stretching. In more general fields of data processing, such as digital signal processing, it is referred to as dynamic range expansion

## 5.2 Models

### 5.2.1 VGG-16

The architecture depicted below is VGG16.

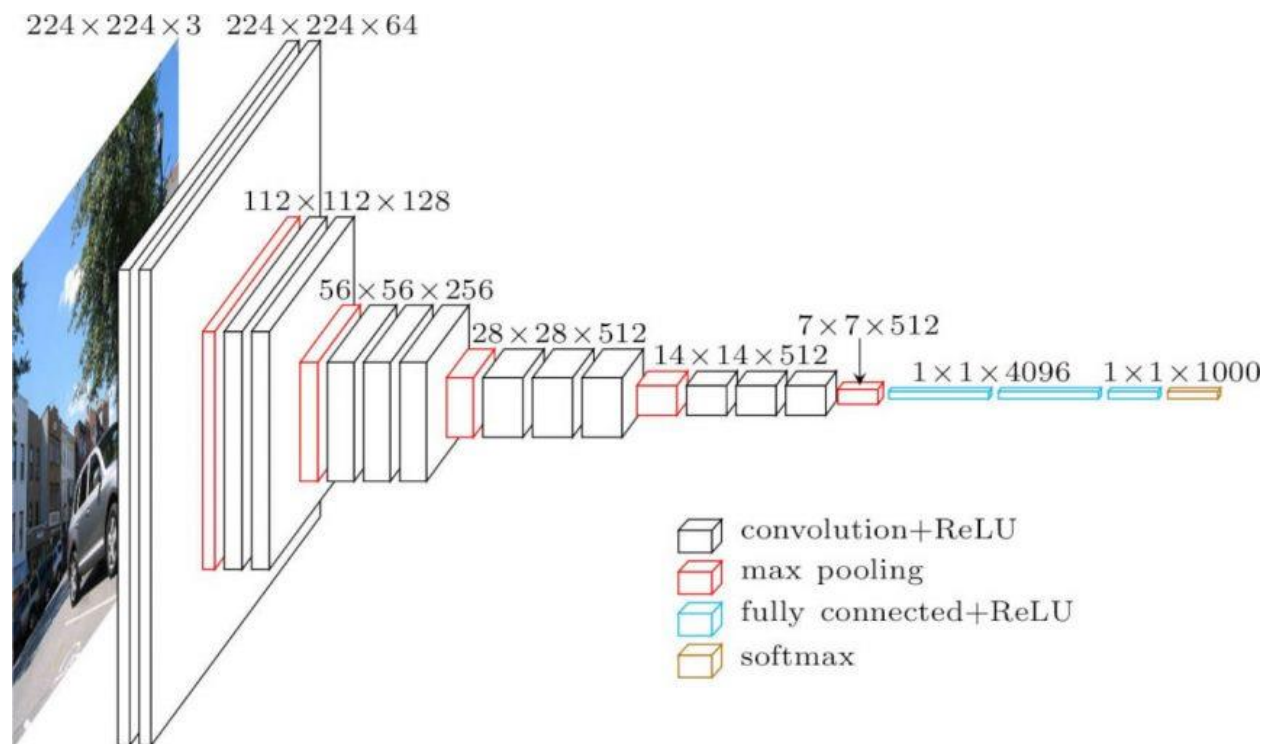


Figure 5.5 VGG-16 Architecture

The input to conv1 layer is of fixed size 224 x 224 RGB image. The image is passed through a stack of convolutional (conv.) layers, where the filters were used with a very small receptive field: 3x3 (which is the smallest size to capture the notion of left/right, up/down, center). In one of the configurations, it also utilizes 1x1 convolution filter, which is a linear transformation of the input channels (followed by non-linearity). The convolution stride is fixed to 1 pixel; the spatial padding of conv. layer input is such that the spatial resolution is preserved after convolution, i.e. the padding is 1-pixel for 3x3 conv. layers. Spatial pooling is carried out by five max-pooling layers, which follow some of the conv. layers (not all the conv. layers are followed by max-pooling). Max-pooling is performed over a 2x2-pixel window, with stride 2.

Three Fully-Connected (FC) layers follow a stack of convolutional layers (which has a different depth in different architectures): the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). The final layer is the soft-max layer. The configuration of the fully connected layers is the same in all networks.

All hidden layers are equipped with the rectification (ReLU) non-linearity. It is also noted that none of the networks (except for one) contain Local Response Normalization (LRN), such normalization does not improve the performance on the ILSVRC dataset but leads to increased memory consumption and computation time.

## 5.2.2 VGG-19

VGG-19 is a trained Convolutional Neural Network, from Visual Geometry Group, Department of Engineering Science, University of Oxford. The number 19 stands for the number of layers with trainable weights. 16 Convolutional layers and 3 Fully Connected layers.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input ( $224 \times 224$ RGB image)					
conv3-64	conv3-64 <b>LRN</b>	conv3-64 <b>conv3-64</b>	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 <b>conv3-128</b>	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 <b>conv1-256</b>	conv3-256 conv3-256 <b>conv3-256</b>	conv3-256 conv3-256 conv3-256 <b>conv3-256</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 5.6 VGG-19 Architecture

utilizes the Architectural style of:

Zero-Center normalization\* on Images, Convolution, ReLU, Max Pooling, Convolution

UNTIL

Fully Connected layer, ReLU, Dropout, Fully Connected, SoftMax, Classification output

**Zero-Center normalization** is the Centralization and Normalization towards Origo.

It normalizes and reduces dimensions - to keep scale centralized - in terms of when we will perform Convolutions later.

The reason this is important - is to bring everything “down in line” in a normalized, streamlined, and orderly fashion - so we have some sense of normality condition.

As in - we want the general structure of what we are parsing - to be normalized and centralized - so that we have a pre-defined boundary that we are being relative towards.

**Convolutions** is the functional operation of performing concatenation of Functional Curves - so that they “add up to encapsulate how they affect each other - in terms of multiplicative relationships” - roughly speaking.

What you are doing - is that you are basically “multiplying a function relationship with another” - so you get the average of how they affect each other - on average.

That is kind of the gist of it.

**ReLU** is a rectifying linear unit. A unit - that is made to rectify and compensate for signal parsing errors - such as when we are forced to encapsulate epsilon (infinitely small) - and we must work around that.

So ReLU “Steers back the signal” to where it is supposed to be headed - so that the signal does not explode or vanish.

There of - ReLU’s act kind of like “Safety stops” on the sides of the roads - so you do not steer off the road - when you are driving. Sort of.

**Max Pooling** is when you pool together the largest sample you can find - in an average area of taking strides.

Strides - is the average functional kernel mapping - that you have in a square diagram plot - that averages out the value samplings of a certain space.

So - you may have a 8x8 total square - with 4 4x4 squares - So - you reduce that to being the max of every 4x4 square - put them together - and get a 4x4 total Square to replace your earlier 8x8.

This effectively takes “the biggest impact features” and there of, “the largest information to retrieve” (read: coarsest features) - And then ignores the smaller features -

So, you are left with a smaller sample space (lower dimensionality) - But - you keep the coarsest features - you keep the largest representative of information.

It is a trade-off for computational sake.

**A Fully Connected Layer** is a fully connected layer - that is utilized for Classification.

Since this is used so sparingly - it is at the end - when the rough feature extraction and averaging of Functional relationships have had its sequence -

Now - it summarizes to run tally on the total output.

**SoftMax** is an applicative algorithm - that normalizes a distribution dynamic - from K amount of composite functional concatenations.

This means - that - instead of having a spread where the values can be anything - 0, -1, 1, 2, etc.

They are all normalized to be in line to a distribution (as the SoftMax act as a regularizer over a distribution)

And then there of - generalized to be across the  $j$ -th Linear function - as it has become baked into the Distribution, we have regressed forward with SoftMax.

**Classification output** is pretty much what it sounds like.

It is the output of classification - where it utilizes Cross Entropy to maximize the Probability - in terms of where each  $K$ -th unit came from - and where it most likely belongs.

There of - it stands in relation to the previous  $K$  functional relationships - where it classifies and infers the probability of placement - of each.

### 5.2.3 Res-Net 50

A residual neural network (ResNet) is an (ANN) of a kind that builds on constructs known from pyramidal cells in the cerebral cortex. Residual neural networks do this by utilizing skip connections, or shortcuts to jump over some layers. Typical Res-Net models are implemented with double- or triple- layer skips that contain nonlinearities (ReLU) and batch normalization in between. An additional weight matrix may be used to learn the skip weights; these models are known as Highway Nets. Models with several parallel skips are referred to as Dense Nets. In the context of residual neural networks, a non-residual network may be described as a plain network.

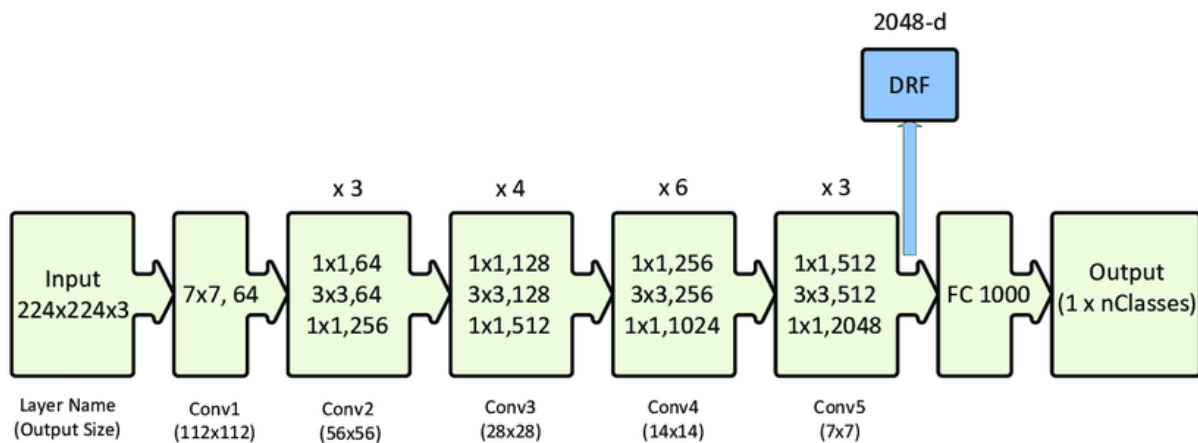


Figure 5.7 ResNet-50 Architecture

ResNet-50 architecture with the residual units, the size of the filters and the outputs of each convolutional layer. DRF extracted from the last convolutional layer of this network is also shown. Key: The notation  $k \times k, n$  in the convolutional layer block denotes a filter of size  $k$  and  $n$  channels. FC 1000 denotes the fully connected layer with 1000 neurons. The number on the top of the convolutional layer block represents the repetition of each unit. N Classes represents the number of output classes.



#### 5.2.4 SVM

SVM are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. The Support Vector Machine (SVM) algorithm is a popular machine learning tool that offers solutions for both classification and regression problems. Developed at AT&T Bell Laboratories by Vapnik with colleagues (Boser et al., 1992, Guyon et al., 1993, Vapnik et al., 1997), it presents one of the most robust prediction methods, based on the statistical learning framework or VC theory proposed by Vapnik and Chervonekis (1974) and Vapnik (1982, 1995). Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong

to a category based on the side of the gap on which they fall.

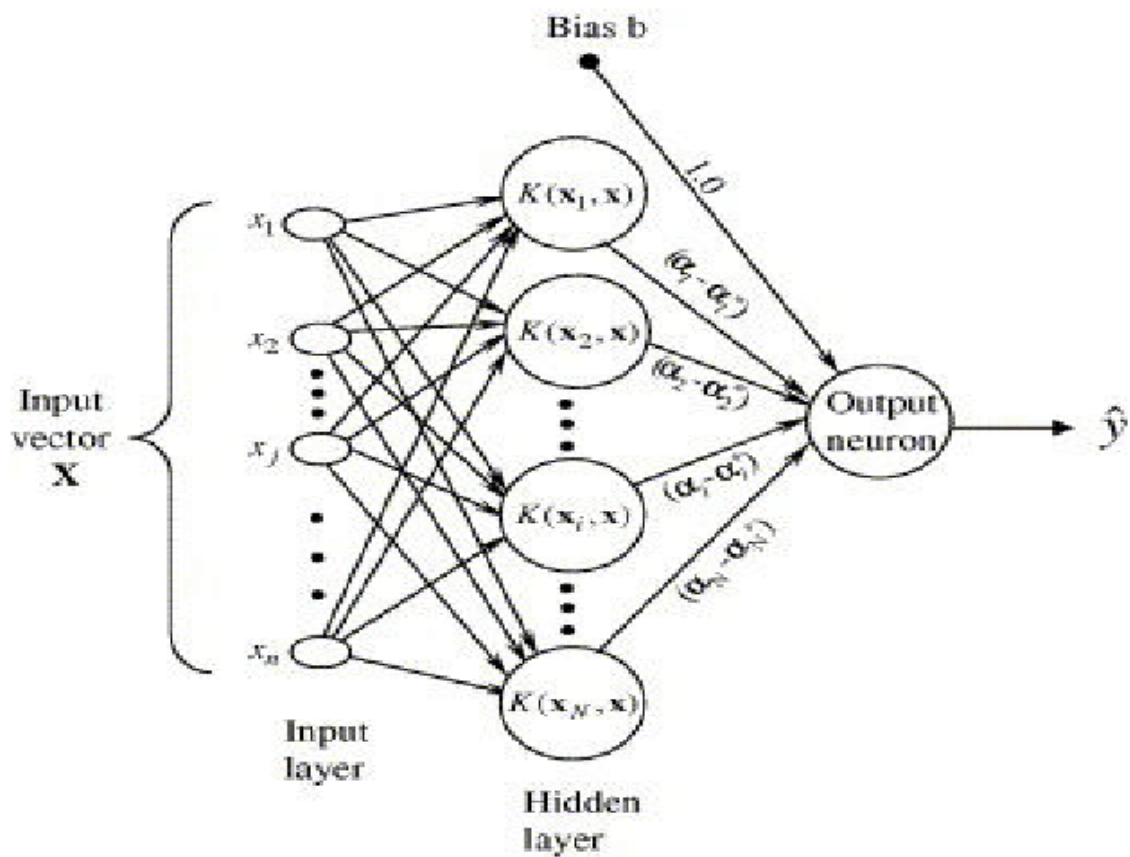


Figure 5.8 SVM Architecture

in SVM model we are trying only in abnormal data because we did not have a normal data when we try it.

## 5.3 Experimental Results for All Architecture

We used multiple CNN Architecture like VGG-16, VGG-19, Res-Net 50 and SVM. We started with SVM and after getting the best results we could get from it we tried VGG-16, VGG19, and Res-Net 50. The results we got from them was better than we got from SVM.

For these architectures we tried optimizers, Activation Function, batch size, number of epochs, loss function, and learning rates until we got the best result for each architecture.

### 5.3.1 Optimizers

**Adam** is an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data.

Adam is different to classical stochastic gradient descent. Stochastic gradient descent maintains a single learning rate (termed alpha) for all weight updates and the learning rate does not change during training. A learning rate is maintained for each network weight (parameter) and separately adapted as learning unfolds.

**SGD(Stochastic Gradient Descent)** is an iterative method for optimizing an objective function with suitable smoothness properties (e.g. differentiable or subdifferentiable). It can be regarded as a stochastic approximation of gradient descent optimization, since it replaces the actual gradient (calculated from the entire data set) by an estimate thereof (calculated from a randomly selected subset of the data). Especially in high-dimensional optimization problems this reduces the computational burden, achieving faster iterations in trade for a lower convergence rate.

**Ada-Max** is an extension of Adam based on the infinity norm. This optimizer takes the idea of Adam one step further, and for the current gradient takes the maximum of  $\beta_2$  x past gradients and the current gradient. This results in a smaller and smoother movement towards the minima.

After experiments we found that SGD is the optimizer with the best result for accuracy.

### 5.3.2 Activation Function

Neural network activation functions are a crucial component of deep learning. Activation functions determine the output of a deep learning model, its accuracy, and the computational efficiency of training a model- which can make or break a large-scale neural network. Activation functions also have a major effect on the neural network's ability to converge and the convergence speed, or in some cases, activation functions might prevent neural networks from converging in the first place.

We used lots of activation functions in our architectures like ReLU, Sigmoid, and SoftMax. But ReLU is the best for fully connected layer, and SoftMax for the output layer.

### 5.3.3 Batch Size

Batch size is a term used in machine learning and refers to the number of training examples utilized in one iteration. The batch size can be one of three options:

- **batch mode:** where the batch size is equal to the total dataset thus making the iteration and epoch values equivalent
- **mini-batch mode:** where the batch size is greater than one but less than the total dataset size. Usually, a number that can be divided into the total dataset size.
- **stochastic mode:** where the batch size is equal to one. Therefore, the gradient and the neural network parameters are updated after each sample.

we used Mini-Batch Gradient descent by updating the weights of the model every k samples where the popular number of k is 32, 64, 128 samples but we used 32 as it had the best results in all architectures.

### 5.3.4 Number of Epochs

The number of epochs is a hyperparameter that defines the number times that the learning algorithm will work through the entire training dataset. One epoch means that each sample in the training dataset has had an opportunity to update the internal model parameters. An epoch is comprised of one or more batches. For example, an epoch that has one batch is called the batch gradient descent learning algorithm.

We used epochs from 4 up to 15.

### 5.3.5 Loss Function

In mathematical optimization and decision theory, a loss function or cost function is a function that maps an event or values of one or more variables onto a real number intuitively representing some "cost" associated with the event. An optimization problem seeks to minimize a loss function.

An objective function is either a loss function or its negative (in specific domains, variously called a reward function, a profit function, a utility function, a fitness function, etc.), in which case it is to be maximized.

In statistics, typically a loss function is used for parameter estimation, and the event in question is some function of the difference between estimated and true values for an instance of data. The concept, as old as Laplace, was reintroduced in statistics by Abraham Wald in the middle of the 20th century. In the context of economics, for example, this is usually economic cost or regret. In classification, it is the penalty for an incorrect classification of an example. In actuarial science, it is used in an insurance context to model benefits paid over premiums, particularly since the works of Harald Cramér in the 1920s. In optimal control, the loss is the penalty for failing to achieve a desired value. In financial risk management, the function is mapped to a monetary loss.

In classical statistics (both frequentist and Bayesian), a loss function is typically treated as something of a background mathematical convention.

We used sparse categorical cross entropy as loss function.

### 5.3.6 Learning Rate

In machine learning and statistics, the learning rate is a tuning parameter in an optimization algorithm that determines the step size at each iteration while moving toward a minimum of a loss function. Since it influences to what extent newly acquired information overrides old information, it metaphorically represents the speed at which a machine learning model "learns". In the adaptive control literature, the learning rate is commonly referred to as gain.

We used learning rates from 0.0001 up to 0.001.

### 5.3.7 Table of Accuracy

*Table 5-1 Accuracy Table*

VGG-16	79% up to 84% on Abnormal
VGG-19	80% up to 85% on Abnormal
SVM	98% For Classifying Views
RES-Net	97% On the Whole Data

*RES-Net accuracy Is 97% the highest one of all models.*

```
100%|██████████| 4959/4959 [00:19<00:00, 248.76it/s]
Epoch 1/4
 1/139 [.....] - ETA: 0s - loss: 6.0904 - accuracy: 0.0938WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python
Instructions for updating:
use `tf.profiler.experimental.stop` instead.
 2/139 [.....] - ETA: 35s - loss: 6.1429 - accuracy: 0.1562WARNING:tensorflow:Callbacks method `on_train_batch_end` is slow compared to th
139/139 [=====] - 48s 344ms/step - loss: 0.6440 - accuracy: 0.8610
Epoch 2/4
139/139 [=====] - 48s 342ms/step - loss: 0.0981 - accuracy: 0.9718
Epoch 3/4
139/139 [=====] - 48s 343ms/step - loss: 0.0319 - accuracy: 0.9889
Epoch 4/4
139/139 [=====] - 47s 341ms/step - loss: 0.0253 - accuracy: 0.9910
52/52 [=====] - 5s 103ms/step - loss: 0.1001 - accuracy: 0.9707
0%|          | 0/4959 [00:00<?, ?it/s][INFO] loss=0.1001, accuracy: 97.0678%
```

*Figure 5.9 Res-Net Model Accuracy*

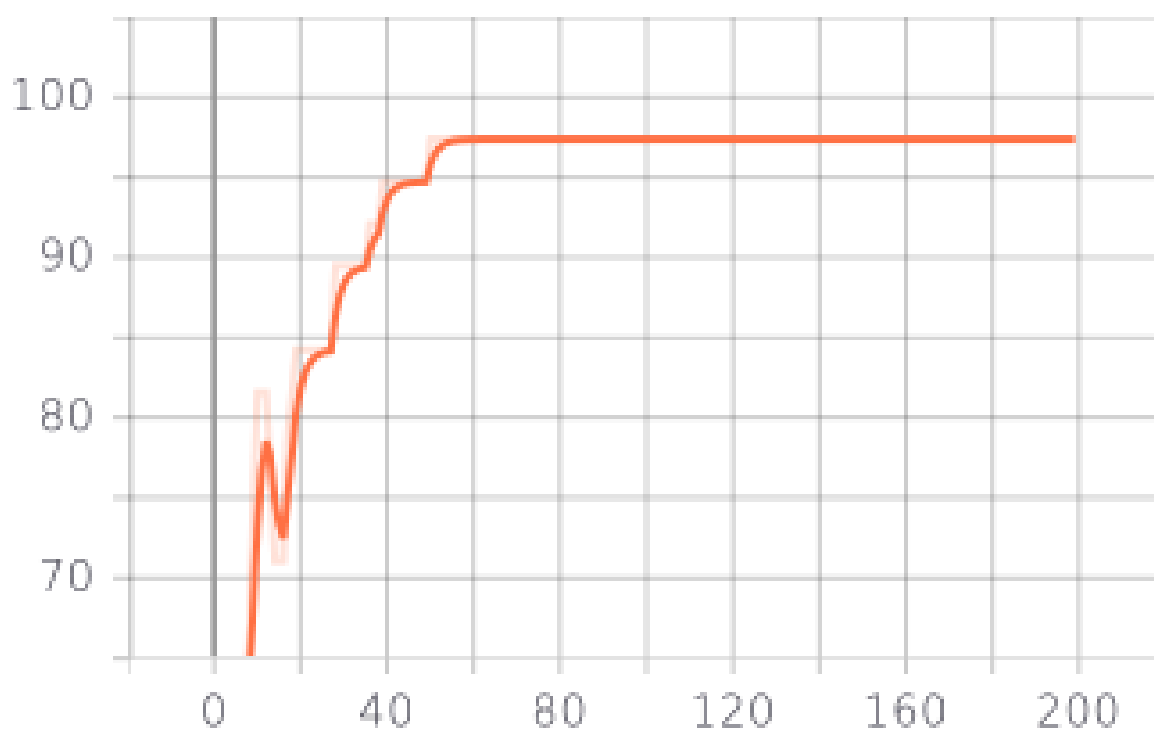


Figure 5.10 Overall Accuracy

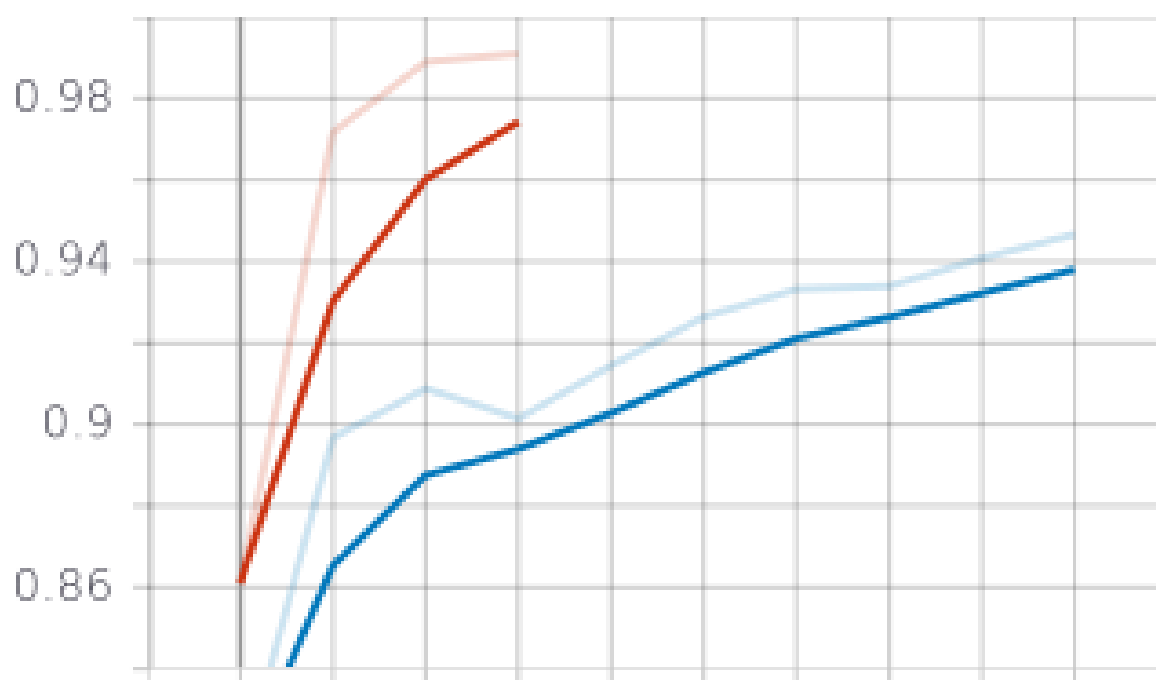
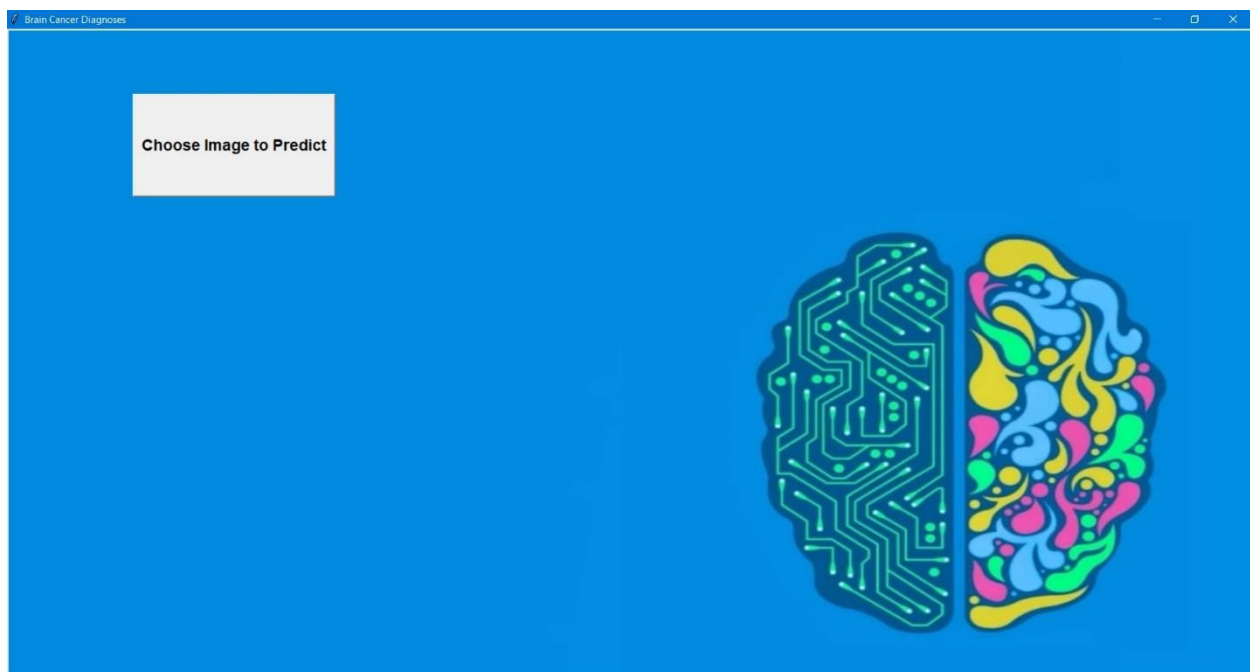


Figure 5.11 Training Accuracy

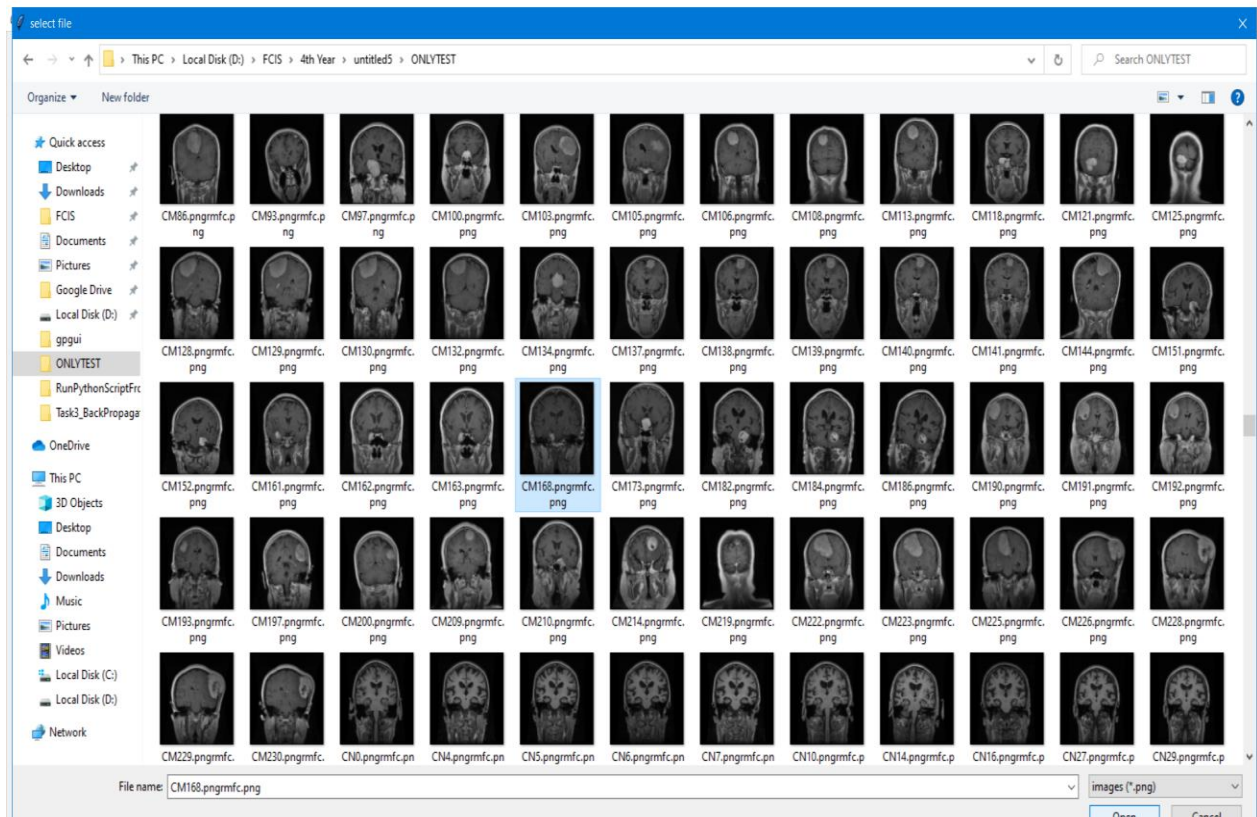
## 6- User Manual

To start using our application user should press the 'Choose Image to Predict' button

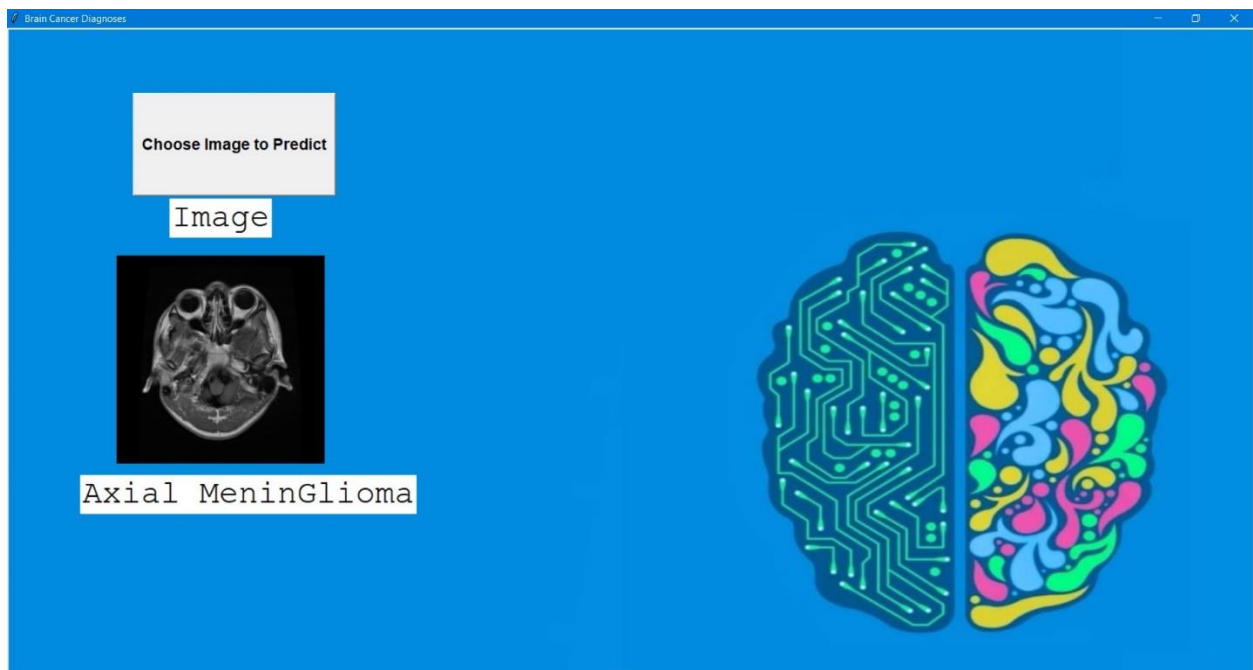




Then the user should choose the image to classify



Then the system will display the classified image along with its classification result



## **7- Conclusion and Future Work**

### **7.1 Conclusion**

In this project, the computer-based techniques for automatic classification of MR images as normal or abnormal using different classifier.

We worked on dataset we collected from multiple sources (Kaggle, some outside hospitals that respond to us).

We resized Images to 224x224 image resolution and tried training the models with RGB images. Then we normalized and removing noise using median filter for images because MR images are too sensitive for noise. Finally, we start our training. The performances of the classifiers in terms of statistical measures such as loss rate and accuracy are analyzed.

We trained some of CNNs models like VGG-16, VGG-19, Res-Net, Volume-Net, Link-Net and SVM. The results indicated that RES-Net approach yielded the better performance when compared to other classifiers as we achieved accuracy of 97% while other classifiers achieved accuracy ranged between 93% to 95%.

With each model we used we tried different hyper parameter like changing learning rate, optimizers and various numbers of epochs.

We used different data augmentation techniques to increase and balance our dataset to achieve higher accuracy.

## 7.2 Future Work

We will try to increase the dataset to improve accuracy and we will get MR image for a breast cancer and lung cancer and more cancer types to add more functionality for our software like make a segmentation for MR image to know the exact cancer place. Also, trying to make a marketing for software and get a sponsor to be real world project.

Another aspect that may improve the performance is to involve more complicated image preprocessing procedures. MR images is susceptible to noise, so inhomogeneity correction and noise removal algorithms can be applied before feature extraction. Although we only use the simple min-max method to normalize intensity values, the experimental results are promising.

## 8- References

- 1) Ali İüÖna, Cem Direkoglu, Melike üahc. Review of MRI-based brain tumor image segmentation using deep learning methods. *Procedia Computer Science* (2016).
- 2) Evangelia I. Zacharaki, Sumei Wang, Sanjeev Chawla, Dong Soo Yoo, Ronald Wolf, Elias R. Melhem, Christos Davatzikos. MRI-BASED CLASSIFICATION OF BRAIN TUMOR TYPE AND GRADE USING SVM-RFE. *IEEE International Symposium on Biomedical Imaging* (2014).
- 3) B. Sudha, P. Gopikannan, A. Shenbagarajan, C. Balasubramanian. Classification of Brain Tumor Grades using Neural Network. *Proceedings of the World Congress on Engineering* (2014).
- 4) Sudipta Roy a, Debnath Bhattacharyya, Samir Kumar Bandyopadhyay, Tai-Hoon Kim. Heterogeneity of human brain tumor with lesion identification, localization, and analysis from MRI. *Elsevier* (2019).
- 5) Zeynettin Akkus, Alfiia Galimzianova, Assaf Hoogi, Daniel L. Rubin, Bradley J. Erickson. Deep Learning for Brain MRI Segmentation: State of the Art and Future Directions. *J Digit Imaging* (2017).
- 6) Jun Cheng, Wei Huang, Shuangliang Cao, Ru Yang, Wei Yang, Zhaoqiang Yun, Zhijian Wang, Qianjin Feng. Enhanced Performance of Brain Tumor Classification via Tumor Region Augmentation and Partition. *PLoS ONE* (2015).
- 7) R. Thillaikkarasi, S. Saravanan. An Enhancement of Deep Learning Algorithm for Brain Tumor Segmentation Using Kernel Based CNN with M-SVM. *Journal of Medical Systems* (2019).
- 8) Xiaomei Zhao, Yihong Wu, Guidong Song, Zhenye Li, Yong Fan and Yazhuo Zhang. Brain Tumor Segmentation Using a Fully Convolutional Neural Network with Conditional Random Fields. *Chinese Academy of Sciences* (2016).
- 9) Geetika Gupta, Rupinder Kaur, Arun Bansal, Munish Bansal. Analysis and Comparison of Brain Tumor Detection and Extraction Techniques from MRI Images. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering* (2014).
- 10) Mandhir Kaur, Rinkesh Mittal. An Efficient Scheme for Brain Tumor Detection of MRI Brain Images Using Euclidean Distance With FVT. *JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLIGENCE* (2014).

- 11) J. Seetha, S. Selvakumar Raja. Brain Tumor Classification Using Convolutional Neural Networks. Biomedical & Pharmacology Journal (2018).
- 12) Vidyasagar, Bhujangarao, Madhu. Qualitative Analysis of Brain tumor MRI. The International Daily journal (2015).
- 13) Ali ARI, Davut HANBAY. Deep learning-based brain tumor classification and detection system. Turkish Journal of Electrical Engineering & Computer Sciences (2018).
- 14) Subhashis Banerjee, Sushmita Mitra. Brain Tumor Detection and Classification from Multi-Channel MRIs using Deep Learning and Transfer Learning. IEEE (2018).
- 15) Zahra Sobhaninia, Safiyeh Rezaei, Alireza Noroozi, Mehdi Ahmadi, Hamidreza Zarrabi, Nader Karimi, Ali Emami, Shadrokh Samavi. Brain Tumor Segmentation Using Deep Learning by Type Specific Sorting of Images. Department of Electrical and Computer Engineering (2018).
- 16) Yuehao Pan, Weimin Huang, Zhiping Lin, Wanzheng Zhu, Jiayin Zhou, Jocelyn Wong, Zhongxiang Ding. Brain Tumor Grading Based on Neural Networks and Convolutional Neural Networks. Society. IEEE Engineering in Medicine and Biology Society (2016).
- 17) Sajid Iqbal<sup>1</sup>, M. Usman Ghani, Tanzila Saba, Amjad Rehman. Brain tumor segmentation in multi-spectral MRI using convolutional neural networks (CNN). EMS Microscopy Academy (2018).
- 18) Jiachi Zhang, Xiaolei Shen, Tianqi Zhuo, Hong Zhou. Brain Tumor Segmentation Based on Refined Fully Convolutional Neural Networks with A Hierarchical Dice Loss. Key Laboratory for Biomedical Engineering (2017).
- 19) Mohammadreza Soltaninejad, Lei Zhang, Tryphon Lambrou, Guang Yang, Nigel Allinson, and Xujiong Ye. MRI Brain Tumor Segmentation using Random Forests and Fully Convolutional Networks. MICCAI (2017).
- 20) Parasuraman Kumar, B. VijayKumar. Brain Tumor MRI Segmentation and Classification Using Ensemble Classifier. International Journal of Recent Technology and Engineering (2019).
- 21) <https://neurohive.io/en/popular-networks/vgg16>
- 22) <https://iq.opengenus.org/vgg19-architecture/vgg19>
- 23) [https://www.researchgate.net/figure/ResNet-50-architecture-26-shown-with-the-residual-units-the-size-of-the-filters-and\\_fig1\\_338603223](https://www.researchgate.net/figure/ResNet-50-architecture-26-shown-with-the-residual-units-the-size-of-the-filters-and_fig1_338603223)

- 24) <https://courses.lumenlearning.com/boundless-psychology/chapter/neurons/>
- 25) <https://community.hpe.com/t5/hpe-blog-uk-ireland/artificial-intelligence-enough-of-the-hype-what-is-it/ba-p/7046672>
- 26) <https://towardsdatascience.com/converting-a-simple-deep-learning-model-from-pytorch-to-tensorflow-b6b353351f5d>
- 27) [https://www.researchgate.net/figure/Architecture-of-support-vector-machine\\_fig2\\_255822126](https://www.researchgate.net/figure/Architecture-of-support-vector-machine_fig2_255822126)