Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 16, Issue. 1: 2025
ISSN : **1906-9685**

AUTOMATED SEGMENTATION OF BRAIN TUMOUR MRI IMAGES USING DEEP LEARNING

CHAPTER 1

INTERDUCTION

Brain tumours can be classified into two types: benign (noncancerous) and malignant (cancerous). The malignant tumours can quickly spread to other tissues in the brain and lead to worsening the patient's condition . When most of the cells are old or damaged, they are destroyed and replaced by new cells. if damaged and old cells are not eliminated with generating the new cells, it can cause problems. The production of additional cells often results in the formation of a mass of tissue, which refers to the growth or tumour. Brain tumour detection is very complicated and difficult due to the size, shape, location and type of tumour in the brain. Diagnosis of brain tumours in the early stages of the tumour's start is difficult because it cannot accurately measure the size and resolution of the tumour . However, if the tumour is diagnosed and treated early in the tumour formation process, the chance of patient's treatment is very high. Therefore, the treatment of tumour depends on the timely diagnosis of the tumour .

Abstract:

A brain tumour is understood by the scientific community as the growth of abnormal cells in the brain, some of which can lead to cancer. The traditional method to detect brain tumours is nuclear magnetic resonance (MRI). Having the MRI images, information about the uncontrolled growth of tissue in the brain is identified Brain tumour detection is done through the application of Machine Learning and Deep Learning algorithms. VGG16 are applied in the detection of the presence of brain tumour, and its performance is analyzed through different metricsIn proposed on LSTM Algorithm The aggressive nature and diversity of gliomas, well-organized and exact segmentation methods used to classify tumours. Merging the result of two separate segmentation networks the method demonstrates a major but simple combinational strategy. By the whole tumour enhanced tumour, and tumour core will define the validation set. Segmenting brain tumours by using MR data for disease investigation and monitoring. Segmenting brain tumours automatically using MR data is crucial for disease investigation and monitoring. Due to the aggressive nature and

diversity of gliomas, well-organized and exact segmentation methods are used to classify tumours intra-tumourally.

Existing System:

• Many methods have achieved good classification results in the classification of images. This Existing review of the classification methods of images from three aspects supervised classification, semi-supervised classification, and unsupervised classification • It is based on the CNN Algorithm Automated segmentation of brain tumour MRI images using deep learning is a significant area of research in medical image analysis. Several existing systems and approaches have been developed to address this challenge. • Acquire a dataset of brain MRI images with corresponding ground truth annotations for tumour regions. Databases like BRATS (Multimodal Brain Tumour Segmentation Challenge) are commonly used. • Perform preprocessing on the MRI images to enhance features and standardize the input data. This may involve skull stripping, intensity normalization, and image registration. Fig. 3.1 Brain Tumour Dataset 8 • Glioma is a type of tumour that originates in the glial cells of the central nervous system. Glial cells provide support and protection for nerve cells, and tumours that arise from these cells are called gliomas. Gliomas can occur in the brain or the spinal cord. They are the most common type of primary brain tumours. • A meningioma is a type of tumour that arises from the meninges, which are the layers of tissue covering the brain and spinal cord. Meningiomas are typically slow-growing tumours and are usually benign (non-cancerous), although they can, in some cases, exhibit more aggressive behavior. • The pituitary gland is a small, pea-sized gland located at the base of the brain, just below the hypothalamus, in a bony structure called the sella turcica. Despite its small size, the pituitary gland plays a crucial role in regulating various physiological processes in the body by producing and releasing hormones.

Existing System Disadvantages:

1. Data Limitations

- **Insufficient annotated datasets:** High-quality, manually segmented brain tumour MRI datasets are limited.
- **Data imbalance:** Some tumour types or grades are underrepresented, leading to biased models.
- **Generalizability issues:** Models trained on a specific dataset (e.g., BRATS) may not perform well on data from other sources or scanners.

2.Model Limitations

- **Overfitting:** Deep learning models, especially with limited training data, tend to overfit.

- **Lack of robustness:** Performance may degrade significantly with noise, motion artifacts, or different MRI protocols.
- **High computational requirements:** Training and deploying deep models require powerful GPUs and significant memory2.

3. Segmentation Accuracy

- **Boundary ambiguity:** Tumour borders are often irregular or diffuse, leading to inaccurate segmentations.
- **Poor performance on small or infiltrative tumours:** Deep models may miss smaller or less distinct regions.

4. Interpretability and Explainability

- **Black-box nature:** Most deep learning models do not provide interpretable reasons for their predictions.
- **Lack of clinician trust:** Without explainable results, clinicians may hesitate to adopt AI tools in practice.

5. Clinical Integration Challenges

- **Regulatory and validation issues:** AI tools require rigorous validation before clinical use.
- **Lack of standardization:** Different hospitals use different imaging protocols, which complicates deployment.
- **Workflow disruption:** Integration into existing clinical workflows is not seamless and often requires additional effort.

6. Training Time and Costs

- **Extensive training time:** Models can take days to train.
- **Resource intensive:** Requires significant computational resources and expert tuning.

Proposed System:

• On the other hand, LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN) commonly used for sequential data, such as time series. • Using LSTM for predicting tumour progression or changes over time in a sequence of MRI images, could be a valid approach. • Calculating deep features using Convolutional Neural Networks (CNNs) involves extracting the activations from intermediate layers of the network, typically before the final classification layer. These intermediate layer activations capture hierarchical and abstract representations of input images, making them valuable features for various computer vision tasks. • Adapt the ResNet-50 architecture for semantic segmentation by converting fully

connected layers into convolutional layers. Train the modified ResNet-50 on the brain MRI dataset for tumour segmentation.

ALGORITHM

• Design the architecture with encoder and decoder components. U-Net, SegNet, and DeepLab are popular architectures for medical image segmentation. • Define a loss function that measures the difference between the predicted segmentation and the ground truth. • Train the model using the training set, validating on the validation set to monitor performance and prevent overfitting. • Post-processing techniques to refine the segmentation results. This may include smoothing, removing small connected components, or other morphological operations.

Advantages of the Proposed System

• High performance
• CNN is improving the accuracy using Liver images.
• Less time duration
• Early detection of tumours

CHAPTER 2

LITERATURE SURVEY:

Title: Revalence of Autism Spectrum Disorder Among Children Aged 8 Years

Year: 2014

Author: N.V. Ramana Murty and Prof. M.S. Prasad Babu

Methodology:

The Autism and Developmental Disabilities Monitoring (ADDM) Network is an active surveillance system that provides estimates of the prevalence of autism spectrum disorder (ASD) among children aged 8 years whose parents or guardians reside within 11 ADDM sites in the United States (Arizona, Arkansas, Colorado, Georgia, Maryland, Minnesota, Missouri, New Jersey, North Carolina, Tennessee, and Wisconsin). ADDM surveillance is conducted in two phases. The first phase involves review and abstraction of comprehensive evaluations that were completed by professional service providers in the community. Staff completing record review and abstraction receive extensive training and supervision and are evaluated according to strict reliability standards to certify effective initial training, identify ongoing training needs, and ensure adherence to the prescribed methodology. Record review and abstraction occurs in a variety of data sources ranging from general pediatric health clinics to specialized programs serving children with developmental disabilities. In addition, most of the ADDM .

Advantage

• In direct outgrowth of this study was a favourable recommendation for CT-based Brain Tumour screening by several prestigious organizations.

Disadvantages

• Administrative costs to be high

• Lack of Real-Time Data.

Title: Development of functional and structural connectivity

Year: 2016

Author: Samy S. Abu Naser, Bashar G. Bastami

Methodology:

Functional and structural maturation of networks comprised of discrete regions is an important aspect of brain development. The default-mode network (DMN) is a prominent network which includes the posterior cingulate cortex (PCC), medial prefrontal cortex (MPFC), medial temporal lobes (MTL), and angular gyrus (AG). Despite increasing interest in DMN function, little is known about its maturation from childhood to adulthood. Here we examine developmental changes in DMN connectivity using a multimodal imaging approach by combining resting-state fMRI, voxel-based morphometry and diffusion tensor imaging-based tractography. We found that the DMN undergoes significant developmental changes in functional and structural connectivity, but these changes are not uniform across all DMN nodes. Convergent structural and functional connectivity analyses suggest that PCC-mPFC connectivity along the cingulum bundle is the most immature link in the DMN of children. Both PCC and mPFC also showed gray matter volume differences, as well as prominent macrostructural and microstructural differences in the dorsal cingulum bundle linking these regions. Notably, structural connectivity between PCC and left MTL was either weak or non-existent in children, even though functional connectivity did not differ from that of adults.

Advantages

• Still effective in cases where number of dimensions is greater than the number of samples.

• Uses a subset of training points in the decision function so it is also memory efficient.

Disadvantage

• It is time complexity being high, and not suitable for large-scale data.

Comparsion Table:

Method/Model	Architecture	Input Modality	Dataset Used	Accuracy / Dice Score	Strengths	Limitations
U-Net	Encoder-Decoder CNN	T1, T2, FLAIR,	BraTS	~0.85–0.90 (Dice)	Simple, effective, low computational	Struggles with small lesions

Method/Model	Architecture	Input Modality	Dataset Used	Accuracy / Dice Score	Strengths	Limitations
		T1c			cost	
3D U-Net	3D CNN	Full 3D volumes	BraTS	~0.88 (Dice)	Captures volumetric context	High memory usage
V-Net	Residual 3D CNN	Multi-modal MRI	BraTS	~0.87–0.89 (Dice)	Good for 3D data, deep network	Slower training
DeepMedic	Multi-scale 3D CNN	Multi-modal MRI	BraTS	~0.85–0.88 (Dice)	Handles class imbalance well	Complex architecture
Attention U-Net	U-Net + Attention	T1, T2, FLAIR, T1c	BraTS	~0.89 (Dice)	Focuses on relevant areas	More parameters, longer training time
nnU-Net	Self-configuring U-Net	Multi-modal MRI	BraTS	~0.90–0.91 (Dice)	Auto-optimizes preprocessing & training	Requires powerful hardware
TransUNet	Transformer + U-Net	T1, T2, FLAIR, T1c	BraTS	~0.91 (Dice)	Strong global context modeling	High computational cost
Swin-UNet	Swin Transformer + U-Net	Multi-modal MRI	BraTS	~0.90–0.92 (Dice)	Hierarchical attention, efficient	Complex implementation

CHAPTER 3

SYSTEM ANALYSIS

Introduction:

The segmentation of brain tumours in Magnetic Resonance Imaging (MRI) is a critical task in medical image analysis, with direct implications for diagnosis, treatment planning, and

monitoring. Traditionally, radiologists manually delineate tumour regions, a process that is time-consuming, prone to human error, and highly dependent on the expertise of the clinician. With the rise of advanced imaging techniques and the exponential growth of medical data, there is a growing need for automated, accurate, and efficient segmentation tools.

This project focuses on the development of an **automated brain tumour segmentation system** using **deep learning techniques**, which have shown promising performance in complex pattern recognition tasks, particularly in image classification and segmentation. Deep learning models, especially convolutional neural networks (CNNs), have the ability to learn spatial hierarchies of features directly from raw image data without the need for manual feature extraction. This makes them particularly suitable for identifying and segmenting abnormal tissue regions in brain MRIs.

The proposed system aims to process MRI images, automatically detect brain tumours, and segment the affected regions with high accuracy. The system will be trained and validated on publicly available annotated MRI datasets (e.g., BraTS), enabling it to generalize well across different cases. The automation of this task not only reduces the workload of radiologists but also enhances the consistency and speed of diagnosis.

The system analysis phase explores the functional and non-functional requirements, existing manual and semi-automated systems, the feasibility of implementation, and the architecture best suited for integrating deep learning models into clinical workflows. This lays the foundation for a robust and reliable AI-based medical tool that can support healthcare professionals in delivering faster and more accurate patient care.

CONCLUSION:

In DL a LSTM is a class of DL , most commonly applied to analyse visual imagery. Now when we think of a neural network we think about matrix multiplications but that is not the case with ConvNet. It uses a special technique called Convolution. Now in mathematics convolution is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other. Multimodal MRI brain tumour image segmentation task, segmenting the entire tumour and tumour core area, enhanced tumour area from normal brain tissue. The research on computer-aided diagnosis and treatment of multimodal MRI brain tumour image segmentation has always been an important topic in the field of medical image processing. The difference in imaging equipment and imaging conditions will cause even the same patient in the same period to have a different MRI with different properties.

Functional Requirements:

1. **Image Input & Preprocessing**

- The system shall accept MRI brain images in common formats (e.g., DICOM, PNG, JPEG).
- The system shall normalize and resize images for consistent input to the deep learning model.
- The system shall perform preprocessing tasks such as skull stripping and noise removal.

2. **Tumor Segmentation**
   - The system shall use a trained deep learning model (e.g., U-Net, V-Net, or similar) to segment tumor regions.
   - The system shall identify and delineate different tumor subregions (e.g., edema, necrotic core, enhancing tumor).
3. **Output Generation**
   - The system shall produce a binary or multi-class mask that highlights tumor regions.
   - The system shall overlay the segmentation mask on the original image for visualization.
   - The system shall allow exporting the segmented images in standard formats.
4. **Performance Metrics Calculation**
   - The system shall compute accuracy metrics such as Dice coefficient, Jaccard index, sensitivity, specificity, etc.
5. **User Interface (if applicable)**
   - The system shall provide a graphical user interface (GUI) or web-based interface for users to upload images and view results.
   - The system shall allow users to zoom in/out and navigate MRI slices (in the case of 3D data).
6. **Model Training (if training is part of the system)**
   - The system shall allow training on annotated MRI datasets.
   - The system shall support data augmentation techniques to enhance model generalization.
   - The system shall save and load trained models.
7. **Error Handling**
   - The system shall notify the user if the input format is incorrect or the image is unreadable.
   - The system shall handle exceptions during inference and training gracefully.

Non- Functional Requirements:

1.Performance

- The system should process and segment an MRI scan in under 10 seconds (depending on hardware).
- The system should support batch processing for multiple images.

2. **Accuracy**
   - The segmentation model should achieve a Dice coefficient of at least 0.85 on test data.
3. **Scalability**
   - The system should be scalable to handle larger datasets and higher-resolution images.
   - The architecture should allow easy integration with PACS or hospital systems (if required).
4. **Usability**
   - The interface should be user-friendly for radiologists or medical staff with minimal training.
   - The results should be interpretable and visually clear.
5. **Maintainability**
   - The code should be modular and well-documented for future updates and model improvements.
   - The system should allow easy re-training or fine-tuning with new data.
6. **Security and Privacy**
   - The system should comply with healthcare data privacy standards (e.g., HIPAA, GDPR).
   - Patient data should be anonymized and securely handled.
7. **Portability**
   - The system should be deployable on both local machines and cloud platforms (e.g., AWS, GCP).
   - It should support containerization (e.g., Docker) for ease of deployment.
8. **Reliability**
   - The system should function consistently under expected workloads and provide consistent outputs.

SYSTEM REQUIREMENTS:

HARDWARE REQUIREMENTS:

• System : Pentium IV 2.4 GHz

• Hard Disk : 200 GB

• Mouse : Logitech.

• Keyboard : 110 keys enhanced

• RAM : 4GB

SOFTWARE REQUIREMENTS:

• O/S : Windows 11

• Language : Python

• Front End : Anaconda Navigator – Spyder

CHAPTER 4

SYSTEM DESIGN

SOFTWARE DESCRIPTION:

Python

Python is one of those rare languages which can claim to be both simple and powerful. You will find yourself pleasantly surprised to see how easy it is to concentrate on the solution to the problem rather than the syntax and structure of the language you are programming in. The official introduction to Python is Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

Features of Python Simple

Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English, although very strict English! This pseudo-code nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the language itself.

Easy to Learn

As you will see, Python is extremely easy to get started with. Python has an extraordinarily simple syntax, as already mentioned.

Free and Open Source

Python is an example of a FLOSS (Free/Libré and Open Source Software). In simple terms, you can freely distribute copies of this software, read its source code, make changes to it, and use pieces of it in new free programs. FLOSS is based on the concept of a community which shares knowledge. This is one of the reasons why Python is so good - it has been created and is constantly improved by a community who just want to see a better Python.

High-level Language

When you write programs in Python, you never need to bother about the low-level

details such as managing the memory used by your program, etc.

Portable
Due to its open-source nature, Python has been ported to (i.e. changed to make it work on) many platforms. All your Python programs can work on any of these platforms without requiring any changes at all if you are careful enough to avoid any system-dependent features. You can use Python on GNU/Linux, Windows, FreeBSD, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, and # -*- coding: utf-8 -*-z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, VX Works, PlayStation, Sharp Zaurus, Windows CE and PocketPC! You can even use a platform like Kivyto create games for your computer and for iPhone, iPad, and Android.

Interpreted
A program written in a compiled language like C or C++ is converted from the source language i.e. C or C++ into a language that is spoken by your computer (binary code i.e. 0s and 1s) using a compiler with various flags and options. When you run the program, the linker/loader software copies the program from hard disk to memory and starts running it. Python, on the other hand, does not need compilation to binary. You just run the program directly from the source code. Internally, Python converts the source code into an intermediate form called byte codes and then translatesthisinto the native language of your computer and then runs it. All this, actually, makes using Python much easier since you don't have to worry about compiling the program, making sure that the proper libraries are linked and loaded, etc. This also makes your Python programs much more portable, since you can just copy your Python program onto another computer and it just works!

Object Oriented
Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In object oriented languages, the program is built around objects which combine data and functionality.

ARCHITECTURAL DESIGN

We suggest an MRI image-based BT classification technique that is completely automated. The specified objective is to categorize three different forms of cancers in brain scans, including gliomas and pituitary tumours, using a three class classification problem. These three BT types are the most prevalent. A DCNN with a U-Net sampling model is used for classification, as well as to extract image features. The open dataset from share is used in the evaluation. Our work is motivated by the following factors, First, greater accuracy in the categorization problem involving meningiomas, gliomas, and endocrine tumours is expected. Medical professionals' treatment regimens would benefit from a precise, computer aided automatic classification approach for the three different types of tumours.

Second, recent classification challenges that employed DL techniques and reliable classifiers were successful in producing highly accurate results. Medical imaging data are hard to attain, which is the third issue. To overcome this practical restriction, advanced design techniques are needed.

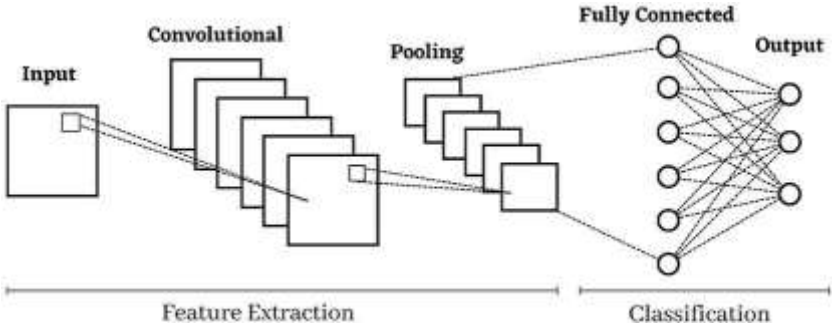


Fig.5.1.1 System Design

BRAIN TUMOUR DETECTION

- The following are the contributions made by this work:
- To extract characteristics from brain MRI images containing tumours, a CNN model was created.
- On datasets containing medical images, it was discovered that the CNN layout produced better classification results.
- The feature maps are produced after numerous convolutional layers have extracted features from the input images.
- The BraTS dataset was used to create a model with 98% overall accuracy;
- Regarding computational complexity, a comparison between the proposed approach and a transfer learning-based strategy is given.

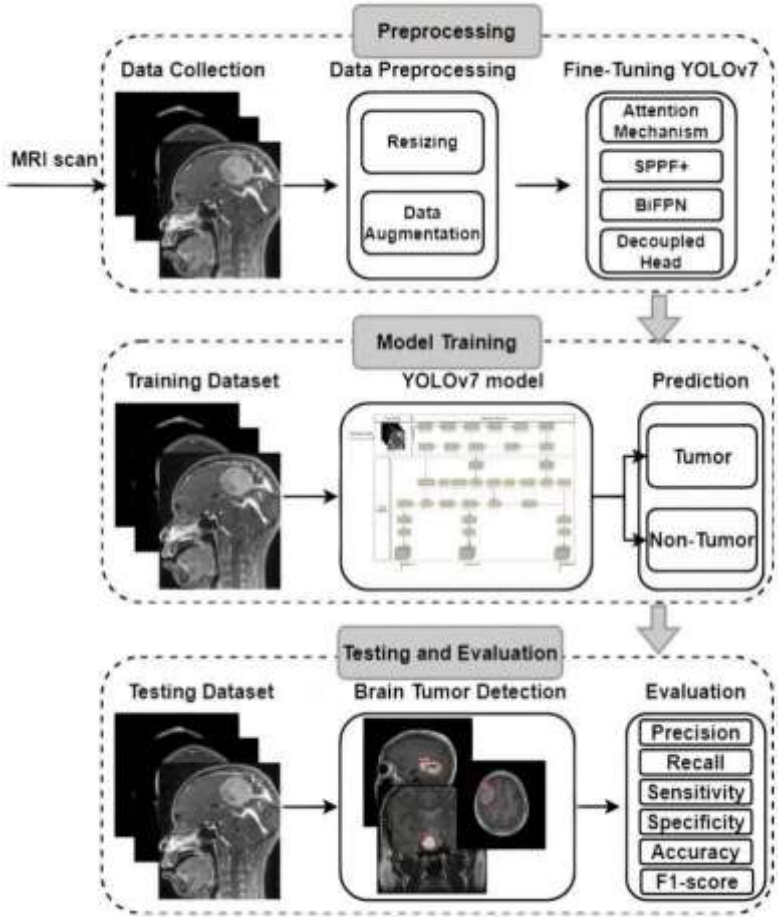


Fig. 5.1.2 Dataset Collection

- Dataset Collection Model training is a crucial step in the development of machine learning models.
- It involves teaching a model to recognize patterns and make predictions by exposing it to a labeled dataset.
- Testing and evaluation are crucial steps in assessing the performance and generalization ability of a trained machine learning model. These steps help determine how well the model is likely to perform on new, unseen data.
- The training dataset is a subset of a larger dataset that is used to train a machine learning model. It consists of input-output pairs, where the inputs are the features or attributes of the data, and the outputs are the corresponding labels or target values. During the training phase, the model learns patterns and relationships from this dataset, adjusting its internal parameters to make accurate predictions.
- YOLOv7 (You Only Look Once version 7) was not officially released or recognized as a standard version of the YOLO (You Only Look Once) object detection model. The most recent official release was YOLOv4, which was developed by Alexey Bochkovskiy and the YOLO community.
- The YOLOv7 model's backbone network is made up of the convolution, batch normalization, and SiLU (CBS) module, the MaxPool1 (MP1) module, and the extended efficient layer aggregation network (E-ELAN).
- YOLOv7′s model architecture consists of two primary parts: the Backbone

network and the head network. The raw input image is preprocessed in order to prepare it for input into the Backbone network.

- SPPF+ is an improvement on the existing Spatial Pyramid Pooling with Fusion (SPPF) architecture. SPPF+ improves upon the performance of SPPF in object detection by using the feature reuse concept from SPPF and Convolutional Spatial Pyramid Convolutional (SPPCSPC).

CHAPTER 5

IMPLEMENTATION

LIST OF MODULES

- Brain Image
- Segmentation
- Splitting Dataset into Train and Test Data
- Classification
- Prediction
- Result Generation

MODULES DESCRIPTION

BRAIN IMAGE

- The data selection is the process of selecting the data for Brain Image dataset.
- In this project, detect the Brain Tumour .The dataset which contains the information about the Brain grayscale images.
- MRI scan images are included in this collection, since they are the gold standard for diagnosing brain tumours. Glioma (2548 images), pituitary (2658 images), meningioma (2582 images), and no tumour (2500 images) were the foursubsetsthat made up our dataset of brain tumours.

SEGMENTATION

- ResNet-50 is a convolutional neural network (CNN) architecture that belongs to the ResNet (Residual Network) family. It was introduced by Microsoft Research in the paper "Deep Residual Learning for Image Recognition" by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, published in 2016. ResNet-50 is one of several variants proposed in the ResNet 18

SPLITTING DATASET INTO TRAIN AND TEST DATA

- Data splitting is the act of partitioning available data into two portions, usually for cross-validate purposes.
- One Portion of the data is used to develop a predictive model and the other to

evaluate the model's performance.
 Separating image data into training and testing sets is an important part of evaluating image processing models.

CLASSIFICATION

In Deep learning, a convolutional neural network **(**CNN/ConvNet**)** is a class of deep neural networks, most commonly applied to analyse visual imagery. Now when we think of a neural network we think about matrix multiplications but that is not the case with ConvNet.. Now in mathematics convolution is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other

PREDICTION

 It’s a process of predicting the Brain Tumour from the dataset.
 This project will effectively predict the data from dataset by enhancing the performance of the overall prediction results.
 If it starts with True then the prediction was correct whether diabetic or not, so true positive is a diabetic person correctly predicted & a true negative is a healthy person correctly predicted.
 Oppositely, if it starts with False then the prediction was incorrect, so false positive is a healthy person incorrectly predicted as diabetic(+) & a false negative is a diabetic person incorrectly predicted as healthy(-).
 Positive or negative indicatesthe output of our program. While true orfalse judges this output whether correct or incorrect.19

RESULT GENERATION

The Final Result will get generated based on the overall classification and prediction.
The performance of this proposed approach is evaluated using some measures like,

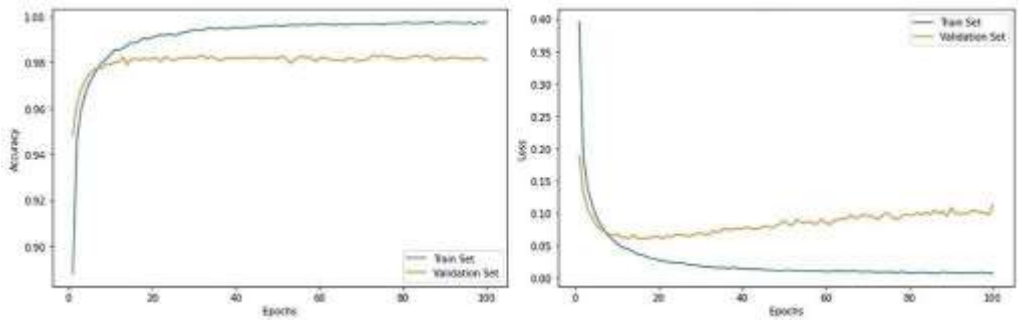


Fig.6.2 Evaluation

Different evaluation metrics are used for the prediction and classification

challenges, such as accuracy, precision, recall, and F1-measure. The effectiveness of the proposed model is assessed using the following evaluation metrics.

Accuracy:

To measure the accuracy of the proposed model, compute the ratio of the false positive, true positive, true negative, and false negative. The Equation represent the accuracy estimate.
Accuracy=TP+TNTP+TN+FP+FN

Precision:

The ratio of real positives to all positives overall (both false and true) in the data. Additionally referred to as a high predicted value. The precision rate is represented in Equation .
Precision=TPTP+FP20

Recall:

The ratio of the true positives in the data to true positives and false negatives is often referred to as sensitivity, the chance of detection, and the rate of a true positive. The recall rate is represented in Equation .
Recall=TPTN+FN

F1-measure:

The weighted average of the precision and recall is known as the F1 measure. The Equation represents the value of F1-measure.
F1−measure=2×Precision+RecallPrecision+Recall
Locate the files of the test highlight vectors that are closest to our own and extract their names from their markings. This information is used to calculate the likelihood of each mark appearing at each point in the framework based on the available data. It follows that a straight line of marked probability is interspersed all across the whole image. CNN seems to be more successful in segmenting tumours in experiments involving tumour augmentation, according to both qualitative and quantitative data. On the other hand, the U-Net is more precise than other methods in regard to the segmentation of tumour cores. Although both networks do equally well in regard to detecting the whole tumour segmentation on their own, the expectations are taken into account. The most current ensemble projections for the following three areas were generated only using U-output net models, for the purposes of the enhanced tumour, tumour core, and whole tumour, we observed the output.

CONCLUSION AND FUTURE ENHANCEMENT

CONCLUSION

In DL a LSTM is a class of DL , most commonly applied to analys visual imagery. Now when we think of a neural network we think about matrix multiplications but that is not the case with Convert Net. It uses a special technique called Convolution. Now in mathematics convolution is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.
Multimodal MRI brain tumour image segmentation task, segmenting the entire tumour and tumour core area, enhanced tumour area from normal brain tissue. The research on computer-aided diagnosis and treatment of multimodal MRI brain tumour image segmentation has always been an important topic in the field of medical image processing. The difference in imaging equipment and imaging conditions will cause even the same patient in the same period to have a different MRI with different properties.

FUTURE ENHANCEMENT

In the future, different dataset will be investigated to check the system robustness and more CNN models will be tested to improve the performance and identification of smallest tumours
Incorporate information from multiple imaging modalities (such as T1-weighted, T2 weighted, and FLAIR) to improve segmentation accuracy. Multi modal fusion can provide a more comprehensive understanding of the tumour characteristics. Continuous collaboration between researchers, clinicians, and machine learning experts is essential for addressing these challenges and making progress in automated brain tumour segmentation using deep learning

APPENDIX

SOURCE CODE

Load Modules

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

Prepare/collect data

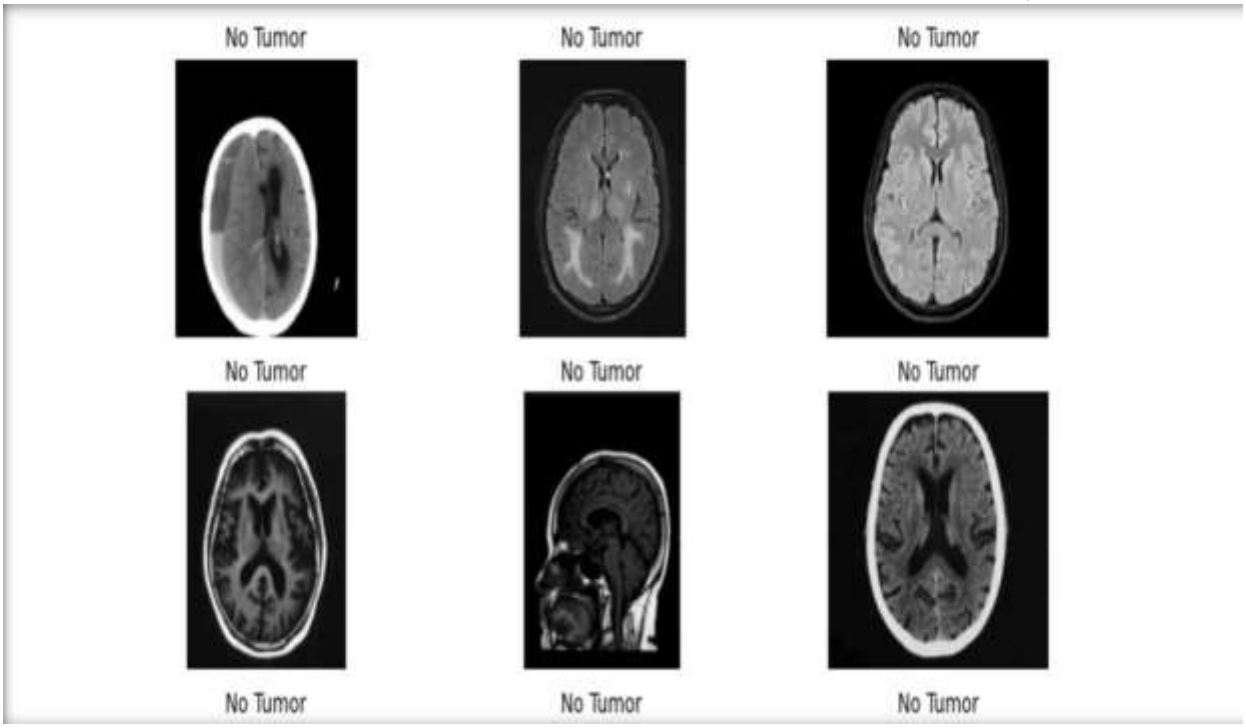
```
import os
path = os.listdir('brain_tumour/Training/')
classes = {'no_tumour':0, 'pituitary_tumour':1}
import cv2
X = []
Y = []
```

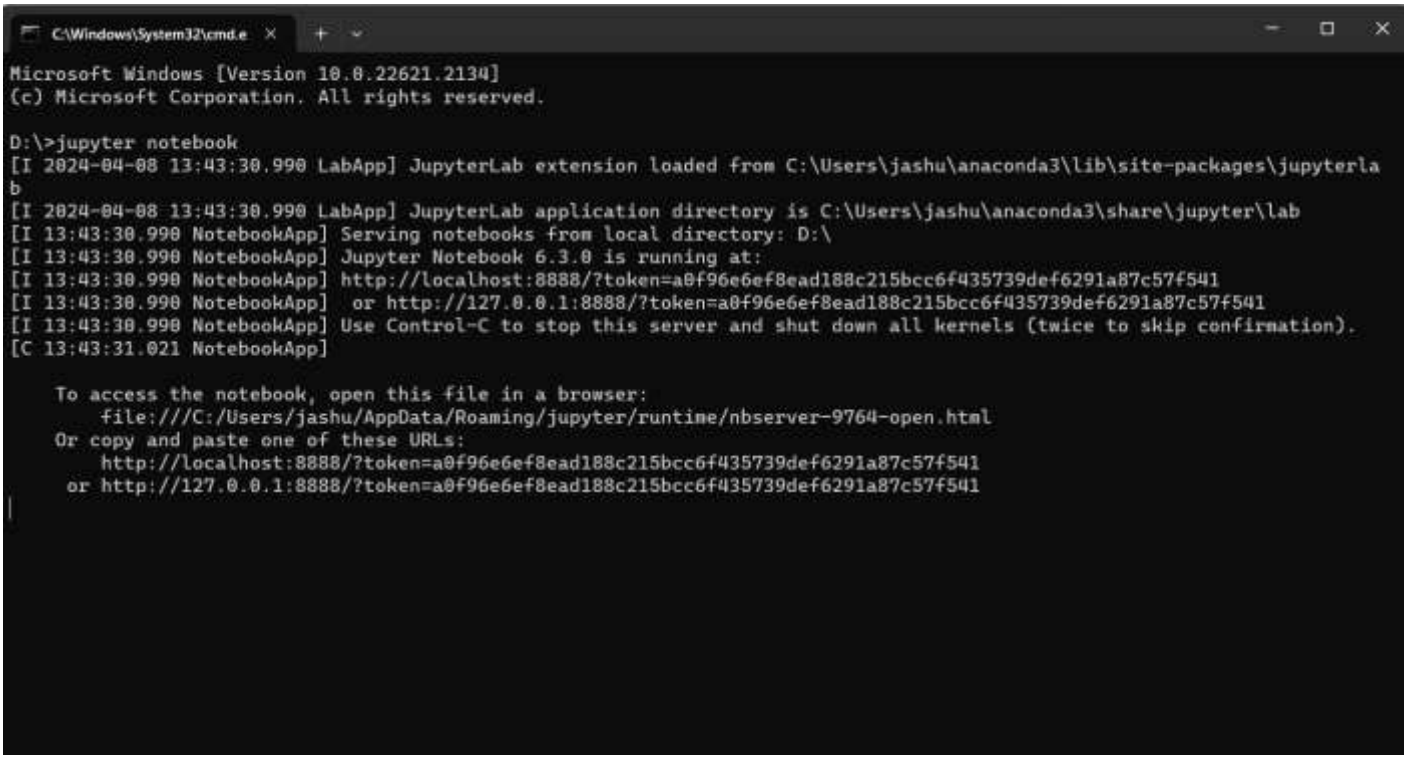
```
for cls in classes:
pth = 'brain_tumour/Training/'+cls
for j in os.listdir(pth):
img = cv2.imread(pth+'/'+j, 0)
img = cv2.resize(img, (200,200))
X.append(img)
Y.append(classes[cls])
pip install opencv-python
Requirement already satisfied: opencv-python in c:\users\jashu\anaconda3\lib\site-pac
K ages (4.9.0.80)
Requirement already satisfied: numpy>=1.17.3 in c:\users\jashu\anaconda3\lib\site-pa
ckages (from opencv-python) (1.20.1)
Note: you may need to restart the kernel to use updated packages.
X = np.array(X)
Y = np.array(Y)
X_updated = X.reshape(len(X), -1)
np.unique(Y)
array([0, 1])
pd.Series(Y).value_counts()
1 472
0 395
dtype: int64
X.shape, X_updated.shape
((867, 200, 200), (867, 40000))
Visualize data
plt.imshow(X[0], cmap='gray')
<matplotlib.image.AxesImage at 0x20c9f5bb610>
2324
Prepare data
X_updated = X.reshape(len(X), -1)
X_updated.shape
(867, 40000)
Split Data
xtrain, xtest, ytrain, ytest = train_test_split(X_updated, Y, random_state=10,
test_size=.20)
xtrain.shape, xtest.shape
((693, 40000), (174, 40000))
Feature Scaling
print(xtrain.max(), xtrain.min())
print(xtest.max(), xtest.min())
xtrain = xtrain/255
xtest = xtest/255
print(xtrain.max(), xtrain.min())
print(xtest.max(), xtest.min())
255 0
```

```
255 0
1.0 0.0
1.0 0.0
Feature Selection: PCAfrom sklearn.decomposition import PCA
print(xtrain.shape, xtest.shape)
pca = PCA(.98)
# pca_train = pca.fit_transform(xtrain)
# pca_test = pca.transform(xtest)
pca_train = xtrain
pca_test = xtest
(693, 40000) (174, 40000)
# print(pca_train.shape, pca_test.shape)
# print(pca.n_components_)
# print(pca.n_features_)
Train Model
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
import warnings
warnings.filterwarnings('ignore')
lg = LogisticRegression(C=0.1)
lg.fit(xtrain, ytrain)
LogisticRegression(C=0.1)
sv = SVC()
25sv.fit(xtrain, ytrain)
SVC()
Evaluation
print("Training Score:", lg.score(xtrain, ytrain))
print("Testing Score:", lg.score(xtest, ytest))
Training Score: 1.0
Testing Score: 0.9482758620689655
print("Training Score:", sv.score(xtrain, ytrain))
print("Testing Score:", sv.score(xtest, ytest))
Training Score: 0.98989898989899
Testing Score: 0.9540229885057471
Prediction
pred = sv.predict(xtest)
misclassified=np.where(ytest!=pred)
misclassified
(array([ 0, 27, 39, 40, 98, 103, 126, 147], dtype=int64),)
print("Total Misclassified Samples: ",len(misclassified[0]))
print(pred[36],ytest[36])
Total Misclassified Samples: 8
1 1
TEST MODEL
dec = {0:'No Tumour', 1:'Positive Tumour'}
```

```
2627
plt.figure(figsize=(12,8))
p = os.listdir('brain_tumour/Testing/')
c=1
for i in os.listdir('brain_tumour/Testing/no_tumour/')[:9]:
plt.subplot(3,3,c)
img = cv2.imread('brain_tumour/Testing/no_tumour/'+i,0)
img1 = cv2.resize(img, (200,200))
img1 = img1.reshape(1,-1)/255
p = sv.predict(img1)
plt.title(dec[p[0]])
plt.imshow(img, cmap='gray')
plt.axis('off')
c+=1
```

SCREENSHOTS


No Tumor
No Tumor
No Tumor
No Tumor
No Tumor
No Tumor
No Tumor
No Tumor
No Tumor


C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.
D:\>jupyter notebook
[I 2024-04-08 13:43:30.990 LabApp] JupyterLab extension loaded from C:\Users\jashu\anaconda3\lib\site-packages\jupyterla
b
[I 2024-04-08 13:43:30.990 LabApp] JupyterLab application directory is C:\Users\jashu\anaconda3\share\jupyter\lab
[I 13:43:30.990 NotebookApp] Serving notebooks from local directory: D:\
[I 13:43:30.990 NotebookApp] Jupyter Notebook 6.3.0 is running at:
[I 13:43:30.990 NotebookApp] http://localhost:8888/?token=a0f96e6ef8ead188c215bcc6f435739def6291a87c57f541
[I 13:43:30.990 NotebookApp] or http://127.0.0.1:8888/?token=a0f96e6ef8ead188c215bcc6f435739def6291a87c57f541
[I 13:43:30.990 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 13:43:31.021 NotebookApp]
To access the notebook, open this file in a browser:
file:///C:/Users/jashu/AppData/Roaming/jupyter/runtime/nbserver-9764-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=a0f96e6ef8ead188c215bcc6f435739def6291a87c57f541
or http://127.0.0.1:8888/?token=a0f96e6ef8ead188c215bcc6f435739def6291a87c57f541