A Project Report

on

3D U-NET CONVOLUTIONAL NEURAL NETWORK FOR SEGMENTATION OF BRAIN TUMOUR TISSUES IN HGG AND LGG MAGNETIC RESONANCE IMAGING

Submitted in partial fulfillment of the

requirement for the award of the degree of

BTech in Computer Science

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Deependra Rastogi
Assistant Professor

Submitted By

Arpit Mohan
19021011511/19SCSE1010326
Kartik Rajput
1902111446/19SCSE1010257

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
DECEMBER, 2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the thesis/project, entitled **"3D U-NET CONVOLUTIONAL NEURAL NETWORK FOR SEGMENTATION OF BRAIN TUMOUR TISSUES IN HGG AND LGG MAGNETIC RESONANCE IMAGING"** in partial fulfillment of the requirements for the award of the BTECH submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of July, 2021 to December, 2021, under the supervision of Deependra Rastogi, Assistant Professor, Department of Computer Science and Engineering, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by us for the award of any other degree of this or any other places.

Arpit Mohan(19SCSE1010326)

Kartik Rajput(19SCSE1010257)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Deependra Rastogi

Assistant Professor

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of Arpit Mohan: 19SCSE1010326 & Kartik Rajput: 19SCSE1010257 has been held on _____ and his/her work is recommended for the award of Btech in Computer Science and Engineering.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: November, 2013

Place: Greater Noida

Abstract

An abnormal rise in the tissue density is referred to as a lump, and if it occurs within the brain , it is referred to as brain tumor. These tumors arise as a result of the uncontrolled and abnormal cell division. They can also be malignant and invasive. A brain scan is a picture of the inside anatomy of the brain. Usually the brain scans are MRI(Magnetic Resonance Imaging). MRI provides an unparalleled view inside the human body.

For the treatment process, the segmentation of tumors from brain MRI is understood to be complicated and also crucial tasks. The proposed process can be further used in surgery, medical preparation, and assessments. In addition to this , the brain MRI classification is also essential. The enhancement of machine learning and technology will aid radiologists in diagnosing tumors without taking invasive steps. As the human assisted manual categorization can result in false/wrong predictions and diagnosis, brain tumor segmentation is one of the most important and difficult task in the field of medical image processing. Furthermore it is a difficult process when there is a huge amount of data to assist.

For the implementation we are using various tools, for coding part the language that has been used is python, for downloading dataset we have downloaded it from Kaggle, the name of the dataset is BraTS2020 dataset, and for implementation we have used Google Colab Platform.

By using 3D U-NET Convolutional Neural Network we will perform the process of segmentation of Brain tumor tissues in HGG and LGG Magnetic Resonance Imaging. A 3D convolutional neural organization (CNN) engineering is planned at the initial step to remove cerebrum growth and extricated cancers are passed to a pre-prepared CNN model for include extraction. The removed components are moved to the connection based choice technique and as the yield, the best elements are chosen. These chose highlights are approved through feed-forward neural organization for conclusive order. After this we get calculated value of hybrid result as the final value.

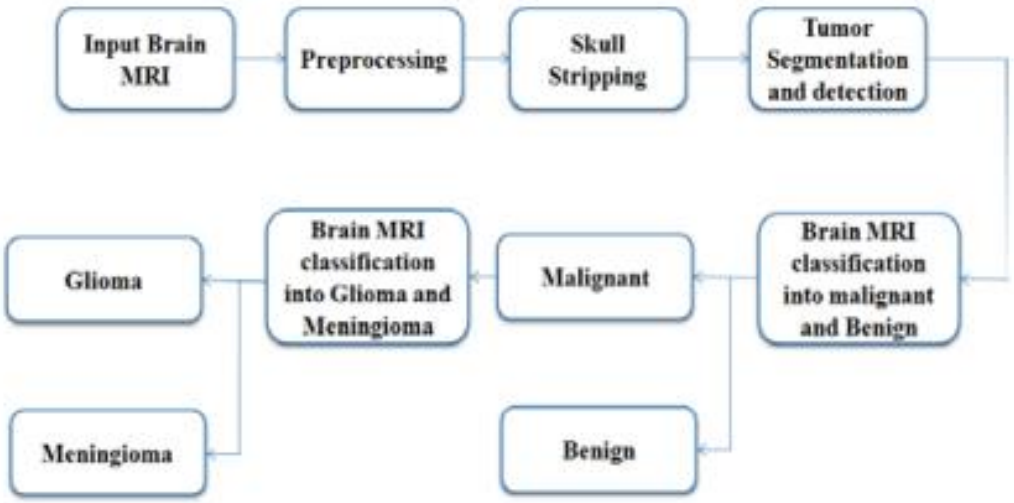
In current work we have a tendency to tend to developed the system to perform segmentation of brain tumor with the help of loss calculated. In future also the use of the performed model will be of great use because of the accuracy of the system in comparison to the manual work.

Keywords- 3D CNN, HGG, LGG, brain tumor, segmentation, healthcare.

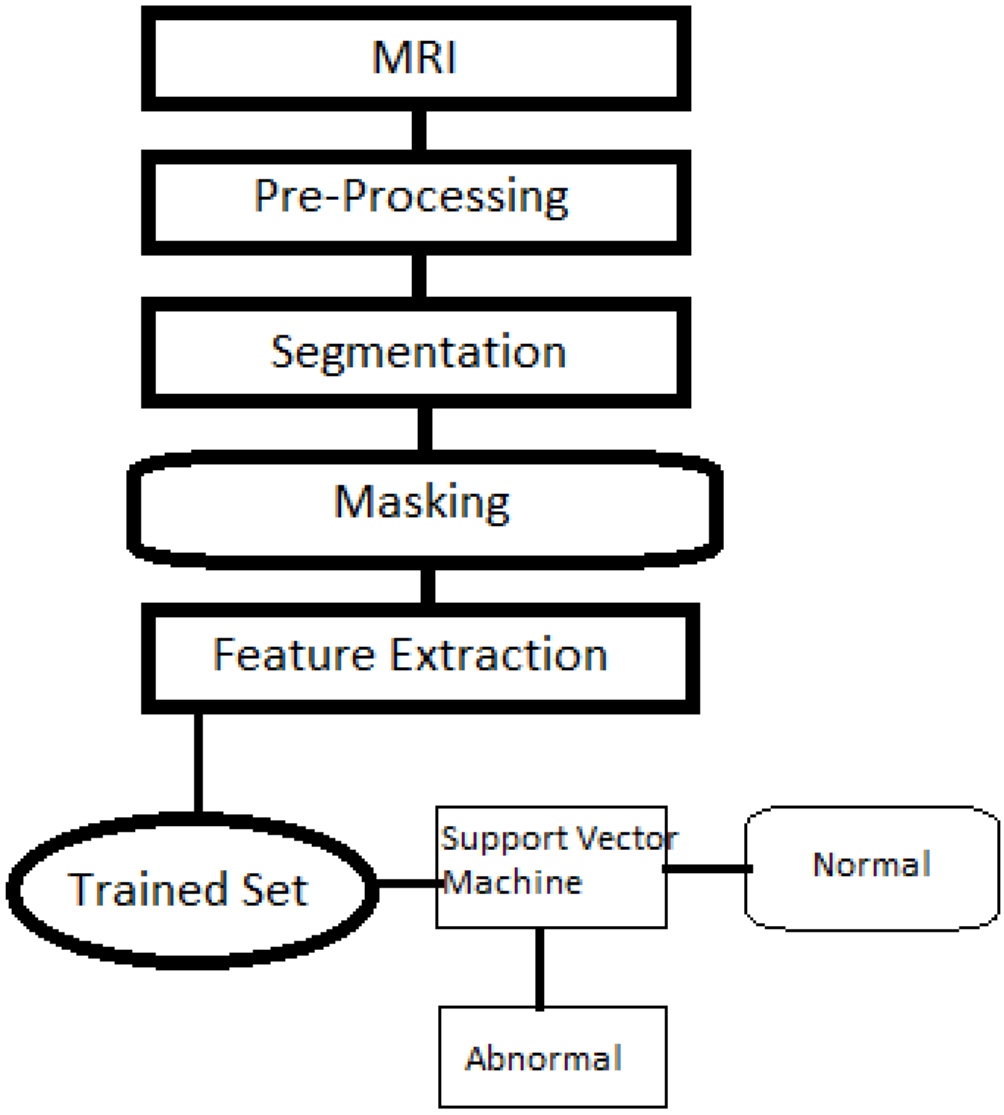
Table of Contents

List of Figures

1. UML Diagram



2. Data Flow Diagram



CHAPTER-1

Introduction

1.1 Introduction

A brain tumor is a collection of abnormal cells in your brain that forms a mass. Your brain is protected by a highly tough skull. Any expansion in such a small location can generate complications. Brain tumors can be malignant (cancerous) or benign (noncancerous) (benign). The pressure inside your skull might rise when benign or malignant tumors get larger. This can result in brain damage, which can be fatal.

There are two types of brain tumors: primary and secondary. The origin of a primary brain tumor is in the brain. The majority of initial brain tumors are harmless. A secondary brain tumor, also known as a metastatic brain tumor, develops when cancer cells from another organ, such as your lung or breast, migrate to your brain.

Brain tumors can affect brain function if they grow large enough to press on surrounding nerves, blood vessels and tissue. Your outcome is determined by such factors as the tumor's type, grade, and location; the success of tumor removal; and your age and overall health. Doctors diagnose brain tumors in about 85,000 people in the U.S. every year. Of those tumors, roughly 60,000 are benign, and about 25,000 are malignant. Brain tumors occur more often in men than women. Although they are most common among older adults, they can develop at any age. Brain tumors are the leading cause of cancer-related death in children under age 14.

Doctors are not sure what causes most brain tumors. Mutations (changes) or defects in genes may cause cells in the brain to grow uncontrollably, causing a tumor.

The only known environmental cause of brain tumors is having exposure to large amounts of radiation from X-rays or previous cancer treatment. Some brain tumors occur when hereditary conditions are passed down among family members. Symptoms of brain tumor include: Headaches that are ongoing or severe; or that occur in the morning or go away after vomiting, Behavior or personality changes, Confusion, Difficulty with balance or coordination, Trouble concentrating, Nausea and vomiting, Numbness, weakness or tingling in one part or side of the body or face,

Problems with hearing, vision or speech, Seizures, Unusual sleepiness, Trouble with memory, thinking, speaking or understanding language.

Doctors use several tests to confirm the presence of a brain tumor. These tests include:

Physical exam and medical history: Your doctor will perform a general health exam, looking for signs of diseases or illnesses. Your doctor will also ask questions about past and current health conditions, surgeries and medical treatments and family history of disease. Blood test: To check for tumor markers (substances secreted into blood by tumors) that are linked to certain types of tumors. Biopsy: Through a small hole in the skull, a doctor uses a needle to take a sample of tissue from the tumor. A laboratory studies the sample to identify details from the tumor, including how fast it is growing and whether it is spreading. Imaging tests: CTs, MRIs, SPECTs and PET scans help doctors locate the tumor and determine if it is cancerous or benign. Your doctor may also look at other parts of the body, such as the lungs, colon or breasts, to identify where the tumor started. Neurological exam: During a neurological exam, your doctor will look for changes in your balance, coordination, mental status, hearing, vision and reflexes. These changes can point to the part of your brain that may be affected by a tumor. Spinal tap: A doctor uses a small needle to remove fluid from around the spine. A laboratory examines this fluid to look for cancer cells, which can indicate a malignant tumor somewhere in the central nervous system.

In most cases, magnetic resonance imaging (MRI) is used to diagnose a brain tumor (MRI). When an MRI reveals a brain tumor, the most frequent technique to diagnose the type of tumor is to examine the results of a biopsy or surgery on a sample of tissue.

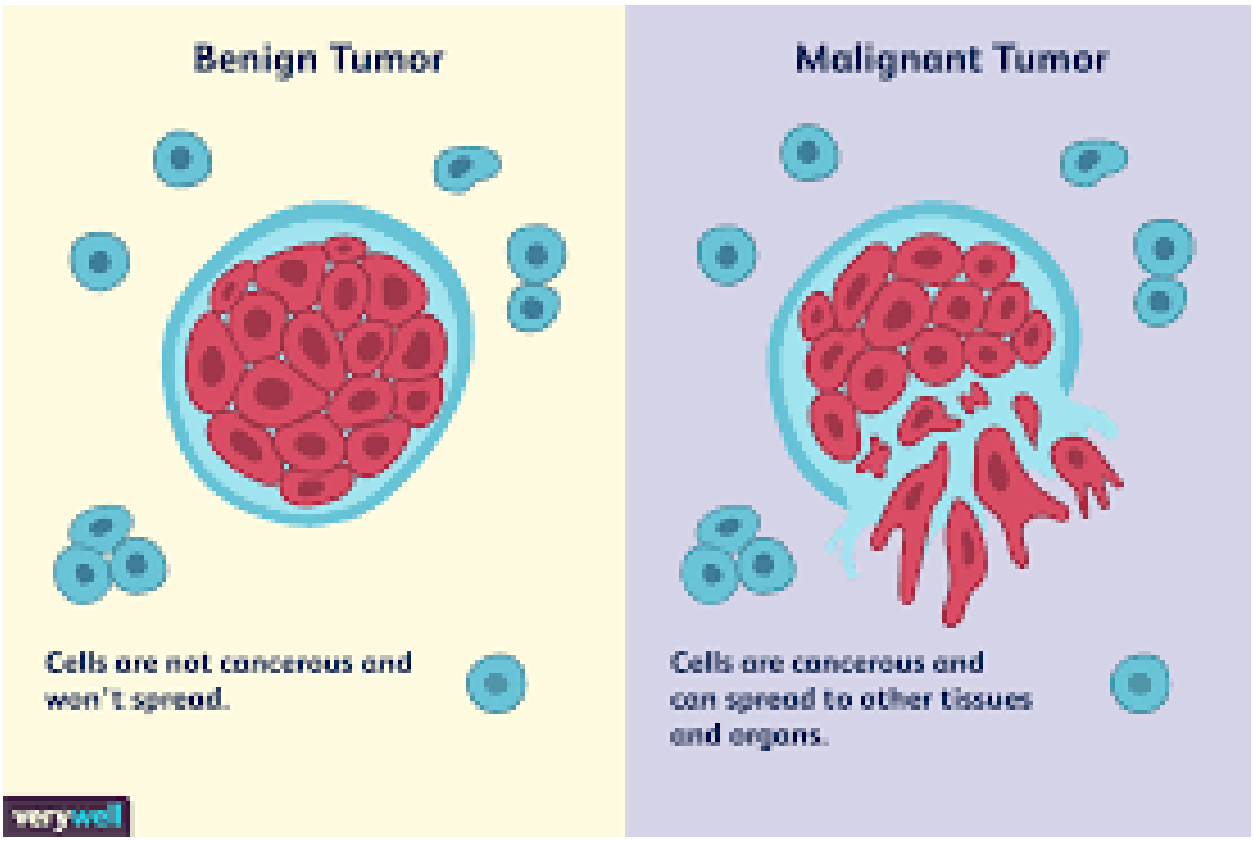
MRI uses magnetic fields, not x-rays, to produce detailed images of the body. MRI can be used to measure tumor size. A special dye called contrast medium is applied before scanning to create a clearer image. This dye can be injected into a patient's vein or given as a pill or ingested liquid. MRIs create images that are much more detailed than CT scans (see below) and are a preferred method for detecting brain tumors. An MRI can be for the brain, spinal cord, or both, depending on the type of tumor suspected and the likelihood that it will spread to the CNS. There are different types of MRI. The results of the neuro-examination, performed by an internist or neurologist, help determine what type of MRI should be used.

In the proposed research paper we are going to detect brain tumor using machine learning and artificial intelligence, A 3D convolutional neural organization (CNN) engineering is planned at the initial step to remove cerebrum growth and extricated cancers are passed to a pre-prepared CNN

model for include extraction. The removed components are moved to the connection based choice technique and as the yield, the best elements are chosen. These chose highlights are approved through feed-forward neural organization for conclusive order.

With reference to various research papers it has been seen that deep learning systems for cerebrum cancer order accomplished preferable outcomes over conventional procedures. Along these lines, in this article, we think about the issue of incorrect division of the mind cancer and hearty convolutional neural elements for characterization. For this reason, another completely auto-mated 3D CNN-based model is suggested that portion the growth at the absolute first stage and further group it into their applicable classifications.

The diagram below shows the two types of brain tumors:



AI plans and assembles calculations that are directed by information. Rather than relying upon people to unequivocally characterize guidelines, AI calculations use preparing sets of real world information to create new information, known as forecasts, that demonstrate to be more exact than those from models planned by people. Inside the field of AI, neural organizations are a subset of calculations worked around a model of counterfeit neurons spread across many interconnected layers. Inside neural organizations, profound

learning depicts intricate and more profound organizations than expected. The benefit of these additional layers is that the organizations can foster a lot extraordinary degrees of deliberation. This is important for explicit assignments, as picture acknowledgment and programmed interpretation which are extremely intricate.

Division is the segment of a computerized picture into comparative locales to improve on the picture portrayal into something more significant and more straightforward to examine. A Convolutional Neural Network is essential for the class of Neural Networks that are utilized with picture information. These have been fruitful in recognizing faces, objects, traffic signs, just as driving machine vision in robots and self-driving vehicles. CNN's have been applied effectively on an assortment of biomedical division issues. Most existent methodologies depended on utilizing 2D CNN's for handling 3D volumes, because of challenges being accounted for when preparing with entire 3D volumes. There 2D structures may be effective sometimes yet they are problematic in their utilization of currently existent 3D data.

At the point when we started, a base code from an existent 3D UNET was utilized for testing. It was recommended to utilize a profound organization that figures out how to create thick volumetric divisions, yet just requires commented on 2D cuts for preparing. This organization could be utilized in two unique situations, the principal application simply points on densification of a scantily explained informational collection; the subsequent one gains from numerous scantily explained informational indexes to sum up to new information. Like us, there are various analysts chipping away at the division of pictures for various applications. Mind sickness research is among the most elevated looked for studies to track down better calculations for the division or forecast of these illnesses. CNNs are utilized to portion explicit illnesses for instance malignant growth, sore, and alzheimer’s disease; fabricating profound neural organizations to examine explicit spaces of the cerebrum in both fix savvy also looking over an entire 3D volume are various ways these scientists took care of the forecasts of these illnesses with a CNN. Among different investigations to make sickness division more straightforward is the cerebrum division, extraction or skull stripping, which cleans the MRI and leaves just the cerebrum. CNNs are likewise utilized for ordinary use research like picture characterization and displaying sentences, which really makes arrangement assignments

more straightforward with a PC. With our CNN, we utilize more than one of these helpful applications some explores have made, for example, skull stripping . This diminishes intricacy for the investigation of the preparation information. If the skull was not eliminated we could get misidentified growths, more slow execution, longer preparing times, and more regrettable outcomes. We approve the proposed model on two informational indexes: the first is the BRaTS2020 information set and clinical information gathered from neurosurgery. The two sets contain T1 weighted MRIs. The BRaTS2020 informational index obviously was made specialists, while we made the best to make our physically veiled set be correspondingly veiled to the BRaTS2020, on an alternate arrangement of information. The utilization of more than one set and information increase lessens the likelihood of overfitting in our model. The 3D volumes are parted into 2D cuts also the subsequent pictures are resampled, normalized, and standardized so the model is taken care of steady information input. The patient examples are picked aimlessly toward the beginning of the preparation in a 80/20 split for preparing and approval.

1.2 Formulation of Problem

Another 3D CNN and connection along FNN based robotized approach is proposed in this work for mind growth location and characterization. The proposed technique comprises of three center advances—another CNN engineering based cerebrum cancer extraction, (b) pretrained profound provisions extraction, and (c) Pearson connection alongside FNN highlights choice for definite grouping.

Image Segmentation:

For the most part, The edge utilized during the time spent picture division by putting every one of the pixels that are higher than the limit level to a frontal area while different pixels to the foundation esteem. Any powerful change as indicated by the pixel power can't be accomplished when utilizing edge strategy . In proposed strategy we utilized Adaptive edge that normally take the dark or shading pictures as information and yields in the type of parallel picture addressing division. Versatile thresholding strategies used to isolate the object of a picture from its experience. The primary distinctive among edge and Adaptive thresholding is that the Adaptive

limit esteem is determined for every pixel in the picture. This procedure gives more power to changes in enlightenment. After utilized versatile thresholding, the district location process is performed on the twofold picture that outcomes from an versatile thresholding step. Area discovery is Image division method that orders pixels in the picture to one or a few separate regions or mass which is a space of contacting pixels with a similar rationale state. The locale location comprises of filtering and marking any new districts, yet additionally consolidating old locales when they end up being associated on a lower column. Along these lines, the picture is filtered and each pixel is separately named with an identifier which connotes the district to which it has a place. The parallel picture result has many article alongside the object of cancer, by utilizing the district identification strategy the greatest region object are extricated (this item is the cancer) and put it in a different picture.

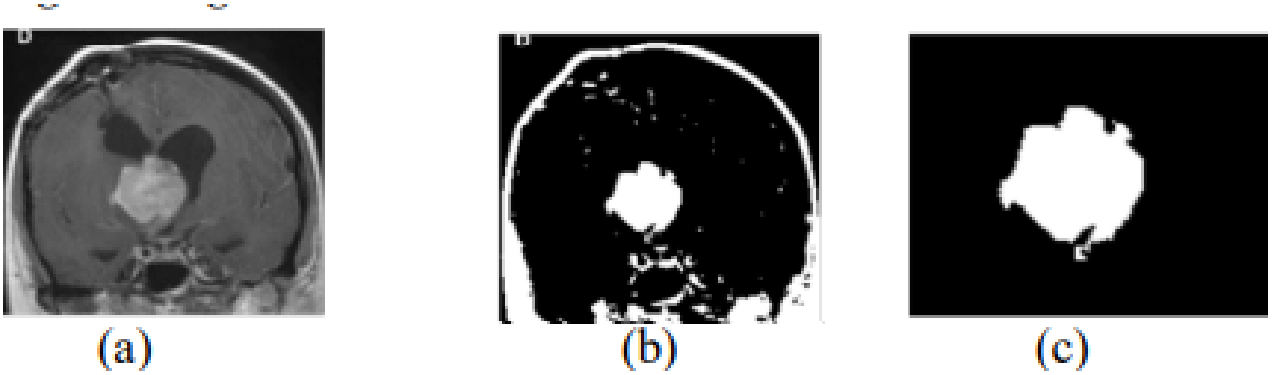


Fig. 4: a) MR Images with Sigma Filter. B) Result of MR Images after Used Region Detection Method. C) Object of Tumor

Classification Using Machine Learning

Surface and Statistical based elements are more famous for identifying the Region of Interest (ROI). In light of these highlights we can isolate the tumorous and non-tumorous MRI. We utilized surface and measurable based components for characterization. Surface based provisions like-Dissimilarity, Homogeneity, Energy, Correlation, ASM and Statistical based highlights including-Mean, Entropy, Centroid, Standard Deviation, Skewness, Kurtosis were removed from the fragmented Brain growth. Further, we removed the Area, Convex Hull Area and Diameter of the growth. Extrapolating Arched Hull Area and Diameter of the growth. Extrapolating these

components from the fragmented MRI, we arranged the picture as the presence of ordinary and strange tissue.

Classification Using CNN

The five-layer proposed approach will give us the honorable outcome for the location of the cancer. Convolution, Max Pooling, Flatten, and two thick layers are the proposed five layer CNN model. Information expansion will be finished prior to fitting the model as CNN is interpretation invariance. We will assess the exhibition in two ways dependent on parting the dataset. We will achieve exactness for 70:30

parting proportion. Then, at that point, at the subsequent cycle, 80% of the pictures will be allocated for preparing and the remainder of the pictures will be licensed for testing. So our proposed model gives the best outcome when the division is 80:20.

We will examine with an alternate number of layers. Further, we will utilize 0.2 as the dropout esteem.

1.2.1. TOOLS AND TECHNOLOGY USED

1. PYTHON

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library. Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0.[33] Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a cycle- detecting garbage collection system (in addition to reference counting). Python 3.0 was released in 2008 and was a major revision of the language that is not completely backward-compatible. Python 2 was discontinued with version 2.7.18 in 2020. Python consistently ranks as one of the most popular programming languages. Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in

the Netherlands as a successor to the ABC programming language, which was inspired by SETL, capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989. Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's "benevolent dictator for life", a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decisionmaker. In January 2019, active Python core developers elected a five-member "Steering Council" to lead the project. Python 2.0 was released on 16 October 2000, with many major new features, including a cycle-detecting garbage collector (in addition to reference counting) for memory management and support for Unicode. Python 3.0 was released on 3 December 2008. It was a major revision of the language that is not completely backward-compatible. Many of its major features were backported to Python 2.6.x and 2.7.x version series. Releases of Python 3 include the 2to3utility, which automates the translation of Python 2 code to Python 3. Python 2.7's end-of-life date was initially set at 2015 then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3. No more security patches or other improvements will be released for it. With Python 2's end-of-life, only Python 3.6.x and later are supported. Python 3.9.2 and 3.8.8 were expedited as all versions of Python (including 2.7) had security issues, leading to possible remote code execution and web cache poisoning.

2.GOOGLE COLAB

You will quickly learn and use Google Colab if you know and have used Jupyter notebook before. Colab is basically a free Jupyter notebook environment running wholly in the cloud. Most importantly, Colab does not require a setup, plus the notebooks that you will create can be simultaneously edited by your team members – in a similar manner you edit documents in Google Docs. The greatest advantage is that Colab supports most popular libraries which can be easily loaded in your notebook.

3. Kaggle (For BraTs2020 dataset)

Kaggle, a subsidiary of Google LLC, is an online community of data scientists and machine learning practitioners. Kaggle allows users to find and publish data sets, explore and build models in a web-based data-science environment, work with

other data scientists and machine learning engineers, and enter competitions to solve data science challenges.

Link for the BraTs2020 dataset: https://www.kaggle.com/awsaf49/brats20-dataset-training-validation

CHAPTER-2 LITERATURE SURVEY

Many of the researchers proposed many methods, and algorithms for to find brain tumor, stroke and other kinds of abnormalities in human brain using m r images. manoj k kowar and sourabh

yadav et al, 2018 his paper “brain tumor detection and segmentation using k- nearest neighbor (k-nn) algorithms ”. They presents the novel techniques for the detection of tumor in brain using segmentation, histogram and thresholding [4]. Rajesh c. patil and dr. A. S. Bhalchandra et al, in his paper “brain tumor extraction from MRI images using matlab”, they focused on meyer's flooding watershed algorithm for segmentation and also presents the morphological operation [5]. Vinay Parameshwarappa and Nandish s. et al, 2018 in his paper “segmented morphological approach to detect tumor in brain images”, they proposed an algorithm for segmented morphological approach [6]. m. karuna and ankita joshi et al, 2017, in his paper “automatic detection of brain tumor and analysis using matlab” they presents the algorithm incorporates segmentation through nero fuzzy classifier. the problem of this system is to train the system by neural network and it desires many input images are used to train the network. the developed system is used only for tumor detection not for other abnormalities [7]. R. B. Dubey, M. Hanmandlu, Shantaram vasikarla et al, 2017, compare the image segmentation techniques in his paper “evaluation of three methods for mri brain tumor segmentation”, they apply preprocessing techniques like; de-noising, image smoothing, image contrast enhancement and comparison of the level set methods and morphological marker controlled watershed approach and modified gradient magnitude region growing technique for mri brain tumor segmentation. they concluded the mgmrgt method gives better result [8]. Sentilkumaran n and Thimmiaraja et al, 2017 compare the image enhancement techniques in his paper “histogram equalization for image enhancement using mri brain images”, they presented the study of image enhancement techniques and comparison of histogram equalization basic method like brightness preserving adaptive histogram equalization (AHE), local histogram equalization (IHE), global histogram equalization (GHE), dynamic histogram equalization using different quality objective measures in mri

images. they also presented the better result on contrast using bpdhe method [10]. r. preetha and g. r. suresh et al, 2016, in his paper “performance analysis of fuzzy c means algorithm in automated detection of brain tumor” they used fuzzy c means clustering for segmentation. that method given the high computational complexity. fcm shows good performance result in segmented the tumor tissue and accuracy of tumor. segmentation was identified by applied the svm classifier [11]. amer albadarneh, hasan najadat and ali m. alraziqi et al, 2016, [12] proposed the method for brain tumor classification of mri images. the research work applied, based on neural network (NN) and k- nearest neighbor (K-NN) algorithms on tumor classification has been achieved 100% accuracy using KNN and 98.92% using nn. many researchers has proposed many algorithms and segmentation techniques to find abnormalities in the brain using MRI images. most of them proposed various algorithms to find the abnormality in the brain like brain tumor.

Hassan Khotanloua et al [6]. proposed a new method to segment brain tumors in 3D MR Images. The first step in the proposed method is the brain MR Images segmentation using a new and powerful approach to detecting tumors. Then tumor detection was performed depend on choosing asymmetric areas. This method considers with the brain symmetry plane and used fuzzy classification. Its result forms the initialization of a segmentation process depend on a combination of a spatial relations and deformable model, leading to accurate segmentation of the brain tumors. Qiang Wang et al [7]. using the information from magnetic resonance (MR) imaging and magnetic resonance spectroscopy (MRS) to assist in clinical diagnosis. The proposed approach consists of several steps including segmentation, feature extraction, feature selection. Classification model construction for used to classify the brain case to the normal or abnormal. A segmentation technique based on fuzzy connectedness was used. They outline the

tumor mass boundaries in the MR Images. The concentric circle technique on the regions of interest was applied to extract features. Feature selection was performing to remove redundant features. Experimental results demonstrate the effectiveness of the proposed approach in classifying brain tumors in MR Images. Yudong Zhanga et al [8]. proposed approach to classify MR Images as abnormal or normal using neural network. The first step in this method was extracted features from MR brain image by employed wavelet transform. And then reduce the number of features using the technique of principle component analysis. The results are given to a neural network. The method applied on 66 images 18 of them was normal and other abnormal. The classification accuracies were 100%. Rajeswari S. et at [9]. Proposed a method based on texture features such as Grey Level Co-occurrence Matrix GLCM of MR Images. They use Sequential Forward selection algorithm to select the discriminative features. The proposed method classify MR Images to normal and abnormal by applied Afterwards an advanced kernel based technique such as Support Vector Machine (SVM) . A. Jayachandran et al [10]. they proposed a hybrid algorithm for detection brain tumor using statistical features and Fuzzy Support Vector Machine classifier. The proposed method consists of four steps. In the first step anisotropic filter was performed for noise reduction. In the second step, the texture features extracted from MR Images. In the third step, the features of MR Images have been reduced using principles component analysis to the most essential features. Final step, the tumor was classified to normal and abnormal by using Supervisor classifier based Fuzzy Support Vector Machine. The accuracy of Classification was 95.80%. PrachiGadpayle et al [11]. developed System for a brain tumor Detection and Classification. The image processing techniques such as preprocessing, image enhancement, image segmentation, morphological operations and feature extraction have been implemented for the detection of brain tumor in the MRI images. The

features texture such Gray Level Cooccurrence Matrix (GLCM) was used in the detected tumor. They classify MRI brain image into abnormal and healthy image using BPNN and K-NN classifier. N.M. Saad et al [12]. proposed method to detect and classify a brain tumor using thresholding and a rule-based classifier. Four types of brain tumor depend on diffusion-weighted imaging were analysed such acute stroke, solid tumor, chronic stroke and necrosis. In the detection and segmentation stage, the image is divided into 8x8 macro-block regions. Adaptive thresholding technique is applied to segment the tumor's region. Statistical features are measured on the region of interest. The rule based classifier was used to classify four types of lesions. The accuracy of classification obtained from this method was 93%, 73%, 84% and 60% for acute stroke, solid tumor, chronic stroke, and for necrosis respectively.

PROJECT DESIGN

This section presents brain tumor detection and classification techniques. The three stages of the proposed system are:

• Brain Tumor Detection.

• Benign and Malignant Brain MRI Classification.
• Glioma and Meningioma Brain MRI Classification.
A. Brain Tumor Detection The methodology to detect the brain tumor from the brain MRI discusses in this section.
1) Tumor Vs. Non-Tumor Dataset: The online data is collected from the online source for tumorous and nontumorous classification. This dataset consists of 154 tumorous MRIs and 91 non-tumorous MRIs.
2) Pre-processing: In the normalization process, the intensity falls within the range of pixel values converted into [0 1] range. In this process, each pixel intensity is divided by the maximum intensity values within an image. Normalization can create binary thresholding by creating a more extensive source. Such MRI images can help to prevent classifications affected by variations of grayscale value.
3) Skull stripping: Skull stripping is a necessary procedure in the biomedical image examination for the efficient analysis of brain tumors from brain MRI. It eliminates the non-brain parts like skin, fat, and skull from the brain MRI.

We developed a customized, highly iterative U-Net that allow for rapid prototyping and testing of designs. The models are generated dynamically based on architecture settings such as depth of the network, segmentation levels for encoders and decoders, activation functions for each block, kernel initializers and constraints. All settings are easily modified, and the changes are reflected in the model summary Keras generates when run. Many existing medical imaging segmentation neural networks, including the leading 3D CNN U-Net, rely on networks dozens of layers deep with many millions of trainable parameters. These models are very flexible in their ability to be used on a variety of different images and segment different features while being able to produce results with impressive accuracy. The goal of our paper is two-fold: develop a neural network that could produce 3D volumes of tumors from brain MRIs, but also to find an alternative design to these vast, extremely deep networks. This goal is inspired by just how long some of these models took to generate a prediction, let along train. Medical image segmentation should be performed as quickly as possible without sacrificing accuracy. Training times are not as important as prediction times, but training should also be a relatively quick process in case the network should need to be retrained with new data. A shallow neural network has clear

disadvantages over deeper networks of similar design due to an objectively fewer number of training parameters. To overcome this obstacle and create a network that can produce very quick and accurate results, we take full advantage of every convolution filter. As seen in figure 1 we strategically designed the encoding and decoding blocks of our U-Net model with the idea of eliminating co-adaptation. Inspired by [13] our work employed 3D varieties of convolution, max pooling, and deconvolution/ upsampling layers. We believe that this increase in dimensionality, while clearly effective, was unnecessary if the network is crafted just so. Our model uses conventional 2D layers and runs through axial slices of an MRI to create a 3D volume instead. This reduction in dimensionality means that instead of training on 1283 voxels per sample, the network trains on 1282 pixels per slice. This results in reduced preprocessing, training, and prediction times.

CHAPTER 3 FUNCTIONALITY/ WORKING OF PROJECT

[ ]:

```
import pandas as pd
import numpy as np
import cv2
from skimage.io import imread
import matplotlib.pyplot as plt
import json
import pickle
import os
import glob
import random
import nibabel as nib
import imageio
from IPython.display import Image
from sklearn.preprocessing import MinMaxScaler
import torch
import torchvision
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
from torch.utils.data import DataLoader
from torch.utils.tensorboard import SummaryWriter
from torchvision import transforms
from torch.utils.data import DataLoader , Dataset
from torchvision.datasets import ImageFolder
from torchvision.utils import make_grid , save_image , draw_segmentation_masks
#!pip install torchsummary
#from torchsummary import summary
```

dataset_path = '../input/brats20-dataset-training-validation/BraTS2020_TrainingData/MICCAI_BraTS2020_TrainingData'

sample_dir = os.listdir((os.path.join(dataset_path , 'BraTS20_Training_355')))
sample_dir

VISUALIZING SAMPLE DATA

+ Code + Markdown

[4]:
```
sample_image_flair = nib.load(os.path.join(dataset_path , 'BraTS20_Training_355' , sample_dir[0])).get_fdata()

print(sample_image_flair.shape)
print(sample_image_flair.max())
```

```
(240, 240, 155)
1854.603271484375
```

[5]:
```
sample_image_flair.reshape(-1, sample_image_flair.shape[-1]).shape
```

[5]:
```
(57600, 155)
```

[6]:
```
scaler = MinMaxScaler((0, 1))
```

[7]:
```
# Scalers are applied to 1D so let us reshape and then reshape back to original shape
sample_image_flair = scaler.fit_transform(sample_image_flair.reshape(-1, sample_image_flair.shape[-1])).reshape(sample_image_
```

```
print(sample_image_flair.dtype)
```

```
(240, 240, 155)
1.0
0.0
float64
```

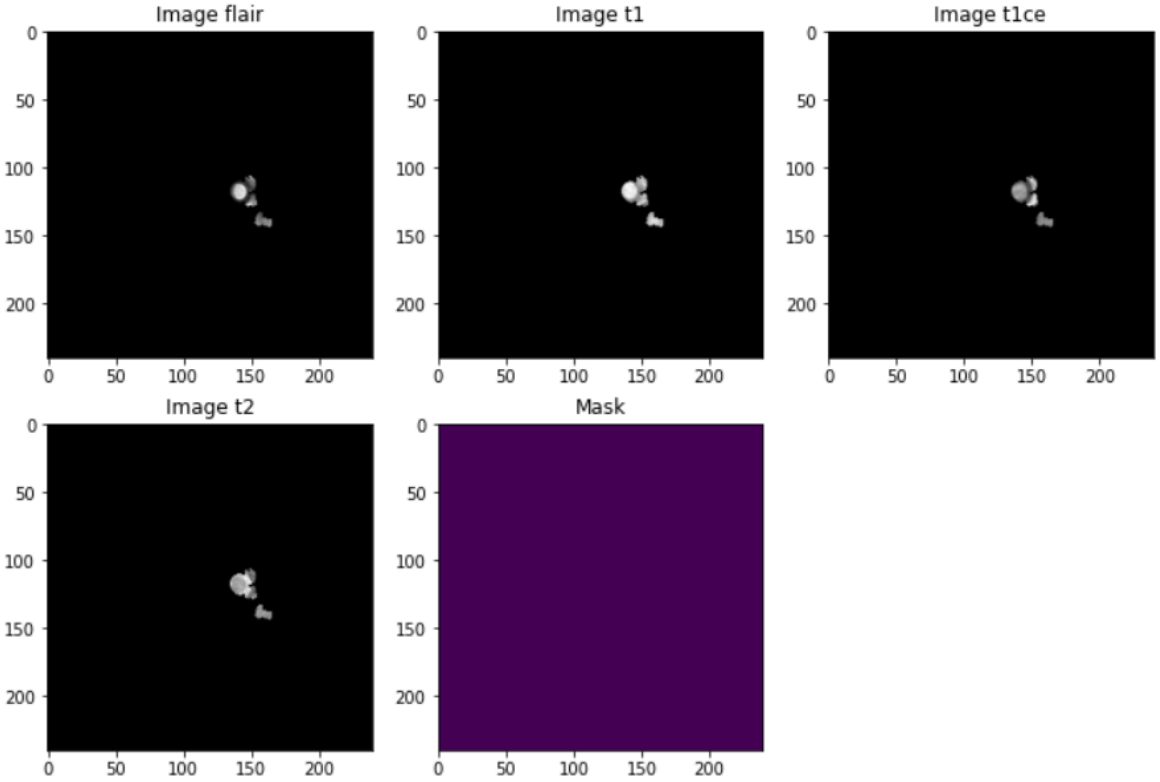
[8]:

```
sample_image_t1 = nib.load(os.path.join(dataset_path , 'BraTS20_Training_355' , sample_dir[3])).get_fdata()
sample_image_t1 = scaler.fit_transform(sample_image_t1.reshape(-1, sample_image_t1.shape[-1])).reshape(sample_image_t1.shape)

sample_image_t1ce = nib.load(os.path.join(dataset_path , 'BraTS20_Training_355' , sample_dir[-1])).get_fdata()
sample_image_t1ce = scaler.fit_transform(sample_image_t1ce.reshape(-1, sample_image_t1ce.shape[-1])).reshape(sample_image_t1c

sample_image_t2 = nib.load(os.path.join(dataset_path , 'BraTS20_Training_355' , sample_dir[2])).get_fdata()
sample_image_t2 = scaler.fit_transform(sample_image_t2.reshape(-1, sample_image_t2.shape[-1])).reshape(sample_image_t2.shape)
```

```
sample_mask = nib.load(os.path.join(dataset_path , 'BraTS20_Training_355' , sample_dir[1])).get_fdata()
print(sample_mask.shape)
sample_mask = sample_mask.astype(np.uint8)

print(np.unique(sample_mask))  # Given mask labels = 0, 1, 2, 4 (Need to re-encode to 0, 1, 2, 3)
```

```
(240, 240, 155)
[0 1 2 4]
```

```
plt.imshow(sample_mask[:,:,n_slice])
plt.title('Mask')
plt.show()
```



```
combined_x = np.stack([sample_image_flair, sample_image_tice, sample_image_t2], axis=3)
combined_x.shape
```

[12]:
```
(240, 240, 155, 3)
```

[13]:
```
def to_categorical(y, num_classes):
    """ 1-hot encodes a tensor """
    return np.eye(num_classes, dtype='uint8')[y]
```

[14]:
```
label_cat = to_categorical(sample_mask, num_classes=4).astype(np.uint8)
label_cat.shape
```

[14]:
```
(240, 240, 155, 4)
```

```
image_norm = cv2.normalize(combined_x[:, :, :, 0], None, alpha=0, beta=255,
                           norm_type=cv2.NORM_MINMAX, dtype=cv2.CV_32F).astype(np.uint8)
image_norm.shape
```

[15]:
```
(240, 240, 155)
```

+ Code + Markdown

Console

[17]:
```
data_all = []
data_all.append(labeled_image)
np.array(data_all).shape
```

[17]: (1, 240, 240, 155, 3)

[18]:
```
# coronal plane
coronal = np.transpose(data_all, [1, 3, 2, 4, 0])
coronal = np.rot90(coronal, 1)
coronal.shape
```

[18]: (155, 240, 240, 3, 1)

```
# transversal plane
transversal = np.transpose(data_all, [2, 1, 3, 4, 0])
transversal = np.rot90(transversal, 2)
transversal.shape
```

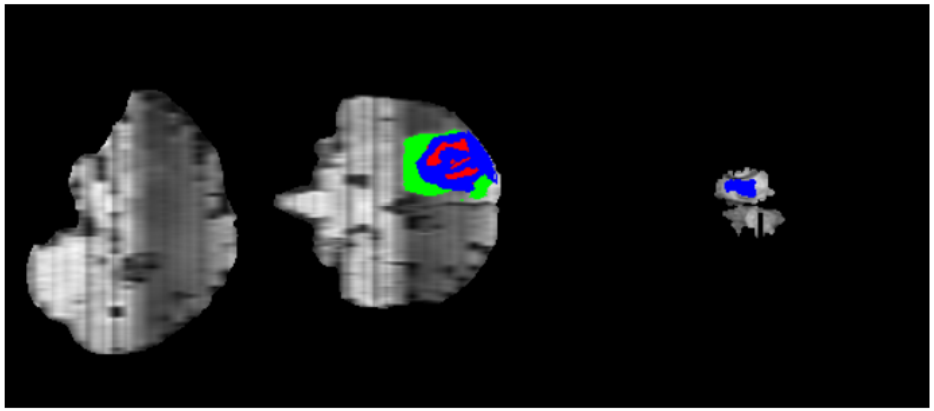
[19]: (240, 240, 155, 3, 1)

```python
fig, ax = plt.subplots(3, 6, figsize=[16, 9])

for i in range(6):
    n = np.random.randint(coronal.shape[2])
    ax[0][i].imshow(np.squeeze(coronal[:, :, n, :]))
    ax[0][i].set_xticks([])
    ax[0][i].set_yticks([])
    if i == 0:
        ax[0][i].set_ylabel('Coronal', fontsize=15)

for i in range(6):
    n = np.random.randint(transversal.shape[2])
    ax[1][i].imshow(np.squeeze(transversal[:, :, n, :]) , cmap='gray')
    ax[1][i].set_xticks([])
    ax[1][i].set_yticks([])
    if i == 0:
        ax[1][i].set_ylabel('Transversal', fontsize=15)

for i in range(6):
    n = np.random.randint(sagittal.shape[2])
    ax[2][i].imshow(np.squeeze(sagittal[:, :, n, :]) , cmap='gray')
    ax[2][i].set_xticks([])
    ax[2][i].set_yticks([])
    if i == 0:
        ax[2][i].set_ylabel('Sagittal', fontsize=15)

fig.subplots_adjust(wspace=0, hspace=0)
```

Coronal
Transversal
Sagittal

```
        img = np.concatenate((x, y, z), axis=1)
        images.append(img)
    imageio.mimsave("./gif.gif", images, duration=0.01)
    return Image(filename="./gif.gif", format='png')
```

[23]:
```
visualize_data_gif(labeled_image)
```

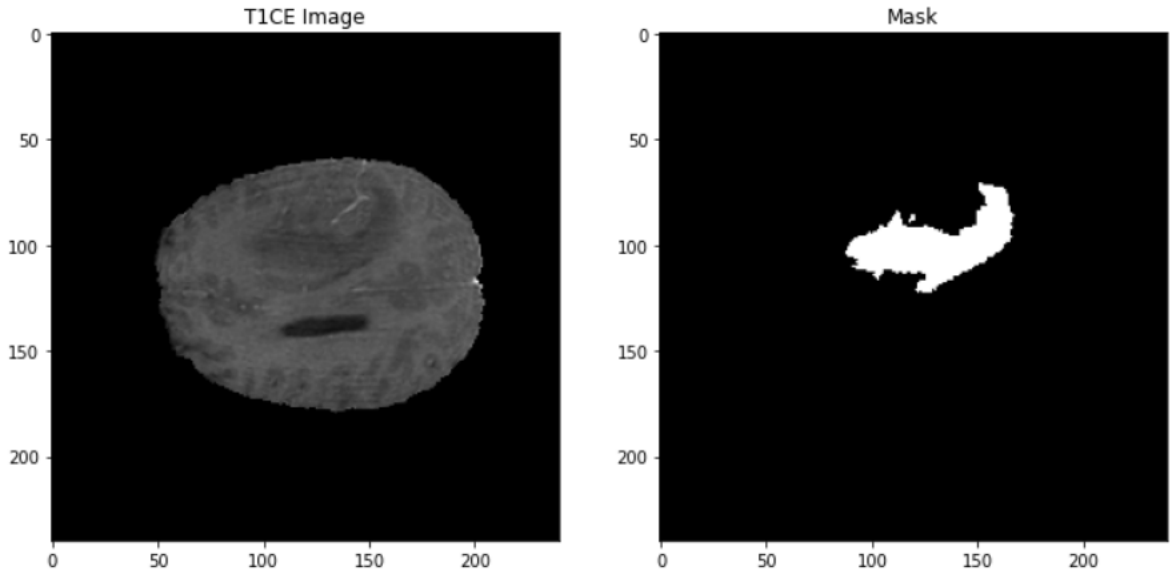
[23]:


DATA PREPARATION

The images are resized to 128 x 128 x 128 from 240 x 240 x 155 for better GAN training.

+ Code + Markdown

```
plt.imshow(mask[:,:,97], cmap='gray')
plt.title('Mask')
plt.show()
```




```
!mkdir ./images
!mkdir ./masks
```

+ Code + Markdown

OPTIMIZATION

A. Loss function

In order for the optimizer to properly evaluate and back-propagate error, we use the dice coefficient (similar to an intersection over union) as a metric to measure the accuracy of the models predictions. A typical accuracy metric quickly proved useless for this image segmentation task as the vast majority of the image was marked as background (zero on the ground truth data). A binary accuracy metric counts the number of pixels, or in this case voxels, are correct, including background data. Rewarding the model for correctly tagging the background as well as nontumor areas as not being a tumor is proved to lead to very suboptimal solutions. The problem of too much rewarding was solved by using a slightly modified dice coefficient as the loss function as shown in equation below. The loss function does not reward the network for correctly predicting background, it only is rewarded for correctly predicting features. The network is then received negative reinforcement for false positives and false negatives. This creates a much more conservative measurement for accuracy of the network. For example, during one training step the network reported an accuracy of 0.9847, but the dice coefficient was only 0.6931. Having a metric that punishes for false positives instead of rewarding for true negatives allows for the optimizer to follow a much more reliable gradient.

$$\mathrm{T} = vec(\mathrm{Truth})$$
$$\mathrm{P} = vec(\mathrm{Pred})$$
$$\mathrm{DSC} = \frac{2 \cdot \sum (T \cdot P) + \epsilon}{\sum \mathrm{Truth} + \sum \mathrm{Pred} + \epsilon}$$

vectorize ground truth
vectorize prediction
e=0.00001
LOSS = 1- DSC

The loss function is computed by first taking the ground truth and prediction tensors and vectorizing or flattening them. This operation keeps each value paired up properly between the truth and prediction tensors. Next, the dice coefficient is computed with a smoothing factor as shown in the equation. Finally, to transform the dice coefficient (ranging from 0 to 1) to a loss

function (ranging from 1 to 0), the dice coefficient is subtracted from 1. This allows smaller values to reflect better predictions, ideal for how a loss function should behave.

B. Optimizer

Another attempt made to overcome the inherent drawbacks of a shallow network was to use the ADADELTA optimizer, an alternative to the standard preferred ADAM optimizer [1]. ADADELTA was chosen because of its low computational costs over SGD and the dynamic adjustments it makes. In addition, ADADELTA does not have the learning rate decay issue that ADAGRAD faces. ADADELTA dynamically regulates its learning rate, and thus does not require extensive guesswork of manual learning rate scheduling. ADADELTA begins with an extremely aggressive learning rate default of 1.0 as determined from the Keras source code. This allows the network to quickly follow the gradient before the learning rate is adjusted by the optimizer and the network fine tunes over time. Our model, specifically, required only a few epochs of training on the BRaTS2020 dataset before it reached a stable solution that then required finetuning over the following epochs.

EXPERIMENTS A. Data The neural network was trained on the BRaTS dataset with preoperative TCGA BGM and TCGA LGG images and segmentations. Due to time restraints in this project, only the post-contrast T1 weighted MRI was used for training. The neural network may very well perform better with all modality data instead of just the postcontrast images. Testing was performed with T1 weighted MRIs from the Brigham and Womens Hospital clinical data of 15 patients with brain tumors. B. Preprocessing Information contained in the MRIs contained in the BRaTS dataset used and the clinical dataset provided by Brigham and Womens Hospital are not consistent with each other in terms of data ranges and even data types stored in the MRI files. A neural network performs best when input data ranges are relatively similar, thus causing neurons to activate in similar manners from one input sample to the next. To give the network the best chance at high performance, the MRIs are all resampled to a cube with sides of 128 voxels in length. The MRI data is then sliced along the axial view and portioned into batches of 60 for the network to process. Each batch of 75 MRI axial slices is normalized to have a mean of 0 and standardized so that voxel values are standard deviations instead of their raw data. The mean and standard deviation are obtained feature-wise through the Keras ImageDataGenerator

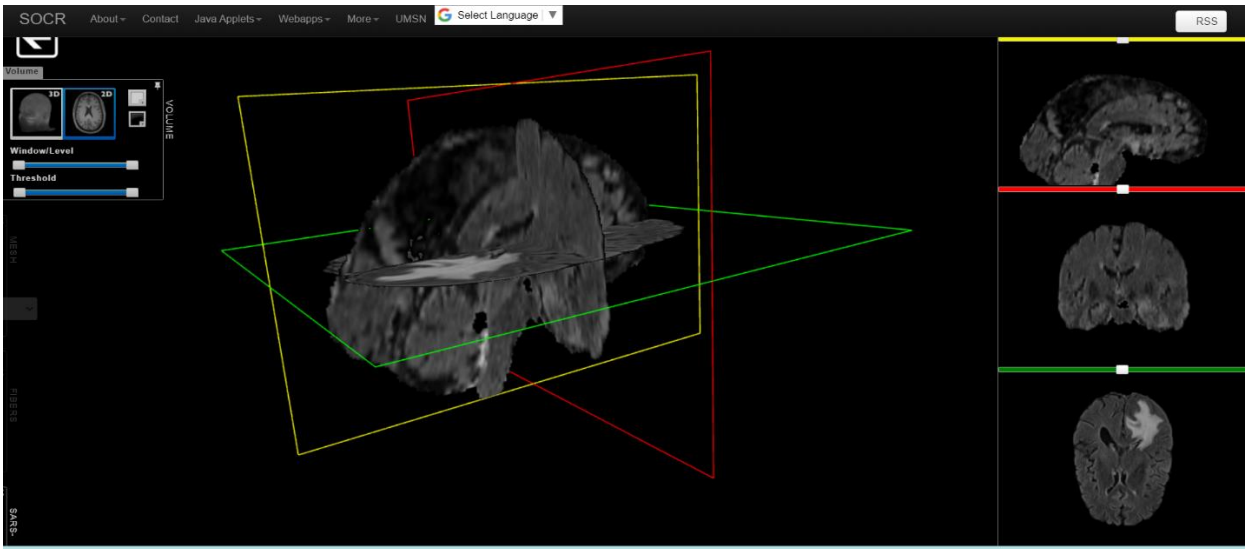
[2]. Since all of the BRaTS data is within similar ranges of values, this method of normalizing and standardizing allows the network to accept any self-consistent MRI data to be used as input regardless of the absolute range of values contained in the MRI. By removing the skull from the MRI scan, leaving a clean brain, there is reduced complexity for analysis of the training data. Failing to remove the skull and preprocess input data results in worse performance, longer training times, and misidentification of tumors in the brain. Despite the BRaTS dataset having relatively consistent MRI data, there is an issue of contrast varying from one patient to the next. The normalization and standardization help alleviate brightness differences, but they only exacerbate the difference in contrast. To solve this, each slice is processed through an adaptive histogram equalization algorithm that boosts contrast and highlights subtle features in the image. We aim to maintain uniform contrast between MRIs and expose details that the neural network may choose to pick up on. We use Keras for image augmentation to artificially generate more data from the MRI slices used for training and validating. There are configuration options for applying random rotations, horizontal and vertical shifts, shearing, and other augmentations. The slices are augmented in real-time on the CPU and then used in training on a single or multiple GPUs. This augmentation process greatly increases the diversity of images with which the network can use to train. Care was taken not to distort the images too greatly such that the brains become unrecognizable, as such augmentation could negatively impact training. C. Training The UNet is trained in batches of 75 images that are 1282 in size containing only a single grayscale channel. The images are chosen in a shuffled order for each epoch and augmented randomly from one epoch to the next by the preprocessing. The BRaTS dataset was used as input to the network after an 80/20 split for training and validation purposes. This split was chosen on a slice by slice basis as opposed to splitting patients by this ratio. This method of splitting gives the network portions of MRIs that can be distributed between both training and validation subsets. During the development phase of the project, the neural network was trained with a version of TensorFlow that supported SSE4.1, SSE4.2, FMA, and AVX instructions. Iterative training was performed on an Intel i7-7700HQ CPU using either 6 of the 8 logical cores, or all 8 logical cores when the computer was not in use. Fully-fledged training was performed 2 nodes of a cluster, each with 2 Nvidia GTX 1080 cards and 2 12-core Xeon E5-2670 CPUs. The cluster nodes 0 20 40 60 80 0.2 0.4 0.6 0.8 Epoch DSC training validation Fig. 2. Training and Validation Graph were only used for training once we saw the model training follow a favorable direction on a

personal computer. Due to the black box nature of neural networks, custom mini prediction code was written to provide rapid prediction previews both after each training step and epoch. A mini prediction is one or more slices of an MRI that are chosen manually as being representative of the one of the test brains. Predictions were performed on these slices only and shown to the user. These previews aided in visualizing what the network was doing and gave a sense of directionality that allowed further iterative changes to be made to the model. Metrics were also saved to a file after each training step to provide insight in to network performance quickly without having to wait for several epochs to discover a trend. Rough estimations of a potential trend could be made based on training step metrics and they were used to quickly change the model if unexpected behavior occurred. D. Evaluation Quick predictions, as mentioned, are generated optionally after each training step as well as after each epoch. These predictions were used as the first level of testing for our network model. Instead of splitting the valuable BRaTS2020 data between training, validation, and testing, we opted to use an entirely different dataset for testing. In theory, the robustness of our model would be determined by how well it handles a different dataset. A downside of testing with our clinical data is the lack of an expert provided ground truth for brain tumor segmentation. Our results data metrics a are therefore limited to using validation data from the BRaTS2020 dataset.

CHAPTER 4 : RESULTS AND DISCUSSION

The dataset was used for training: our data set and the BRaTS2020 dataset. Both datasets contain T1 weighted MRIs. The BRaTS2020 dataset contains image masks created by experts, while we manually masked our set. The use of two datasets as well as image augmentation reduces the chances of over-fitting the model. The 3D volumes are split into 2D slices along the axial orientation and the resulting images are resampled, standardized, and normalized so that the model is fed consistent input data. Patient MRIs are randomly chosen at an 80/20 split for training and validation and only image slices that contain tumors are used by the network. Due to the lack of ground truth data in the clinical dataset, the results shown rely on validation data from the BRaTS dataset. Twenty percent of the axial slices were used from validation only and the scores were calculated using the dice coefficient formula show before. After a short training session of only 18 epochs, and then another session of 6 epochs, the network scored 0.8029 on the validation subset (with a raw binary accuracy of 0.9915), details are show in table [1]. Upon further inspection, it would appear most of the error in the predictions come from false negatives near the bottom of the brain. We believe this is because the network has virtually no knowledge of cerebellum, which has a different texture and gradations of gray in it that the cerebral cortex. Most of the tumors marked in the BRaTS dataset are within the cerebral cortex, causing a lack of knowledge and throwing of the network with completely new data that the convolutional layers are likely unsure of what to do with. This training session only spanned a few hours, but the ADADELTA optimizer converged quickly on a solution in only 8-10 epochs. The validation score remained greater than the training score for nearly the entire time, which leads us to believe there is some improvement to be made to the model or preprocessing. Shown in figure 3 are four examples of predictions outputted by the neural network. The red outline shows the region the network selected as containing a tumor. Despite these images being just twodimensional slices, the network does save the volumetric segmentation alongside the original input MRI data, both resampled to the same cubic size. Most of the time, the networks segmentation is spot on. There are only a few cases of false negatives by the network, but they

occur on the very edge slices (top and bottom) of the tumor in some situations. The network also struggles to properly segment tumors that are both dark and near the edge of the brain.




```
    with torch.no_grad():
        fake = gen(fixed_noise)

        img_grid_real = make_grid(real.detach().cpu(), nrow = 8, pad_value = 255 , padding=2, normalize=True)
        img_grid_fake = make_grid(fake.detach().cpu(), nrow = 8, pad_value = 255 , padding=2, normalize=True)
        img_list_real.append(img_grid_real)
        img_list_fake.append(img_grid_fake)
```

```
Epoch [0/200] Batch 0/11                Loss D: 70324.8203, loss G: 0.9640
Epoch [0/200] Batch 1/11                Loss D: 2004.7880, loss G: -4.8542
Epoch [0/200] Batch 2/11                Loss D: 1138.0015, loss G: -9.2344
Epoch [0/200] Batch 3/11                Loss D: 2235.2551, loss G: -13.6432
Epoch [0/200] Batch 4/11                Loss D: 3263.6675, loss G: -18.3727
Epoch [0/200] Batch 5/11                Loss D: 83.7430, loss G: -23.1501
Epoch [0/200] Batch 6/11                Loss D: 8493.4131, loss G: -24.5885
Epoch [0/200] Batch 7/11                Loss D: 298.1227, loss G: -17.7471
Epoch [0/200] Batch 8/11                Loss D: 4.6574, loss G: -13.2524
Epoch [0/200] Batch 9/11                Loss D: 21.4367, loss G: -22.3237
Epoch [0/200] Batch 10/11                Loss D: 323.9487, loss G: -23.3495
Epoch [1/200] Batch 0/11                Loss D: 172.8365, loss G: -24.4021
Epoch [1/200] Batch 1/11                Loss D: 23.1582, loss G: -25.8164
Epoch [1/200] Batch 2/11                Loss D: 25.7704, loss G: -26.9055
Epoch [1/200] Batch 3/11                Loss D: 206.7379, loss G: -28.0657
Epoch [1/200] Batch 4/11                Loss D: 61.7243, loss G: -29.6945
Epoch [1/200] Batch 5/11                Loss D: 2659.1628, loss G: -28.5168
```

+ Code + Markdown

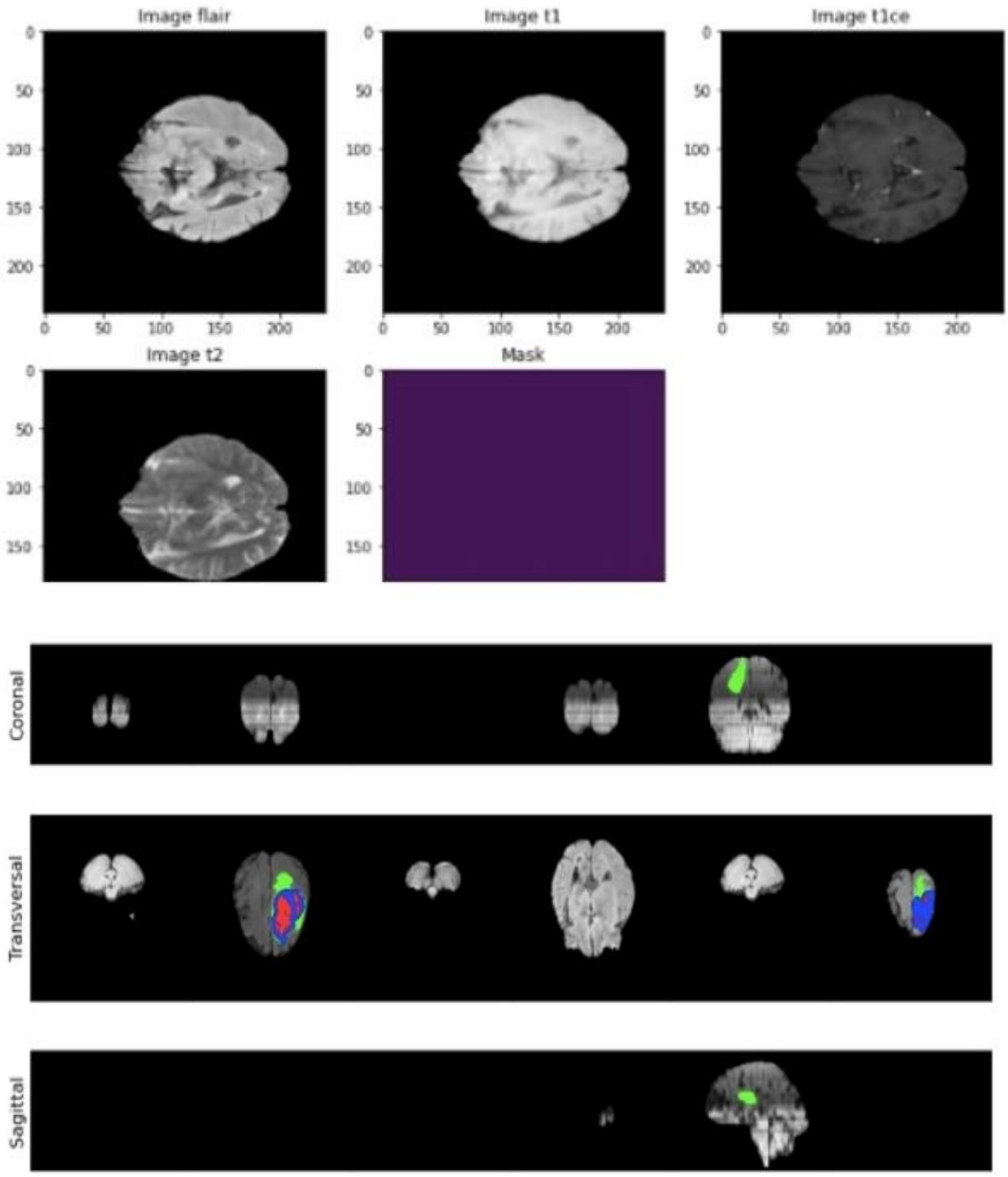
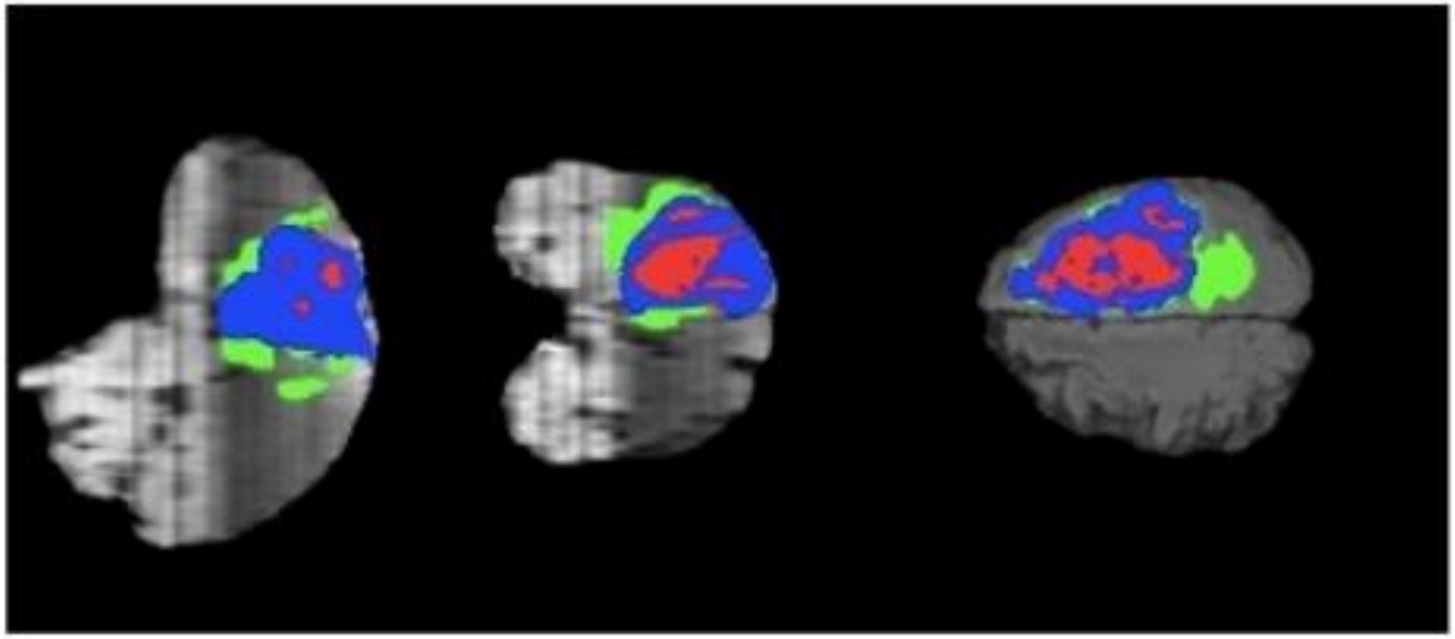


Image flair
Image t1
Image t1ce
Image t2
Mask
Coronal
Transversal
Sagittal

```
visualize_data_gif(labeled_image)
```



```
Epoch [0/200] Batch 0/11              Loss D: 16808.3867, loss G: -0.8526
Epoch [0/200] Batch 1/11              Loss D: 635.1337, loss G: -7.1956
Epoch [0/200] Batch 2/11              Loss D: 127.8195, loss G: -12.2072
Epoch [0/200] Batch 3/11              Loss D: 320.9958, loss G: -15.2158
Epoch [0/200] Batch 4/11              Loss D: 617.8706, loss G: -19.1845
Epoch [0/200] Batch 5/11              Loss D: 214.8277, loss G: -22.7162
Epoch [0/200] Batch 6/11              Loss D: 50.3683, loss G: -26.8755
Epoch [0/200] Batch 7/11              Loss D: 47.7347, loss G: -26.0926
Epoch [0/200] Batch 8/11              Loss D: 517.8479, loss G: -28.0716
Epoch [0/200] Batch 9/11              Loss D: 101.5571, loss G: -18.7752
Epoch [0/200] Batch 10/11              Loss D: 12.7327, loss G: -21.3932
```

The hybrid loss calculated here is the final result and with the help of the result we perform segmentation of brain tumor.

CHAPTER 5: CONCLUSION AND FUTURE SCOPE

Brain tumor detection is early to save the life of patients and It help reduce the cost of medical bills. the SVM classifier is suited both for unstructured and semi-structured datasets like images, texts are good. The SVM model does not affect overfitting. It consuming a lot of time for large datasets and training. the accuracy rate was 90% detection of the brain tumor.

Here an algorithm like CNN-based segmentation methods has been developed for the detection of brain tumor from MRI brain images by performing different operations like Edge Detection, Thresholding followed by segmentation. Here we are using two types of segmentation methods like edge-based segmentation and Region-based segmentation to segment or partition the digital image into multiple segments of pixels. By applying neural network algorithm for training with balanced classes and then refining it with proportions near the originals binary CNN to identify the complete tumor. In the future with improved algorithms more accuracy can be achieved in the results of detecting the tumors' size and stage of the cancer. This project can also be extended into finding other organ related diseases. With more data available in the future this project can be helpful for animals as well.

The aim of the present study was to design, implement, and evaluate a software pattern recognition system to improve classification accuracy between primary and metastatic brain tumors on MRI. Here, several existing brain tumor segmentation and detection methodology has been discussed for MRI of brain image. All the steps for detecting brain tumor have been

discussed including pre-processing steps. Pre-processing involves several operations like non local, Analytic correction methods, Markov random field methods and wavelet based methods has been discussed. Quality enhancement and filtering are important because edge sharpening, enhancement, noise removal and undesirable background removal are improved the image quality as well as the detection procedure. Among the different filtering technique discussed above, median filter suppressed the noise without blurring the edges and it is better outlier without reducing sharpness of the images, mean filter are much greater sensitive than that of median filter in the context of smoothing the image.

FUTURE SCOPE

Build an app-based user interface in hospitals which allows doctors to easily determine the impact of tumor and suggest treatment accordingly Since performance and complexity of Conv Nets depend on the input data representation we can try to predict the location as well as stage of the tumor from Volume based 3D images. By creating three dimensional (3D) anatomical models from individual patients, training, planning and computer guidance during surgery is improved.

Improve testing accuracy and computation time by using classifier boosting techniques like using more number images with more data augmentation, fine-tuning hyper parameters, training for a longer time i.e. using more epochs, adding more appropriate layers etc.. Classifier boosting is done by building a model from the training data then creating a second model that attempts to correct the errors from the first model for faster prognosis. Such techniques can be used to raise the accuracy even higher and reach a level that will allow this tool to be a significant asset to any medical facility dealing with brain tumors. For more complex datasets, we can use U-Net architecture rather than CNN where the max pooling layers are just replaced by upsampling ones. Ultimately we would like to use very large and deep convolutional nets on video sequences where the temporal structure provides very helpful information that is missing or far less obvious in static images. Unsupervised transfer learning may attract more and more attention in the future.

Quick predictions, as mentioned, are generated optionally after each training step as well as after each epoch. These predictions were used as the first level of testing for our network model.

Instead of splitting the valuable BRaTS2020 data between training, validation, and testing, we opted to use an entirely different dataset for testing. In theory, the robustness of our model would be determined by how well it handles a different dataset. A downside of testing with our clinical data is the lack of an expert provided ground truth for brain tumor segmentation. Our results data metrics are therefore limited to using validation data from the BRaTS2020 dataset.

REFRENCES

[1] Adam optimizer. URL https://github.com/tensorflow/tensorflow/ blob/master/tensorflow/python/training/adam.py#L39.

[2] Keras documentation. URL https://keras.io/preprocessing/ image/#imagedatagenerator-class.

[3] Giovanni Montana Adrien Payan. Predicting alzheimer's disease: a neuroimaging study with 3d convolutional neural networks, 2015. URL https://arxiv.org/abs/1502.02506.

[4] Geoffrey E. Hinton Alex Krizhevsky, Ilya Sutskever. Imagenet classification with dcnn, 2012. URL http://papers.nips.cc/paper/ 4824-imagenet-classification-with-deep-convolutional-neural-networks. pdf.

[5] Carole Sudre Lucas Fidon Dzhoshkun I. Shakir Guotai Wang Zach Eaton-Rosen-Robert Gray Tom Doel Yipeng Hu Tom Whyntie Parashkev Nachev Marc Modat Dean C. Barratt Sbastien Ourselin M. Jorge Cardoso Tom Vercauteren. Eli Gibson*, Wenqi Li*. "niftynet: a deep-learning platform for medical imaging." computer methods and programs in biomedicine, 158 (2018): 113-122., 2018. URL https://arxiv.org/pdf/1709. 03485.

[6] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[7] B. Devkota, Abeer Alsadoon, P.W.C. Prasad, A. K. Singh, A. Elchouemi, "Image Segmentation for Early Stage Brain Tumor Detection using Mathematical Morphological Reconstruction," 6th International Conference on Smart Computing and Communications, ICSCC 2017, 7-8 December 2017, Kurukshetra, India.

[8] Song, Yantao & Ji, Zexuan & Sun, Quansen & Yuhui, Zheng. (2016). “A Novel Brain Tumor Segmentation from Multi-Modality MRI via A Level-Set-Based Model”. Journal of Signal Processing Systems. 87. 10.1007/s11265- 016-1188-4.

[9] Ehab F. Badran, Esraa Galal Mahmoud, Nadder Hamdy, “An Algorithm for Detecting Brain Tumors in MRI Images”, 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence, 2017.

[10] Pei L, Reza SMS, Li W, Davatzikos C, Iftekharuddin KM. “Improved brain tumor segmentation by utilizing tumor growth model in longitudinal brain MRI”. Proc SPIE Int Soc Opt Eng. 2017.

[11] Dina Aboul Dahab, Samy S. A. Ghoniemy, Gamal M. Selim, “Automated Brain Tumor Detection and Identification using Image Processing and Probabilistic Neural Network Techniques”, IJIPVC, Vol. 1, No. 2, pp. 1-8, 2012

[12] Adrinne M. Mendrik Nal Kalchbrenner, Edward Grefenstette. A convolutional neural network for modelling sentences, 2014. URL https://arxiv.org/pdf/1404.2188.pdf.

[13] Soeren S. Lienkamp Thomas Brox Ozgun Cicek, Ahmed Abdulkadir and Olaf Ronneberger. 3d u-net: Learning dense volumetric segmentation from sparse annotation, 2016. URL https://lmb.informatik.uni-freiburg.de/Publications/2016/ CABR16/cicek16miccai.pdf.

[14] Phil Blunsom Linda S. de Vries Manon J. N. L. BendersIvana Igum Pim Moeskops, Max A. Viergever. Automatic segmentation of mr brain images with a cnn, 2016. URL https://ieeexplore.ieee.org/document/7444155/.

[15] Ujjwalkarn. An intuitive explanation of convolutional neural networks, 2017. URL ujjwalkarn.me/2016/08/11/ intuitive-explanation-convnets.

[16] Classification of Brain Tumors: https://www.aans.org/en/Media/ Classifications-of-Brain-Tumors. Access on 29 June 2020.

[17] A PADMA and R.Sukanesh, "Automatic Classification and Segmentation of Brain Tumor in CT Images using Optimal Dominant Gray level Run length Texture Features" International Journal of Advanced Computer Science and Applications(IJACSA), 2(10), pp. 53- 59, 2011.

[18] A. Hamamci, N. Kucuk, K. Karaman, K. Engin, and G. Unal, "TumorCut: Segmentation of Brain Tumors on Contrast Enhanced MR Images for Radiosurgery Applications," in IEEE Transactions on Medical Imaging, vol. 31, no. 3, pp. 790-804, March 2012.

[19] Aneja, Deepali, and Tarun Kumar Rawat. "Fuzzy Clustering Algorithms for Effective Medical Image Segmentation." International Journal of Intelligent Systems and Applications, Vol. 5, pp. 55-61, 2013.

[20] R. Sonavane and P. Sonar, "Classification and segmentation of brain tumor using Adaboost classifier," 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC), Jalgaon, pp. 396-403, 2016.

[21] V. Wasule and P. Sonar, "Classification of brain MRI using SVM and KNN classifier," 2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS), Chennai, pp. 218-223, 2017.

[22] M. M. Saleck, A. ElMoutaouakkil, and M. Mouçouf, "Tumor Detection in Mammography Images Using Fuzzy C-means and GLCM Texture Features," 2017 14th International Conference on Computer Graphics, Imaging and Visualization, Marrakesh, pp. 122-125, 2017

[23] M. H. O. Rashid, M. A. Mamun, M. A. Hossain, and M. P. Uddin, "Brain Tumor Detection Using Anisotropic Filtering, SVM Classifier and Morphological Operation from MR Images,"

2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2), Rajshahi, pp. 1-4, 2018.

[24] Ren, T., Wang, H., Feng, H., Xu, C., Liu, G., & Ding, P, "Study on the improved fuzzy clustering algorithm and its application in brain image segmentation," Appl. Soft Comput., Vol. 81, pp. 1-9, 2019. [10] Preetham Kumar and Vijayakumar B, "Brain Tumour MR Image Segmentation and Classification Using by PCA and RBF Kernel Based Support Vector Machine," J Middle-East Journal of Scientific Research, Vol. 23, Issue 9, pp.2106-2116, 2015.

[25] S. T. Kebir and S. Mekaoui, "An Efficient Methodology of Brain Abnormalities Detection using CNN Deep Learning Network," 2018 International Conference on Applied Smart Systems (ICASS), Medea, Algeria, pp. 1-5, 2018.

[26] M. Talo, U. B. Baloglu, O. Yıldırım, & U. Rajendra Acharya, "Application of deep transfer learning for automated brain abnormality classification using MR images," Cognitive Systems Research, Vol. 54, pp. 176-188, 2019.