ARTIFICIAL INTELLIGENCE ASSISTED BRAIN TUMOR DETECTION

A Project Report

Submitted in partial fulfilment of the requirements

for the award of the Degree of

MASTER OF SCIENCE (INFORMATION TECHNOLOGY)

By

SAIKUMAR SAMBASHIVARAO KOLIPAKA

Seat Number:- **4135577**

Under the esteemed guidance of

Ms. Seema Bhatkar

Assistant Professor, Department of Information Technology

DEPARTMENT OF INFORMATION TECHNOLOGY

VIDYALANKAR SCHOOL OF INFORMATION TECHNOLOGY

(Affiliated to University of Mumbai)

Mumbai, 400 037

MAHARASHTRA

2022 - 2023

VIDYALANKAR SCHOOL OF INFORMATION TECHNOLOGY

(Affiliated to University of Mumbai)

MUMBAI-MAHARASHTRA-400037

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled, **"Artificial Intelligence Assisted Brain Tumor Detection"**, is bona fide work of **SAIKUMAR KOLIPAKA** bearing Seat No :- **4135577** submitted in partial fulfilment of the requirements for the award of degree of MASTER OFSCIENCE in INFORMATION TECHNOLOGY from University of Mumbai.

Internal Guide **Coordinator**

Internal Examiner **External Examiner**

Date: **College Seal** **Principal**

RESEARCH PAPER

Artificial Intelligence Assisted Brain Tumor Detection

Ameya Gadekar
21306A1020
Masters Of Science In Information Technology
ameyagadekar07@gmail.com

Saikumar Kolipaka
21306A1032
Masters of Science In Information Technology
saikumarkolipaka31@gmail.com

ABSTRACT: - As we know deep learning has been proven to be superior in detecting diseases which could significantly improve the accuracy and speed of the diagnosis. Our team will collect brain MRI scans and will approach to develop a model that could detect and localize tumors. Our project is going to set layered deep learning pipeline to perform classification and segmentation. In this process an input of image Brain MRI scan will be feeded into Resnet deep learning classifier model this model will give you two outputs one where the tumor is detected and other where it is not detected. If the tumoris detected it would be send to Resunet segmentation model where the detection of the tumor will be located on pixel level, and if the tumor is not detected the patient is healthy.

I. INTRODUCTION

In today's world Artificial Intelligence is revolutionizing Healthcare in many areas such as: Disease Diagnosis with medical imaging, Surgical Robots and Maximizing Hospital Efficiency. Deep learning has been proven to be superior in detecting diseases from X-rays, MRI scans and CT scans which could significantly improve the speed and accuracy of diagnosis. The MRI Scans are collected and the model is developed. This would drastically reduce the cost of cancer diagnosis & help in early diagnosis of tumors which would essentially be a life saver

A tumor is a collection of aberrant cells that form a tissue. These aberrant cells consume regular body cells, kill them, and continue to expand in size. Brain tumor is one of these tumors. Nerve cells, brain cells, glands, and membranes thatsurround the brain are all affected. Imaging and pathology can both be used to diagnose a tumor.

An MRI (Magnetic Resonance Imaging) scan of a brain tumor produces a cross-section picture of the brain. In this report, we provide two models based on Artificial Convolutional Neural Networks that uses mathematical formulae and algebraic operations to analyze and predict if the tumor exists and to localize the area of tumor on MRI Images.

1.1.1 Prediction

Prediction can be used to automatically sort the ".cv" file. Prediction is used in situations where there are multiple possibilities to a singleproblem. There are many types of algorithm that can be used k-means, agglomerative algorithm.

1.1.2 Classification

When the data are being used to predict a category, supervised learning is also called classification. When there are only two choices, it's called two-class or binomial classification. That is clustering of the nearest approximate value.

1.1.3 Regression

When a value is being predicted, as with stock prices, supervised learning is called regression.

1.2.1 Aim

The main aim of our project is to help doctors to improve speed and accuracy of detecting and localizing the brain tumor based on MRI scan. This would drastically reduce Tumor Diagnosis and help in early diagnosis of tumor which would essentially be a life saver.

1.2.2 Objective

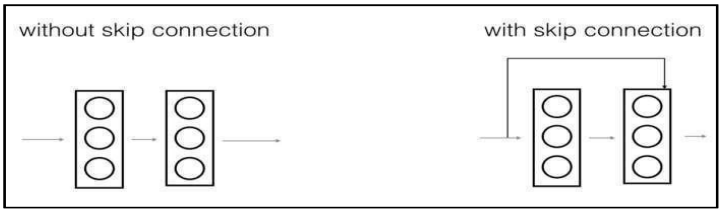
Deep learning has been proven to be superior in detecting diseases from X-rays, MRI scans and CT scans which could significantly improve the speed and accuracy of diagnosis. Time consumption is less for detecting and localizing tumor. Our project is going to set layered deep learning pipeline to perform classification and segmentation.

II. RELATED WORKResNet

ResNet, or Residual Networks, is a type of classic neural network that is used in many computer vision applications. The main innovation with ResNet was that it enabled us to effectively train incredibly deep neural networks with 150+ layers. However, just stacking layers together does not work toincrease network depth. Because of the well-known vanishing gradient problem, deepnetworks are difficult to train. When gradients are back-propagated to older layers, repeated multiplication can result in exceedingly tiny gradients. When a result, as the network penetrates deeper, its performance becomessaturated or even begins to degrade fast.

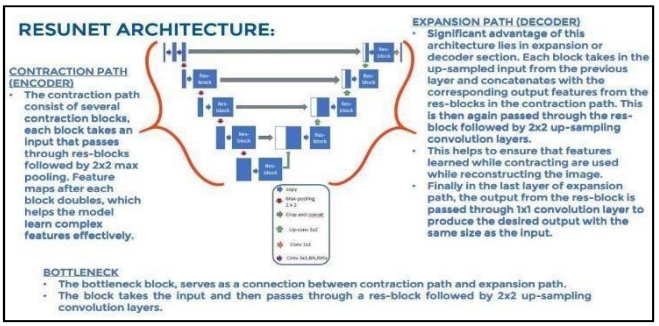
Skip Connection: The Strength of ResNet:

The notion of skip connection was initially presented by ResNet. The skip connection is depicted in the diagram below. The graphic on the left shows convolution layers being stacked one on top of the other. On the right, we continue to build convolution layers as previously, but now we additionally include the ResNet employs skip connections. This aids in resolving the vanishing gradient issue.



ResUNet -

To avoid the vanishing gradients difficulties that deep architectures have, the ResUNet design combines the UNet backbone architecture with residual blocks. Unet's architecture is based on Fully Convolutional Networks and has been tweaked to function well with segmentation tasks. Like a U-Net, the RESUNET has an encoding network, adecoding network, and a bridge linking the two networks. Each 3 x 3 convolution in the U-Net is followed by a ReLU activation function.



Classification Model.

A classification model tries to derive some conclusion from the training input data. It will anticipate the class labels/categories of the fresh data. In order for a classification model to perform well, it must be trained on a large amount of data. Imagedatagenrator from the keras library is being utilised to deliver data at such a large scale. By altering the variables,new sorts of data will be created from current photographs. Some operations are performed by the proposed convolutional neural network model, which are listed below:

Original input in the convolution block's output. This is referred to as a skip connection. To add the output from an earlier layer to a later layer,

Convolution operation -

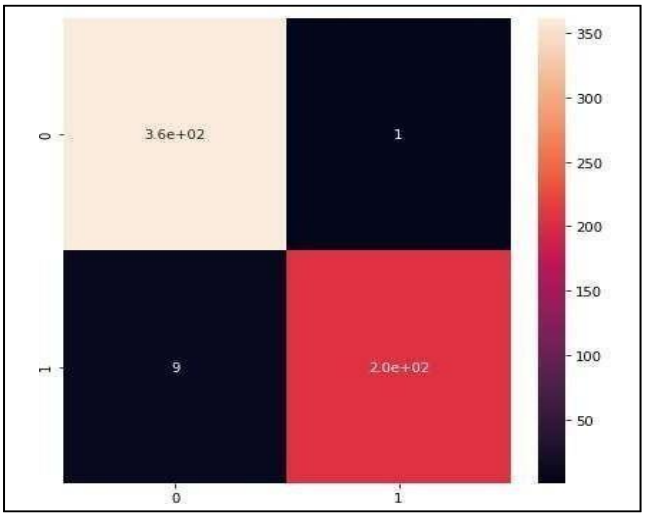
- Max pooling
- Flattening
- Full Connection

Image Segmentation.

Humans are able to see pictures by catching reflected light rays, which is a challenging process. So, how can machines be designed to accomplish the same thing? The pictures are seen by the computer as matrices that must be processed in order to gain significance. Image segmentation is the process of dividing an image into separate segments, each with its own entity. Convolutional Neural Networks have performed well for basic pictures, but not sowell for complicated ones. Other algorithms, such as Res-Net and ResU-Net, come into play here.

Confusion Matrix.

The Confusion Matrix is a machine learning classification performance metric. It is a performance metric for machine learning classification problems with two or more classes as output. There are four different combinations of projected and actual values in this table. It's great for assessing things like recall, precision, specificity, accuracy, and, most crucially, AUC-ROC curves.



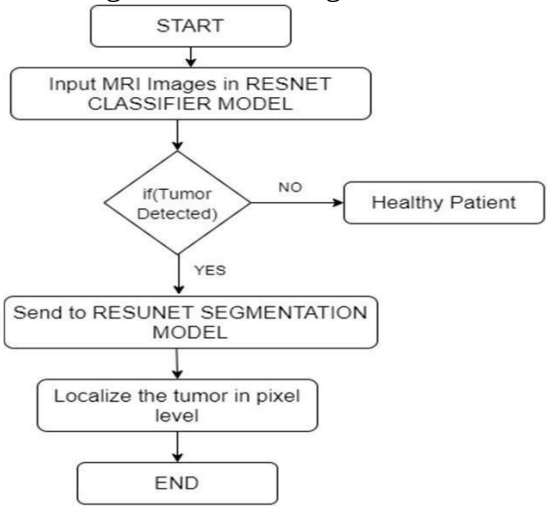
I. IMPLEMENTATION AND DESIGN.

As we know deep learning has been proven to be superior in detecting diseases which could significantly improve the accuracy and speed of the diagnosis. Our team will collect brain MRI scans and will approach to develop a model that could detect and localize tumors.

Our project is going to set layered deep learning pipeline to perform classification and segmentation. In this process an input of image Brain MRI scan will be feeded into ResNet deep learning classifier model this model will give you two outputs one where the tumor is detected and other where it is not detected.

If the tumor is detected it would be send toResUNet segmentation model where the detection of the tumor will be located on pixel level, and if the tumor is not detected the patientis healthy.

Fig: Flowchart Diagram



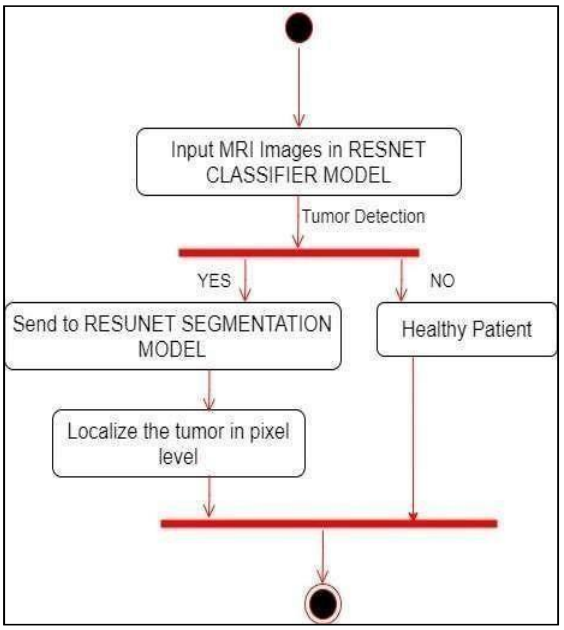


Fig: Activity Diagram

III. CONCLUSION :-

Tumors are a life-threatening menace to humanity. The neurological system, circulatory system, organ function, and so on are all affected by a brain tumour. There are several approaches for detecting cancers. Magnetic resonance imaging (MRI) is one way. The MRI produces cross-sectional pictures of the brain, which aids us in uncovering new areas of the brain. In our project, we used a dataset that we compiled from several sources. We had obtained 3929 brain MRI images in total. We created a Resnet Classifier Model to categorise MRI images as positive (label 1 = tumour) or negative (label 0 = no tumour) (i.e. do not have tumor). By evaluating our classifier modelusing a test set, we were able to achieve a test accuracy of 98 percent.

I. REFERENCE

[1]. Top 16 companies in AI-powered medical imaging https://research.aimultiple.com/looking- for-better-medical-imaging-for-early- diagnostic-and-monitoring-contact-the- leading-vendors-here/

[2]. Introduction to U-Net and Res-Net forImage Segmentation https://aditi- mittal.medium.com/introduction-to-u- net-and-res-net-for-image-segmentation- 9afcb432ee2f/

[3]. https://www.webmd.com/cancer/brain- cancer/qa/what-is-a-tumor, November,21 2019.

[4]. https://en.wikipedia.org/wiki/Brain_tumor, November, 21 2019.

[5] https://en.wikipedia.org/wiki/Convolutional_neural_network,December,21,2019.

[6]. https://en.wikipedia.org/wiki/Loss_function, December, 21 2019.

[7]. https://en.wikipedia.org/wiki/Mathematical_optimization, December, 21 2019

ACKNOWLEDGEMENT

It gives me immense pleasure in expressing my heartfelt thanks to the people who were part ofthis project in numerous ways. I owe my thanks to all those who gave endless support right from the conception of the project idea to its implementation, it would not have materialized without the help of many.

The dedication, hard work, patience and correct guidance makes any task proficient & a successful achievement. Intellectual and timely guidance not only helps in trying productive but also transforms the whole process of learning and implementing into an enjoyable experience.

I would like to thank our Principal "**Dr. Rohini Kelkar**" and vice principal **"Mr. Asif Rampurawala"** for providing this opportunity, a special thanks to our MSc IT coordinator "**Ms. Beena Kapadia**" for their support, blessings and for being a constant source of inspirationto us. With immense gratitude, I would like to convey my special honour and respect to "**Ms. Seema Bhatkar**" **(Project Guide)** who took keen interest in checking the minute details of theproject work and guided us throughout the same.

A sincere thanks to the non-teaching staff for providing us with the long lab timings that we could receive along with the books and with all the information we needed for this project, without which the successful completion of this project would not have been possible.

Finally, I wish to avail this opportunity & express a sense of gratitude and love to my friends and my beloved parents for their support, strength and help for everything.

Mr. SAIKUMAR SAMBASHIVARAO KOLIPAKA

DECLARATION

I hereby declare that the project entitled, **"Artificial Intelligence Assisted Brain Tumor Detection"** done at Vidyalankar School of Information Technology, has not been in any case duplicated to submit to any other universities for the award of any degree.To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **MASTER OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

SAIKUMAR SAMBASHIVARAO KOLIPAKA

TABLE OF CONTENTS

CHAPTER 1 : INTRODUCTION

1.1 Background.

Artificial intelligence (AI) is the branch of computer science that enables machines to learn, reason, and problem solve.
In recent decades, AI has been developed with the aim of improving the management of patients with brain tumours.
This review article explores the role AI currently plays in managing patients undergoing brain tumour surgery, and explores how AI may impact this field in the future.

1.2 Problem Statement.

In general, the brain is connected to whole senses, organs, blood, and nerval cells in the human's body. Tumors can affect the brain, which concurrently and seamlessly destroys cells in that area causing a brain-damage.
Any damage to the brain will affect the normal and typical situation of the body so that the body enters a state of instability.
If the person is lucky to discover the tumor early, the medical role comes in the surgical procedure.
It is important for the tumor to be detected early.
Tumors are detected through established medical procedures, such as MRI images and are taken for tests.
What interests us is the images taken by MRI, because we will be feeding our model on it.
MRI produces cross-sectional images of the brain, and each image is analyzed separately to determine the location of the tumor.

1.3 Purpose, Scope and Applicability.

1.3.1 Purpose

The main aim of our project is to help doctors to improve speed and accuracy of detecting and localizing the brain tumor based on MRI scan. This would drastically reduce Tumor Diagnosis and help in early diagnosis of tumor which would essentially be a life saver.

1.3.2 Scope

1. Deep learning has been proven to be superior in detecting diseases from X-rays, MRI scansand CT scans which could significantly improve the speed and accuracy of diagnosis.

2. Time consumption is less for detecting and localizing tumor.
3. Our project is going to set layered deep learning pipeline to perform classification andsegmentation.

1.3.3 Applicability

1. To achieve more accuracy for detecting the tumor by updating the models.
2. To develop a web based application to provide it to doctors for their betterment.

1.4 Objective.

Deep learning has been proven to be superior in detecting diseases from X-rays, MRI scans and CT scans which could significantly improve the speed and accuracy of diagnosis.

Time consumption is less for detecting and localizing tumor.

Our project is going to set layered deep learning pipeline to perform classification and segmentation.

Some limitations remain as challenges such as the differentiation of brain tumors from simulating lesions, tumor characterization and grading, and the discrimination of recurrent tumors from tissue necrosis.

In recent years, this has culminated in a large rise in publications on AI in radiology, regarding a vast variety of potential applications including identification, segmentation, classification, and outcome prediction.

CHAPTER 2 : SURVEY OF TECHNOLOGIES

2.1 Introduction.

In today's world Artificial Intelligence is revolutionizing Healthcare in many areas such as: Disease Diagnosis with medical imaging, Surgical Robots and Maximizing Hospital Efficiency. Deep learning has been proven to be superior in detecting diseases from X-rays, MRI scans and CT scans which could significantly improve the speed and accuracy of diagnosis. The MRI Scans are collected and the model is developed. This would drastically reduce the cost of cancer diagnosis & help in early diagnosis of tumors which would essentially be a life saver

2.2 Literature Review

A tumor is a collection of aberrant cells that form a tissue. These aberrant cells consume regular body cells, kill them, and continue to expand in size.

Brain tumor is one of these tumors. Nerve cells, brain cells, glands, and membranes thatsurround the brain are all affected. Imaging and pathology can both be used to diagnose a tumor.

An MRI (Magnetic Resonance Imaging) scan of a brain tumor produces a cross-section picture of the brain. In this report, we provide two models based on Artificial Convolutional Neural Networks that uses mathematical formulae and algebraic operations to analyze and predict if the tumor exists and to localize the area of tumor on MRI Images.

Artificial Intelligence has a role in the grading, prediction of treatment response, and prognosis of gliomas and the assessment of extra-axial brain tumors and pediatric tumors.

2.3 Comparative Analysis

Comparing and analysing the methods and techniques implemented in this project with many of the previous work done on this or related topic gives various insights and results thathelped us to improve on the methods implemented on this project

Computers are now being trained to "see" the patterns of disease often hidden in our

cells and tissues. Now comes word of yet another remarkable use of computer-generated artificial intelligence (AI): swiftly providing neurosurgeons with valuable, real-time information about what type of brain tumor is present, while the patient is still on the operating table.

2.4 Research Gap

As far as our project goes, comparing it with various papers and studies done online people and domain experts have been working on these kinds of methods with different applications have covered the various technological gap but the idea that we are implementingthese techniques on generally shows a great research gap as there have been no instances where resource finding used these techniques to ease the whole process of searching the resources.

During past years many researchers have given contribution in this field but there is still a need to do the research on tumor Diagnosis on fundamental and technical analysis,this provides strong motivation for a system that can efficiently extract data from different web sources can be done using the modules, which would be helpful for individual for selection of brain tumor images.

Research gaps and challenges between existing techniques are listed and detailed, which helps researchers to upgrade future works. Here, this paper provides an overview of the applications of Brain Tumor Diagnosis forecasting to determine what can be done in the future.

CHAPTER 3 : REQUIREMENTS AND ANALYSIS

3.1 Problem Definition.

In general, the brain is connected to whole senses, organs, blood, and nerval cells in the human'sbody. Tumors can affect the brain, which concurrently and seamlessly destroys cells in that area causing a brain-damage.

Any damage to the brain will affect the normal and typical situation of the body so that the bodyenters a state of instability.

If the person is lucky to discover the tumor early, the medical role comes in the surgical procedure. It is important for the tumor to be detected early.

Tumors are detected through established medical procedures, such as MRI images and are taken for tests. What interests us is the images taken by MRI, because we will be feeding our modelon it.

MRI produces cross-sectional images of the brain, and each image is analyzed separately to determine the location of the tumor.

3.2 Requirements Specification / Analysis.

Based on the objectives as well as the user requirements, this section outlines the various requirements to be met in the research.

a. Functional Requirements

These are functions or processes the proposed system and its components must perform.They are a definition of what users of the system expect form it. For the system, the functional requirements include :-

- The system should allow a user to select the images from csv file.

b. Non Functional Requirements

Unlike the functional requirements, non-functional requirements place constraints or limits in how the proposed system will achieve its functional requirements. They describe howwell the system does its functions and are classified based on the needs of the users. The non-functional requirements of the system include :-

- Usability- The interaction with the system will be simple to allow show better results.
- Reliability - The reliability of the model will highly depend on the accuracy of the data collected . As this data will be used to train the model which will be used in prediction.
- Interoperability – This is the degree to which the developed system will be able to facilitate of couple the different interfaces with other systems.
- Scalability – This describes the degree in which the system is able to expand its processing or functional capabilities outward or upward with the aim of supporting growth and user requirements.
- Persistent storage- the proposed system components and devices should be able to retain data or information after device's power have been shut down or eliminated.

3.3 Planning and Scheduling

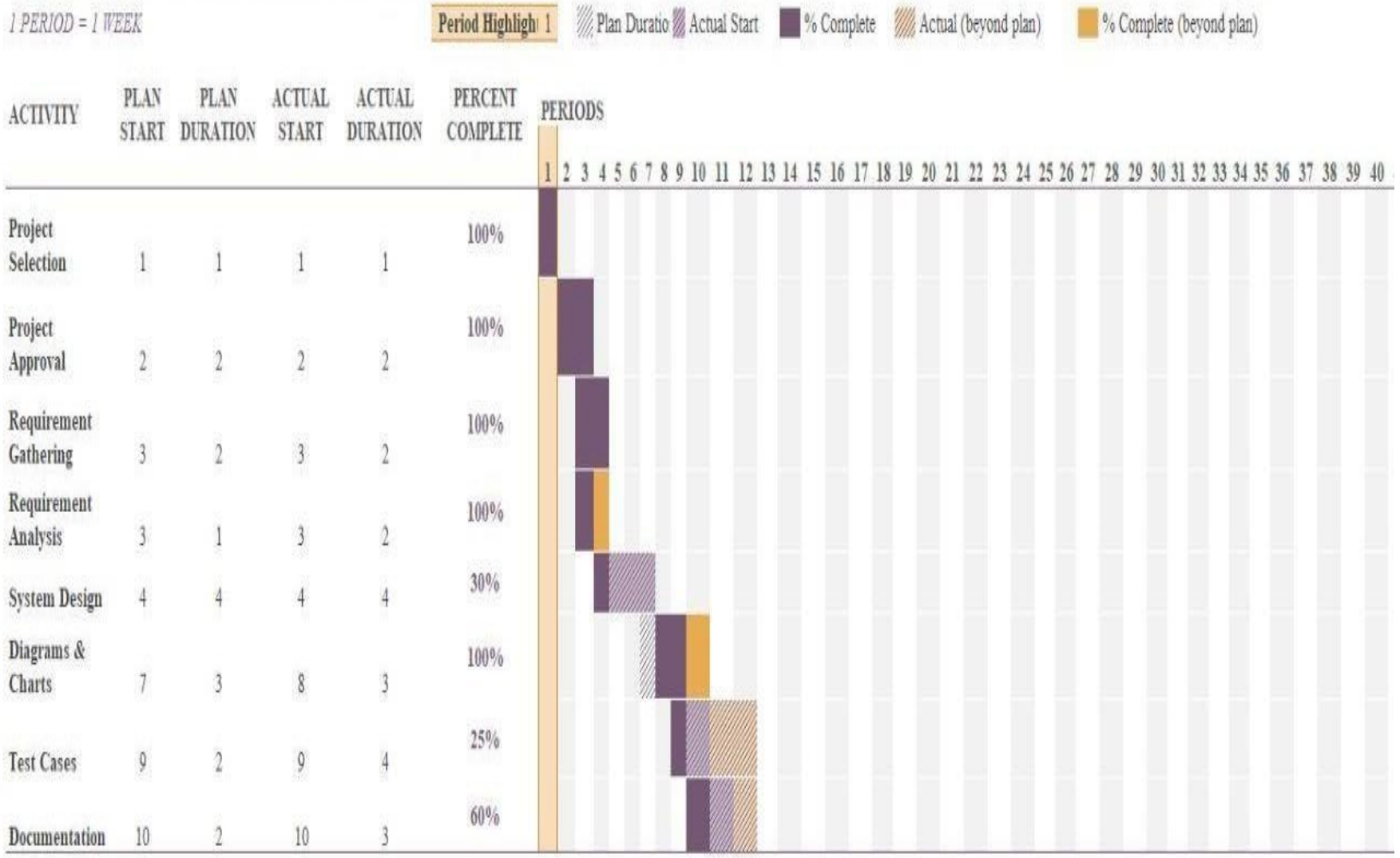
3.3.1 GANTT CHART:-

A Gantt chart makes it simple to create, view and monitor project activities and tasks over a given time. Typically, each activity, process or task in a Gantt chart is represented by ahorizontal bar scaled parallel to a calendar and/or dates.

The start and end of each bar represent the start and the end of that particular activity.Gantt charts help in quickly understanding the different tasks in a project, their schedule (startand end date) and any overlapping tasks or activities.

A Gantt chart is constructed with a horizontal axis representing the total time span of the project, broken down into increments (for example, days, weeks, or months) and a verticalaxis representing the tasks that make up the project (for example, if the project is outfitting your computer with new software, the major tasks involved might be: conduct research, choosesoftware, install software).

ARTIFICIAL INTELLIGENCE ASSISTED BRAIN TUMOR DIAGNOSIS



ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Project Selection	1	1	1	1	100%
Project Approval	2	2	2	2	100%
Requirement Gathering	3	2	3	2	100%
Requirement Analysis	3	1	3	2	100%
System Design	4	4	4	4	30%
Diagrams & Charts	7	3	8	3	100%
Test Cases	9	2	9	4	25%
Documentation	10	2	10	3	60%

PERIODS

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40

3.4 Hardware and Software Requirements:-

a) Hardware Requirements

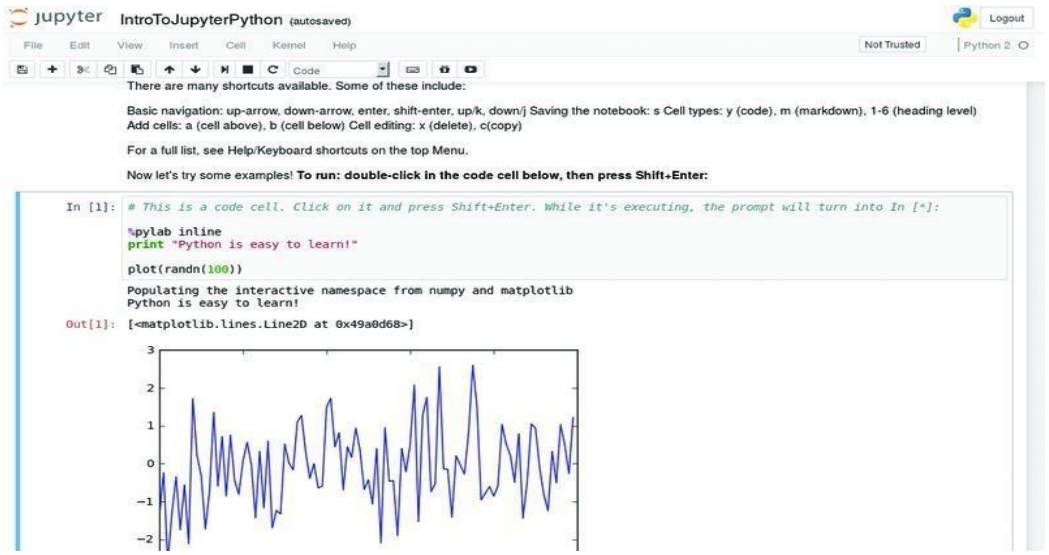
Windows 10/11 :- For using an Jupyter Notebook, you'll most likely need the 64- bit version of Windows 10.

- **Processor :-** Intel i5 to Intel i9. The Intel processor should have support for 2 GHZ or faster CPU.
- **Graphics Processor:** AMD Radeon Integrated Graphics or NVIDIA Graphic Card upto 5GB or more
- **Memory :-** You'll need at least 5 GB RAM. For some, the minimum memory requirement may be higher. It's important to note that 4 GB of disk storage would not make up for memory as that's a requirement.
- **Storage :-** Every emulator has different storage requirements, but you should keep at least 15 GB in mind. It's always better to have more space than required.

b) Software Requirements

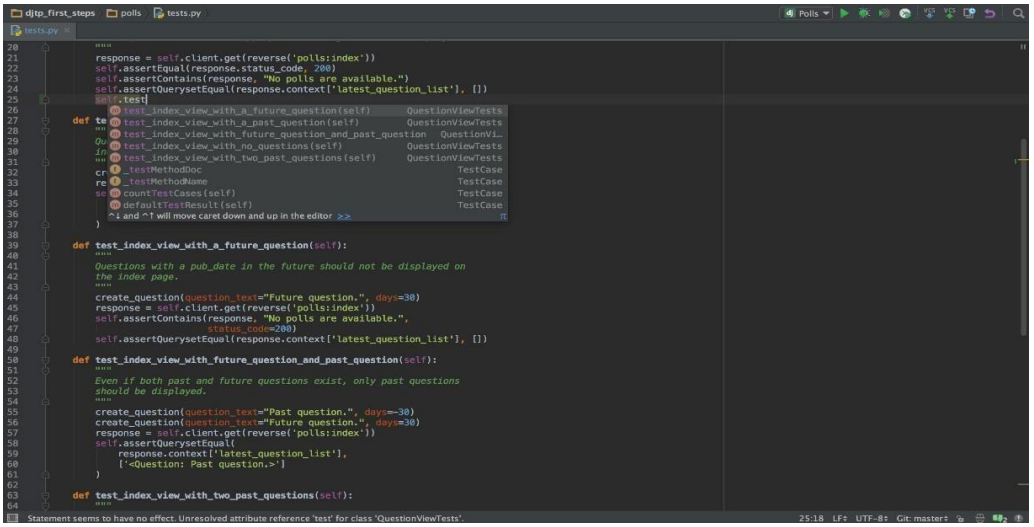
i. Jupyter Notebook

Jupyter Notebook is an open-source web application that allows us to create and share codesand documents. It provides an environment, where you can document your code, run it, lookat the outcome, visualize data and see the results without leaving the environment.



ii. Pycharm

Pycharm provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as data science with Anaconda. PyCharm is cross-platform, with Windows, macOS and Linux versions.



CHAPTER 4 : SYSTEM DESIGN

4.1 Basic Modules.

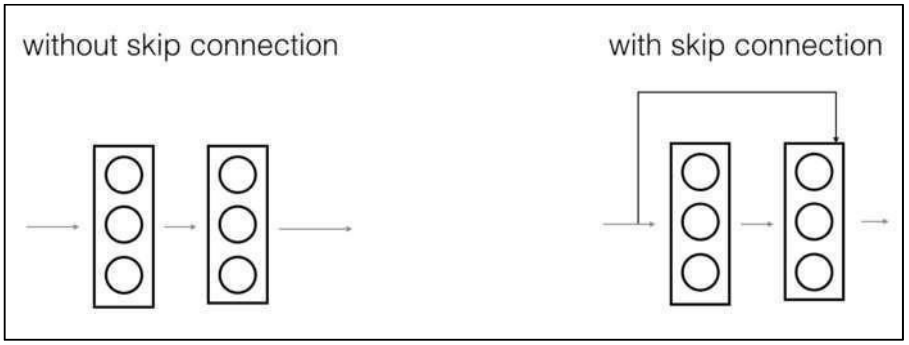
Resnet Model:-

ResNet, short for Residual Networks is a classic neural network used as a backbone for many computer vision tasks.The fundamental breakthrough with ResNet was it allowed us to train extremely deep neural networks with 150+layers successfully.

However, increasing network depth does not work by simply stacking layers together. Deep networks are hard to train because of the notorious vanishing gradient problem as the gradient are back-propagated to earlier layers, repeated multiplication may make the gradient extremely small. As a result, as the network goes deeper, its performance gets saturated or even starts degrading rapidly.

Skip Connection: The Strength of ResNet:

ResNet first introduced the concept of skip connection. The diagram below illustrates skip connection. The figure on the left is stacking convolution layers together one after the other.On the right we still stack convolution layers as before but we now also add the original inputto the output of the convolution block. This is called skip connection.



ResNet uses skip connection to add the output from an earlier layer to a later layer. This helps it mitigate the vanishing gradient problem.

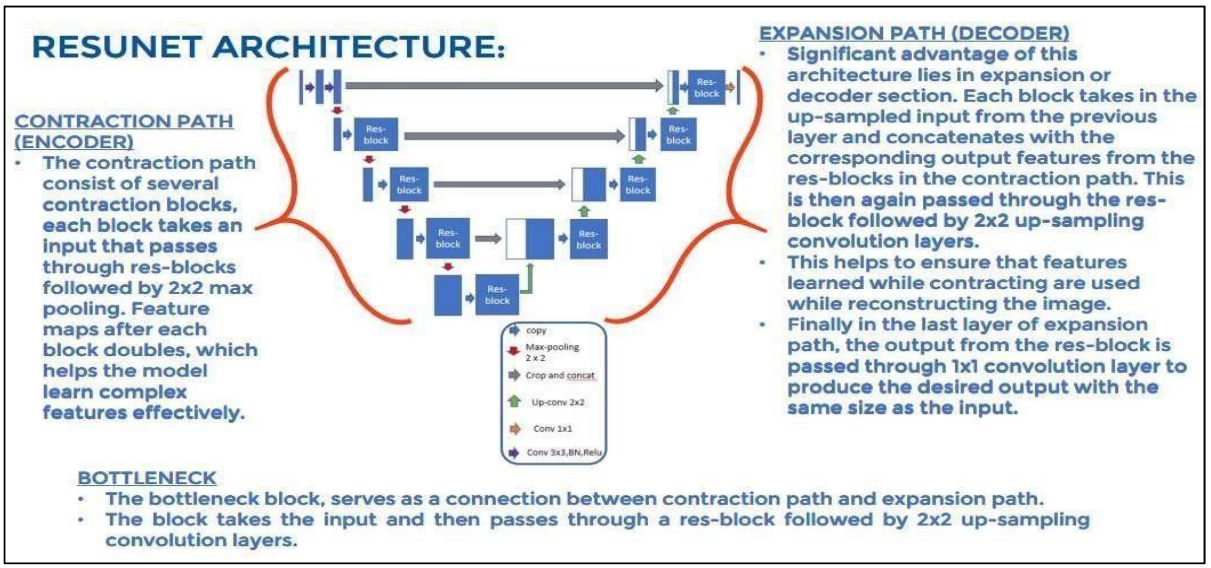
Resunet Model;-

ResuNet architecture combines UNet backbone architecture with residual blocks to overcome the vanishing gradients problems present in deep architectures.

Unet architecture is based on Fully Convolutional Networks and modified in a way that it performs well on segmentation tasks.

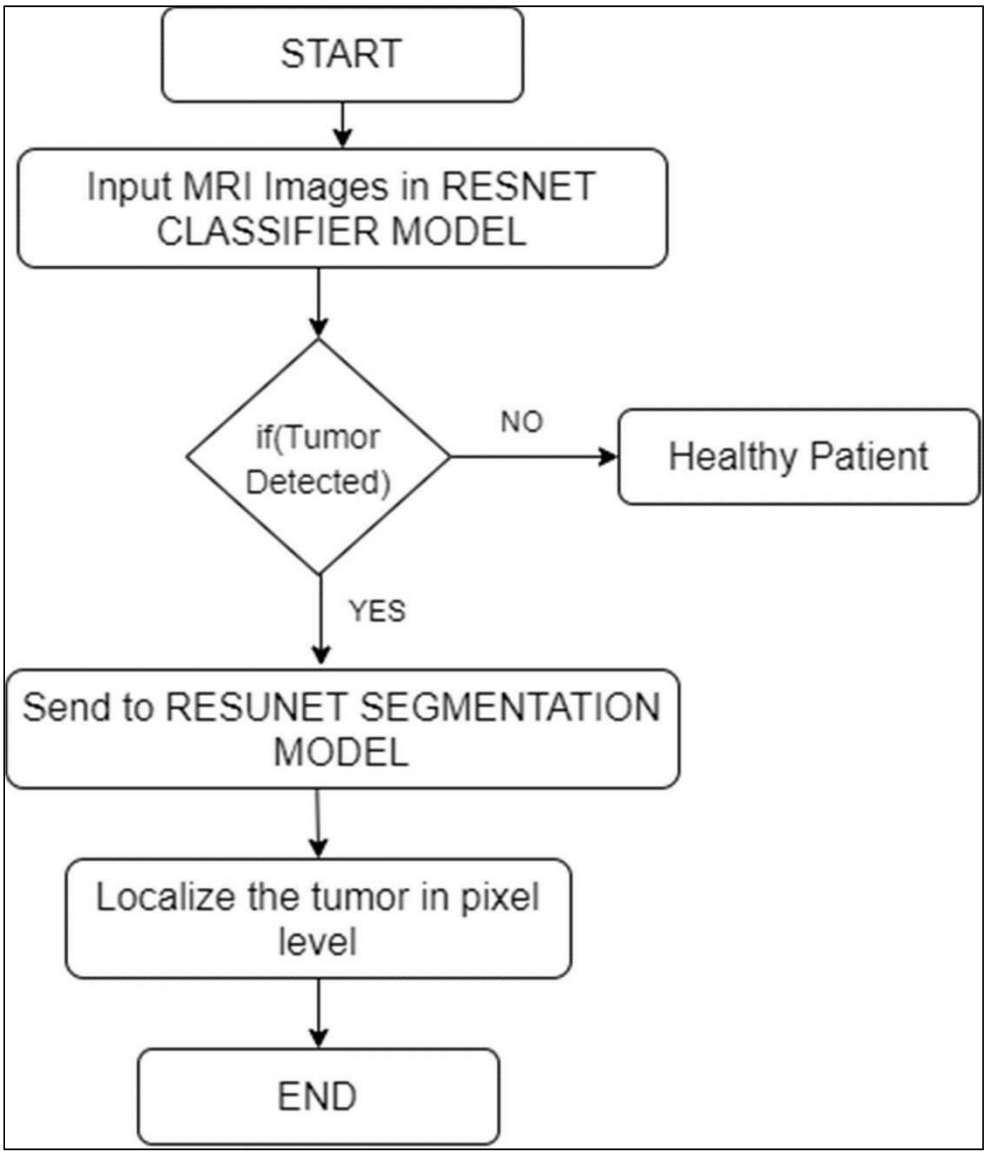
The RESUNET consists of an encoding network, decoding network and a bridge connecting both these networks, just like a U-Net.

The U-Net uses two 3 x 3 convolution, where each is followed by a ReLU activation function. In the case of RESUNET, these layers are replaced by a pre-activated residual block.



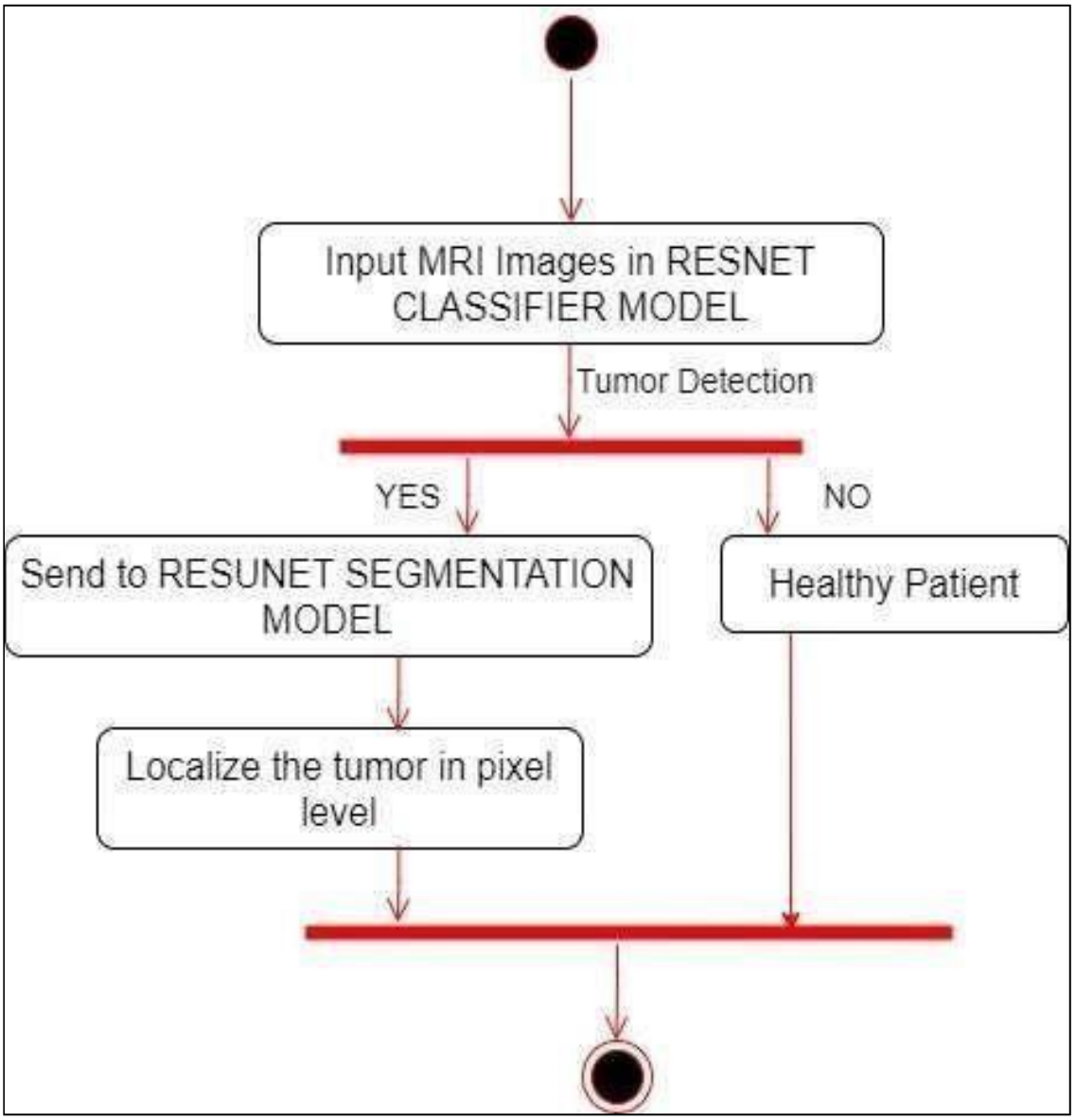
4.2 Diagrams.

4.2.1 Flowchart Diagram



A flowchart is a picture of the separate steps of a process in sequential order. It is a generic tool that can be adapted for a wide variety of purposes, and can be used to describe various processes, such as a manufacturing process, an administrative or service process.

4.2.2 Activity Diagram.



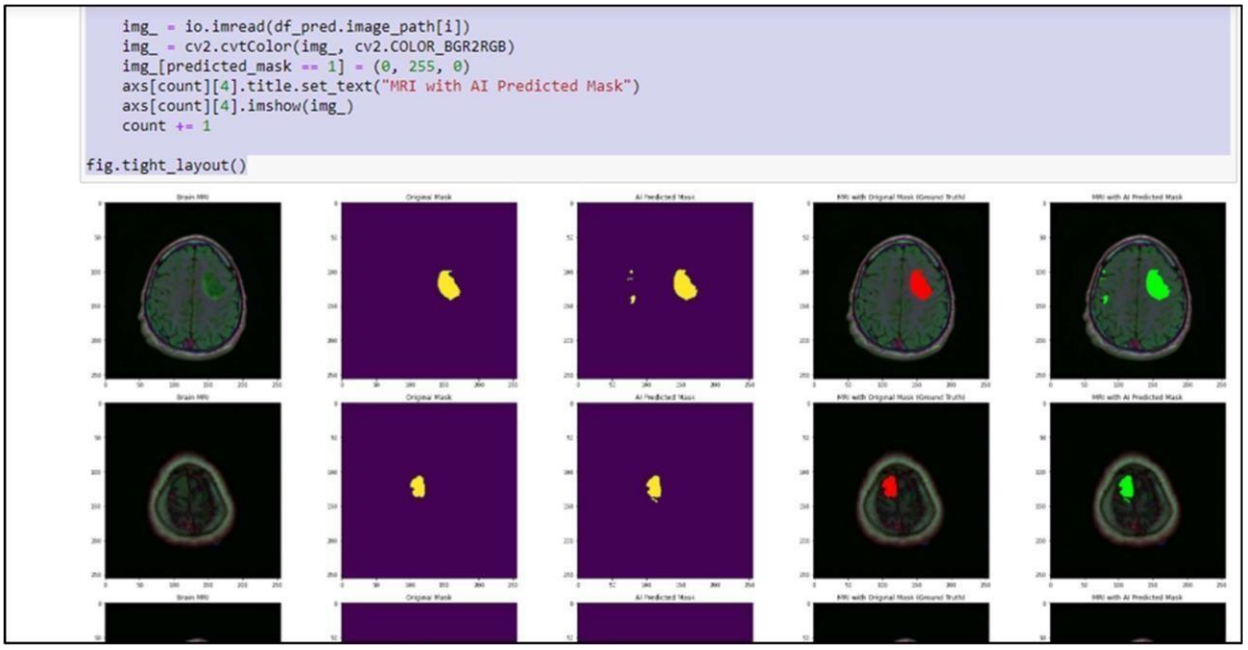
Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent.

4.3 Test Cases Design:-

Test Case 1(Positive):-

If all operations are performed properly and all the data which is trained works within a givenformat, the output generates AI generated mask images

OUTPUT:-



Test Case 2 (Negative):

All the further process depend upon the training of both models that are ResNet and ResUnet models.

If the training of the model is not completed as to be expected it cannot be considered, andfurther process (prediction of brain tumor mask) cannot take place.

Test Case 3 (Tentative):

The AI predicted mask can also show some false negative outputs which are not generallypresent.

CHAPTER 5 : IMPLEMENTATION OF CODE WITH IMAGES

5.1 Coding Details.

PART I – RESNET CLASSIFIER MODEL

1. Python Libraries used in our Project

```
import pandas as pd #--
import numpy as np #--
import cv2 #opencv-python #--
import matplotlib.pyplot as plt #--
import random
from skimage import io #Image Processing
import plotly.graph_objects as go
from tensorflow.keras.layers import *
from tensorflow.keras.models import Model, load_model
from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.keras.callbacks import ReduceLROnPlateau, EarlyStopping,
ModelCheckpoint, LearningRateScheduler
from sklearn.metrics import confusion_matrix
import seaborn as sns
from sklearn.metrics import accuracy_score
import tensorflow as tf
from keras_preprocessing.image import ImageDataGenerator
from sklearn.model_selection import train_test_split
```

2. Displaying the data set using panadas library

```
brain_df = pd.read_csv('data_mask.csv')

brain_df
```

3. Train Test Split

```
train, test = train_test_split(brain_df_train, test_size = 0.15)
```

4. ImageDataGenerator

```
datagen = ImageDataGenerator(rescale=1./255., validation_split = 0.15)

# Create a data generator for test images

test_datagen=ImageDataGenerator(rescale=1./255.)
```

5. Flow from Dataframe

```
train_generator=datagen.flow_from_dataframe(

dataframe=train,

directory= './',

x_col='image_path',

y_col='mask',

subset="training",

21

batch_size=16,

shuffle=True,

class_mode="categorical",

target_size=(256,256))

valid_generator=datagen.flow_from_dataframe(

dataframe=train,
```

```
directory= './',
x_col='image_path',
y_col='mask',
subset="validation",
batch_size=16,
shuffle=True,
class_mode="categorical",
target_size=(256,256))
test_generator=test_datagen.flow_from_dataframe(
dataframe=test,
directory= './',
x_col='image_path',
y_col='mask',
batch_size=16,
shuffle=False,
class_mode='categorical',
target_size=(256,256))
```

6. ResNet50 as Base Model from tensorflow

```
basemodel = ResNet50(weights = 'imagenet', include_top = False, input_tensor =
Input(shape=(256, 256, 3)))
```

7. Freezing the base model

```
for layer in basemodel.layers:
layer.trainable = False
```

8. Transfer Learning on base model

```
headmodel = basemodel.output
headmodel = AveragePooling2D(pool_size = (4,4))(headmodel)
headmodel = Flatten(name= 'flatten')(headmodel)
headmodel = Dense(256, activation = "relu")(headmodel)
headmodel = Dropout(0.3)(headmodel)
headmodel = Dense(256, activation = "relu")(headmodel)
headmodel = Dropout(0.3)(headmodel)
# headmodel = Dense(256, activation = "relu")(headmodel)
# headmodel = Dropout(0.3)(headmodel)
headmodel = Dense(256, activation = "relu")(headmodel)
```

22

```
headmodel = Dropout(0.3)(headmodel)
headmodel = Dense(2, activation = 'softmax')(headmodel)
model = Model(inputs = basemodel.input, outputs = headmodel)
```

9. Using Early Stopping

```
earlystopping = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=20)
# save the best model with least validation loss
checkpointer = ModelCheckpoint(filepath="classifier-resnet-weights.hdf5", verbose=1,
save_best_only=True)
```

10. Loading pretrained model

```
with open('resnet-50-MRI.json', 'r') as json_file:
json_savedModel= json_file.read()
```

```
# load the model

model = tf.keras.models.model_from_json(json_savedModel)

model.load_weights('weights.hdf5')

model.compile(loss = 'categorical_crossentropy', optimizer='adam', metrics= ["accuracy"])
```

11. Testing the model on train dataset

```
test_predict = model.predict(test_generator, steps = test_generator.n // 16, verbose =1)
```

12. Obtaining Accuracy of the model

```
# Obtain the accuracy of the model

accuracy = accuracy_score(original, predict)

accuracy
```

13. Plotting Confusion Matrix

```
# plot the confusion matrix

cm = confusion_matrix(original, predict)

plt.figure(figsize = (7,7))

sns.heatmap(cm, annot=True)
```

14. Output

Displaying the data set using pandas library

	patient_id	image_path	mask_path	mask
0	TCGA_CS_5395_19981004	TCGA_CS_5395_19981004/TCGA_CS_5395_19981004_1.tif	TCGA_CS_5395_19981004/TCGA_CS_5395_19981004_1_...	0
1	TCGA_CS_5395_19981004	TCGA_CS_4944_20010208/TCGA_CS_4944_20010208_1.tif	TCGA_CS_4944_20010208/TCGA_CS_4944_20010208_1_...	0
2	TCGA_CS_5395_19981004	TCGA_CS_4941_19960909/TCGA_CS_4941_19960909_1.tif	TCGA_CS_4941_19960909/TCGA_CS_4941_19960909_1_...	0
3	TCGA_CS_5395_19981004	TCGA_CS_4943_20000902/TCGA_CS_4943_20000902_1.tif	TCGA_CS_4943_20000902/TCGA_CS_4943_20000902_1_...	0
4	TCGA_CS_5395_19981004	TCGA_CS_5396_20010302/TCGA_CS_5396_20010302_1.tif	TCGA_CS_5396_20010302/TCGA_CS_5396_20010302_1_...	0
...	...	...	...	...
3924	TCGA_DU_6401_19831001	TCGA_HT_A61B_19991127/TCGA_HT_A61B_19991127_86...	TCGA_HT_A61B_19991127/TCGA_HT_A61B_19991127_86...	0
3925	TCGA_DU_6401_19831001	TCGA_HT_A61A_20000127/TCGA_HT_A61A_20000127_87...	TCGA_HT_A61A_20000127/TCGA_HT_A61A_20000127_87...	0
3926	TCGA_DU_6401_19831001	TCGA_HT_A61B_19991127/TCGA_HT_A61B_19991127_87...	TCGA_HT_A61B_19991127/TCGA_HT_A61B_19991127_87...	0
3927	TCGA_DU_6401_19831001	TCGA_HT_A61A_20000127/TCGA_HT_A61A_20000127_88...	TCGA_HT_A61A_20000127/TCGA_HT_A61A_20000127_88...	0
3928	TCGA_DU_6401_19831001	TCGA_HT_A61B_19991127/TCGA_HT_A61B_19991127_88...	TCGA_HT_A61B_19991127/TCGA_HT_A61B_19991127_88...	0

Resnet50 base model from TensorFlow

```
In [13]: basemodel = ResNet50(weights = 'imagenet', include_top = False, input_tensor = Input(shape=(256, 256, 3)))

In [14]: basemodel.summary()
```

```
Model: "resnet50"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            [(None, 256, 256, 3) 0
__________________________________________________________________________________________________
conv1_pad (ZeroPadding2D)       (None, 262, 262, 3)  0           input_1[0][0]
__________________________________________________________________________________________________
conv1_conv (Conv2D)             (None, 128, 128, 64) 9472        conv1_pad[0][0]
__________________________________________________________________________________________________
conv1_bn (BatchNormalization)   (None, 128, 128, 64) 256         conv1_conv[0][0]
__________________________________________________________________________________________________
conv1_relu (Activation)         (None, 128, 128, 64) 0           conv1_bn[0][0]
__________________________________________________________________________________________________
pool1_pad (ZeroPadding2D)       (None, 130, 130, 64) 0           conv1_relu[0][0]
__________________________________________________________________________________________________
pool1_pool (MaxPooling2D)       (None, 64, 64, 64)   0           pool1_pad[0][0]
__________________________________________________________________________________________________
conv2_block1_1_conv (Conv2D)    (None, 64, 64, 64)   4160        pool1_pool[0][0]
```

Transfer Learning on Resnet50 model

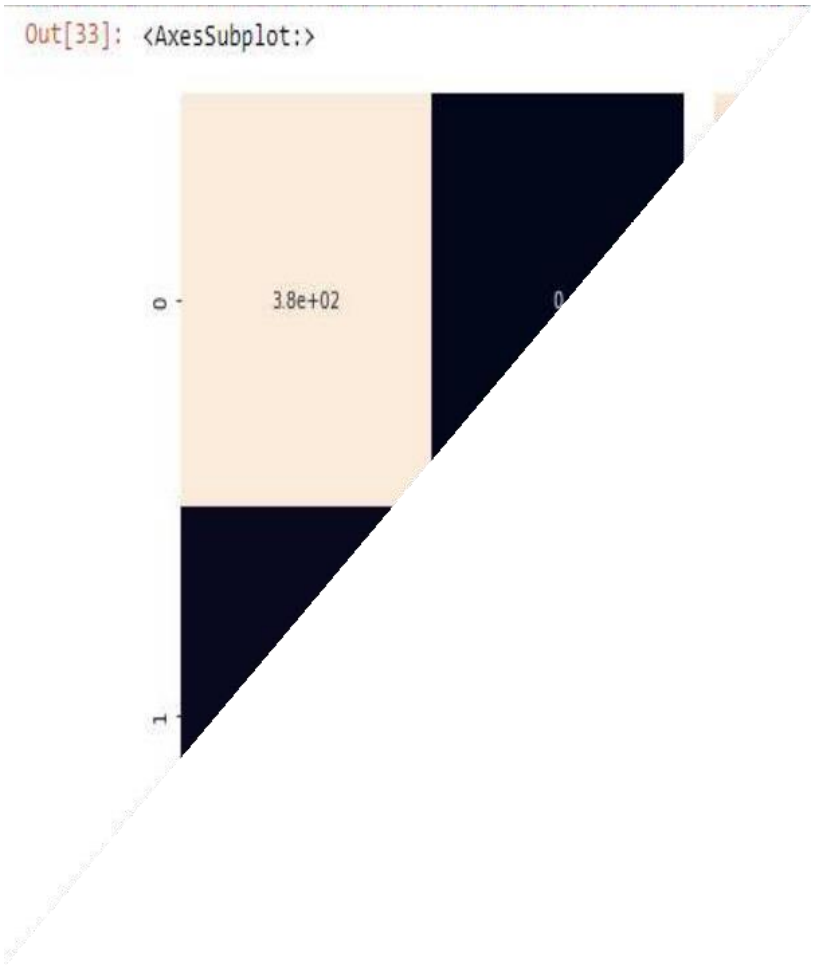
Showing the predicted class on test data set

```
In [28]: predict

Out[28]: array(['0', '1', '0', '0', '0', '0', '1', '1', '0', '0', '0', '1', '0',
                '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '1', '0', '0',
                '1', '0', '0', '0', '0', '0', '0', '0', '0', '0', '1', '0', '0',
                '0', '0', '0', '0', '1', '1', '1', '1', '0', '0', '0', '1', '0',
                '0', '0', '0', '1', '1', '0', '0', '0', '1', '1', '0', '0', '0',
                '0', '0', '0', '0', '0', '0', '1', '1', '0', '0', '1', '0', '0',
                '1', '1', '0', '0', '0', '0', '1', '0', '0', '0', '0', '0', '0',
                '1', '1', '0', '1', '0', '1', '0', '0', '0', '0', '1', '0', '0',
                '1', '0', '1', '0', '0', '0', '0', '1', '0', '0', '0', '0', '0',
                '0', '0', '1', '0', '0', '0', '0', '1', '0', '1', '0', '0', '1',
                '0', '0', '1', '0', '0', '1', '0', '0', '1', '1', '0', '0', '0',
                '0', '0', '1', '0', '0', '1', '0', '0', '0', '0', '1', '1', '0',
                '0', '0', '1', '0', '0', '0', '1', '1', '1', '1', '0', '0', '0',
                '0', '1', '0', '0', '0', '0', '0', '0', '0', '1', '0', '1', '1',
                '1', '0', '0', '0', '0', '1', '1', '1', '1', '0', '0', '0', '1',
                '0', '0', '0', '0', '1', '1', '1', '0', '0', '0', '1', '1', '1',
                '0', '0', '0', '0', '0', '1', '0', '0', '0', '0', '0', '0', '0',
                '1', '0', '0', '0', '1', '1', '0', '0', '1', '0', '1', '1', '0',
                '0', '0', '1', '0', '0', '0', '0', '0', '1', '0', '0', '0', '0',
                '1', '0', '1', '1', '0', '0', '0', '1', '0', '0', '0', '0', '0',
                '0', '0', '1', '1', '0', '1', '0', '0', '0', '0', '1', '1', '0',
                '1', '0', '0', '0', '0', '0', '0', '1', '0', '1', '1', '1', '1',
                '0', '0', '0', '0', '0', '0', '0', '0', '0', '1', '1', '1', '1',
                '0', '0', '0', '0', '1', '0', '0', '0', '0', '1', '1', '0', '0',
                '1', '1', '0', '0', '0', '1', '1', '0', '1', '1', '0', '1', '0',
                '1', '0', '0', '0', '1', '0', '1', '1', '1', '1', '0', '1', '0',
```

Showing the obtained accuracy by the model

```
In [30]: # Obtain the accuracy of the model
         accuracy = accuracy_score(original, predict)
         accuracy

Out[30]: 0.9861111111111112
```

Showing the obtained accuracy by the model



Plotting the confusion matrix on the test data set

PART II – RESUNET SEGMENTATION MODEL

15. Get the data frame containing MRIs which have masks associated with them.

```
brain_df_mask = brain_df[brain_df['mask'] == 1]
brain_df_mask.shape
```

16. Spliting the data into train and test data.

```
from sklearn.model_selection import train_test_split
X_train, X_val = train_test_split(brain_df_mask, test_size=0.15)
X_test, X_val = train_test_split(X_val, test_size=0.5)
```

17. Creating separate list for imageId, classId to pass into the generator.

```
train_ids = list(X_train.image_path)
train_mask = list(X_train.mask_path)
val_ids = list(X_val.image_path)
val_mask= list(X_val.mask_path)
```

18. Data Generator.

```
from utilities import DataGenerator
# create image generators
training_generator = DataGenerator(train_ids,train_mask)
validation_generator = DataGenerator(val_ids,val_mask)
```

19. Defining function for resblock and upsampling (function to upscale and concatenate the values passsed).

```
def resblock(X, f):
# make a copy of input
```

```
X_copy = X
# main path
X = Conv2D(f, kernel_size = (1,1) ,strides = (1,1),kernel_initializer ='he_normal')(X)
X = BatchNormalization()(X)
X = Activation('relu')(X)
X = Conv2D(f, kernel_size = (3,3), strides =(1,1), padding = 'same', kernel_initializer
='he_normal')(X)
X = BatchNormalization()(X)
# Short path
X_copy = Conv2D(f, kernel_size = (1,1), strides =(1,1), kernel_initializer
='he_normal')(X_copy)
X_copy = BatchNormalization()(X_copy)
# Adding the output from main path and short path together
X = Add()([X,X_copy])
X = Activation('relu')(X)
return X
def upsample_concat(x, skip):
x = UpSampling2D((2,2))(x)
merge = Concatenate()([x, skip])
return merge
input_shape = (256,256,3)
# Input tensor shape
X_input = Input(input_shape)
```

```
# Stage 1
conv1_in = Conv2D(16,3,activation= 'relu', padding = 'same', kernel_initializer
='he_normal')(X_input)
conv1_in = BatchNormalization()(conv1_in)
conv1_in = Conv2D(16,3,activation= 'relu', padding = 'same', kernel_initializer
='he_normal')(conv1_in)
conv1_in = BatchNormalization()(conv1_in)
pool_1 = MaxPool2D(pool_size = (2,2))(conv1_in)
# Stage 2
conv2_in = resblock(pool_1, 32)
pool_2 = MaxPool2D(pool_size = (2,2))(conv2_in)
# Stage 3
conv3_in = resblock(pool_2, 64)
pool_3 = MaxPool2D(pool_size = (2,2))(conv3_in)
# Stage 4
conv4_in = resblock(pool_3, 128)
pool_4 = MaxPool2D(pool_size = (2,2))(conv4_in)
# Stage 5 (Bottle Neck)
conv5_in = resblock(pool_4, 256)
# Upscale stage 1
up_1 = upsample_concat(conv5_in, conv4_in)
up_1 = resblock(up_1, 128)
# Upscale stage 2
```

```
up_2 = upsample_concat(up_1, conv3_in)
up_2 = resblock(up_2, 64)
# Upscale stage 3
up_3 = upsample_concat(up_2, conv2_in)
up_3 = resblock(up_3, 32)
# Upscale stage 4
up_4 = upsample_concat(up_3, conv1_in)
up_4 = resblock(up_4, 16)
# Final Output
output = Conv2D(1, (1,1), padding = "same", activation = "sigmoid")(up_4)
model_seg = Model(inputs = X_input, outputs = output )
```

20. EarlyStopping

```
earlystopping = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=20)
checkpointer = ModelCheckpoint(filepath="ResUNet-weights.hdf5", verbose=1,
save_best_only=True)
```

21. Creating Dataframe for result

```
df_pred = pd.DataFrame({'image_path': image_id,'predicted_mask': mask,'has_mask':
has_mask})
df_pred
```

22. Merging the dataframe containing predicted results with the original test data.

```
df_pred = test.merge(df_pred, on = 'image_path')
df_pred.head()
```

23. Displaying the dataset with the output generated by the ResUNet Model

```
count = 0
fig, axs = plt.subplots(10, 5, figsize=(30, 50))
for i in range(len(df_pred)):
if df_pred['has_mask'][i] == 1 and count < 10:
# read the images and convert them to RGB format
img = io.imread(df_pred.image_path[i])
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
axs[count][0].title.set_text("Brain MRI")
axs[count][0].imshow(img)
# Obtain the mask for the image
mask = io.imread(df_pred.mask_path[i])
axs[count][1].title.set_text("Original Mask")
axs[count][1].imshow(mask)
# Obtain the predicted mask for the image
predicted_mask = np.asarray(df_pred.predicted_mask[i])[0].squeeze().round()
axs[count][2].title.set_text("AI Predicted Mask")
axs[count][2].imshow(predicted_mask)
# Apply the mask to the image 'mask==255'
img[mask == 255] = (255, 0, 0)
axs[count][3].title.set_text("MRI with Original Mask (Ground Truth)")
axs[count][3].imshow(img)
img_ = io.imread(df_pred.image_path[i])
```

```
img_ = cv2.cvtColor(img_, cv2.COLOR_BGR2RGB)

img_[predicted_mask == 1] = (0, 255, 0)

axs[count][4].title.set_text("MRI with AI Predicted Mask")

axs[count][4].imshow(img_)

count += 1

fig.tight_layout()
```

24. Output

ResUNet - Model summary

```
In [38]: model_seg.summary()
```

conv2d (Conv2D)	(None, 256, 256, 16)	448	input_2[0][0]
batch_normalization (BatchNorma	(None, 256, 256, 16)	64	conv2d[0][0]
conv2d_1 (Conv2D)	(None, 256, 256, 16)	2320	batch_normalization[0][0]
batch_normalization_1 (BatchNor	(None, 256, 256, 16)	64	conv2d_1[0][0]
max_pooling2d (MaxPooling2D)	(None, 128, 128, 16)	0	batch_normalization_1[0][0]
conv2d_2 (Conv2D)	(None, 128, 128, 32)	544	max_pooling2d[0][0]
batch_normalization_2 (BatchNor	(None, 128, 128, 32)	128	conv2d_2[0][0]
activation (Activation)	(None, 128, 128, 32)	0	batch_normalization_2[0][0]
conv2d_3 (Conv2D)	(None, 128, 128, 32)	9248	activation[0][0]
conv2d_4 (Conv2D)	(None, 128, 128, 32)	544	max_pooling2d[0][0]

Creating dataframe for the result

creating a dataframe for the result

```
In [45]: df_pred = pd.DataFrame({'image_path': image_id,'predicted_mask': mask,'has_mask': has_mask})
         df_pred
```

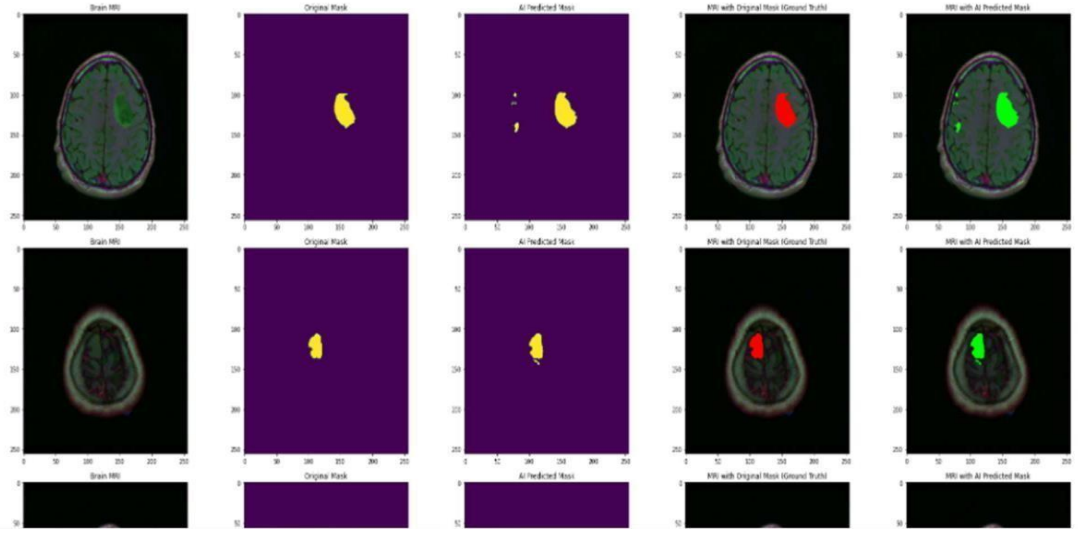
Out[45]:

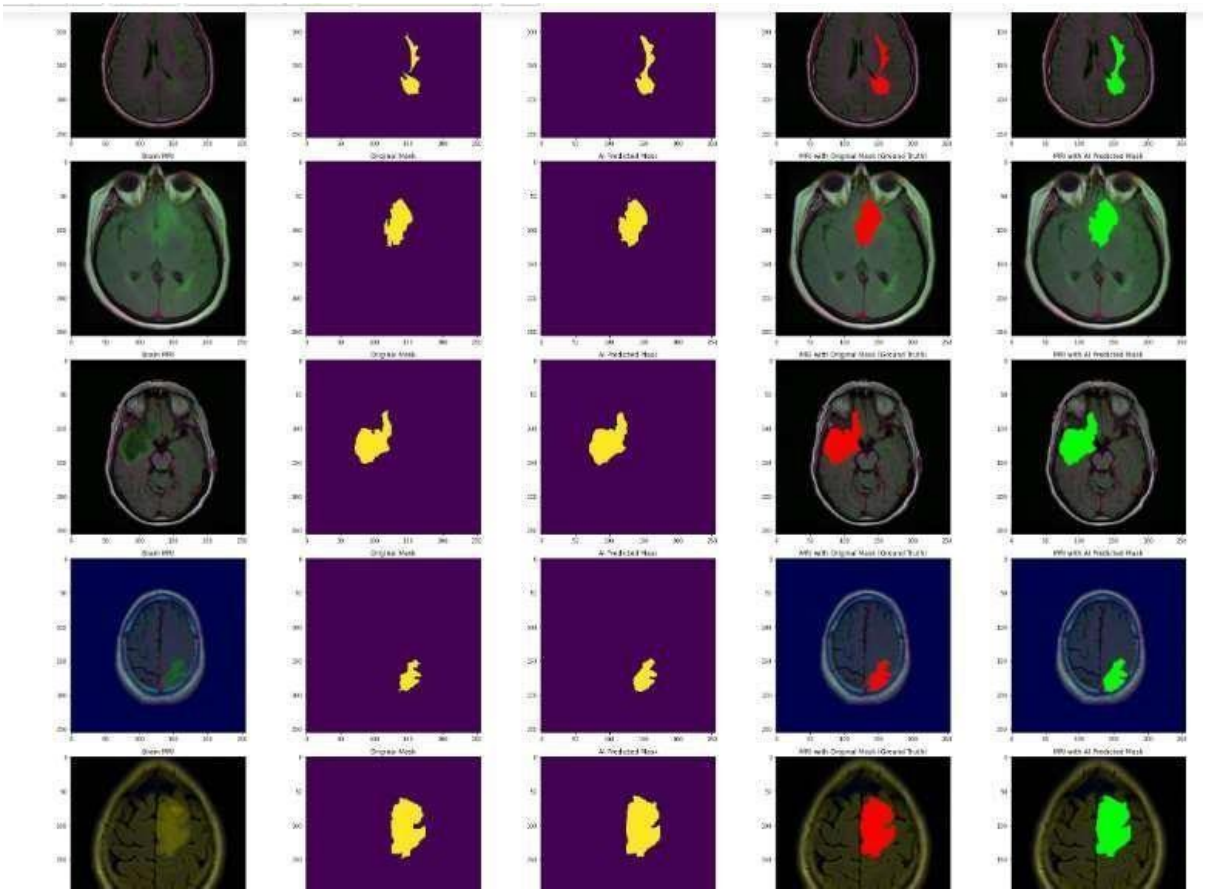
	image_path	predicted_mask	has_mask
0	TCGA_HT_8018_19970411/TCGA_HT_8018_19970411_3.tif	No mask	0
1	TCGA_DU_7309_19960831/TCGA_DU_7309_19960831_13...	No mask	0
2	TCGA_FG_5964_20010511/TCGA_FG_5964_20010511_4.tif	No mask	0
3	TCGA_FG_A60K_20040224/TCGA_FG_A60K_20040224_24...	No mask	0
4	TCGA_CS_4944_20010208/TCGA_CS_4944_20010208_17...	No mask	0
...	...	...	...
585	TCGA_DU_5852_19950709/TCGA_DU_5852_19950709_23...	No mask	0
586	TCGA_DU_5854_19951104/TCGA_DU_5854_19951104_26...	[[[[8.205896e-07], [2.6940888e-06], [5.185762e...	1
587	TCGA_DU_6405_19851005/TCGA_DU_6405_19851005_4.tif	No mask	0
588	TCGA_HT_7616_19940813/TCGA_HT_7616_19940813_3.tif	No mask	0
589	TCGA_DU_6401_19831001/TCGA_DU_6401_19831001_45...	No mask	0

590 rows × 3 columns

Merge the dataset containing predicted results

```
    img_ = io.imread(df_pred.image_path[i])
    img_ = cv2.cvtColor(img_, cv2.COLOR_BGR2RGB)
    img_[predicted_mask == 1] = (0, 255, 0)
    axs[count][4].title.set_text("MRI with AI Predicted Mask")
    axs[count][4].imshow(img_)
    count += 1

fig.tight_layout()
```





Predicted Result image 1

CHAPTER 6 : TEST CASES AND PLAN

6.1 Testing

1) Testing is the process or method of finding errors in a software application or program so that the application functions according to the end users requirements.
2) Testing should be started as early as possible to reduce the cost and time to rework and produce the software that is bug-free & can be delivered to the customer.
3) Testing ensures that the project is being completed in every term.

6.2 Test Plan

The aim of this phase is to test the whole project provides proper indication to the user about the parking space available, which includes all main features and some testing of software.

SR.NO	TEST NAME	TEST DESCRIPTION
1.	Normal Test	Consist in testing the features trying to cover all the requirements.
2.	Component Test	Consist in testing each component alone, before integrating it in the system.
3.	Random Test	Consist in selecting some random part to add more tests to normal test and to go in detailed.
4.	Faulty Test	Consist in trying to feed the system with invalid values and see how the system works.
5.	Integration Test	Consist in testing modules after their integration in the system.

CHAPTER 7 : FUTURE SCOPE AND CONCLUSION

7.1 Future Scope :-

This project can be improved by uploading it to a website where physicians and patients can access it for greater usage. More diverse types of anomalies can also be discovered by training the model with greater precision. Training time can also be decreased by introducing more powerful hardware resources, such as high-end processors and GPUs.

7.2 Conclusion :-

Tumors are fatal threat on humanity. A brain tumor effects the nervous system, blood system, organs function…etc. Many methods are available to detect the tumors. One method utilizes MRIs. A cross-sector images of the brain are produced from the MRI, and that helps us more in discovering more area from the brain. A dataset collected by us from various sources was used in our project. Total we had collected 3929 brain MRI Images. We designed an Resnet Classifier Model to classify MRI images as positive, label equal 1, (i.e. having tumor) or negative, label 0, we experimented with our classifier model with a test set and we came up with a test accuracy 98% by evaluation. Also plotted a confusion matrix on test data set which predicted 190 MRI images are having brain tumor and 380 MRI images are not having brain tumor and some images were in false positive and false negative section. The final output displays the comparison between the original predicted brain tumor mask with the AI generated brain tumor mask which the model has created.

REFERENCES

[1]. Top 16 companies in AI-powered medical imaging https://research.aimultiple.com/looking-forbetter-medical-imaging-for-early-diagnostic-and-monitoring-contact-the-leading-vendors-here/

[2]. Introduction to U-Net and Res-Net for Image Segmentation https://aditi-mittal.medium.com/introduction-to-u-net-and-res-net-for-image-segmentation-9afcb432ee2f/

[3]. https://www.webmd.com/cancer/brain-cancer/qa/what-is-a-tumor, November, 21 2019.

[4]. https://en.wikipedia.org/wiki/Brain_tumor, November, 21 2019.

[5]. https://en.wikipedia.org/wiki/Convolutional_neural_network, December, 21 2019.

[6]. https://en.wikipedia.org/wiki/Loss_function, December, 21 2019.

[7]. https://en.wikipedia.org/wiki/Mathematical_optimization, December, 21 2019