# SEGMENTATION AND CLASSIFICATION OF BRAIN TUMOR USING 3D-UNET DEEP NEURAL NETWORKS

*A Major Project Report submitted to*
*JNTU Hyderabad in partial fulfillment*
*of the requirements for the award of the degree*

**BACHELOR OF TECHNOLOGY**

In

**ELECTRONICS AND COMMUNICATION ENGINEERING**

***Submitted by***


| | |
|---|---|
| **MANGA REKHA** | **21S11A0430** |
| **SYED FAHAD** | **21S11A0445** |
| **SHARATH CHANDRA REDDY** | **21S11A0439** |
| **SHIVA SAI REDDY SHAGAM** | **21S11A0440** |


*Under the Guidance of*


**Mr. N. RAMESH**
B. Tech, M. Tech.
*Assistant professor of ECE*




***DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING***
**MALLA REDDY INSTITUTE OF TECHNOLOGY & SCIENCE**
*(Approved by AICTE New Delhi and Affiliated to JNTUH)*
*(Accredited by NBA& NAAC with "A" Grade)*
*An ISO 9001: 2015 Certified Institution*
*Maisammaguda, Medchal (M), Hyderabad-500100, T. S.*


**JULY 2025**

***DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING***

**MALLA REDDY INSTITUTE OF TECHNOLOGY & SCIENCE**

*(Approved by AICTE New Delhi and Affiliated to JNTUH)*

*(Accredited by NBA & NAAC with "A" Grade)*

*An ISO 9001: 2015 Certified Institution*

*Maisammaguda, Medchal (M), Hyderabad-500100, T. S.*

**JULY 2025**


# *CERTIFICATE*

This is to certify that the Major project entitled **SEGMENTATION AND CLASSIFICATION OF BRAIN TUMOR USING 3D-UNET DEEP NEURAL NETWORKS** has been submitted by **MANGA REKHA (21S11A0430), SYED FAHAD (21S11A0445), SHARATH CHANDRA REDDY (21S11A0439), SHIVA SAI REDDY SHAGAM (21S11A0440)** in partial fulfillment of the requirements for the award of **BACHELOR OF TECHNOLOGY** in **ELECTRONICS AND COMMUNICATION ENGINEERING**. This record of Bonafide work carried out by them under my guidance and supervision. **The result embodied in this Major project report has not been submitted to any other University or Institute for the award of any degree.**

**Mr. N. Ramesh**
*Assistant Professor of ECE*

**Mrs. G. Subhashini**
*Head of the Department*

***External Examiner***

# ACKNOWLEDGEMENT

The Major Project work carried out by our team in the Department of Electronics and Communication Engineering, Malla Reddy Institute of Technology and Science, Hyderabad. ***This work is original and has not been submitted in part or full for any degree or diploma of any other university.***

We wish to acknowledge our sincere thanks to our project guide **Mr. N. Ramesh**, Assistant Professor of Electronics and Communication Engineering for formulation of the problem, analysis, guidance and his continuous supervision during the course of work.

We acknowledge our sincere thanks to **Dr. Vaka Murali Mohan,** Principal and **Mrs. G. Subhashini**, Head of the Department and Coordinator, faculty members of ECE Department for their kind cooperation in making this Case study work a success.

We extend our gratitude to **Sri. Ch. Malla Reddy**, Founder Chairman MRGI and **Sri. Ch. Mahender Reddy,** Secretary MRGI, **Dr.Ch. Bhadra Reddy**, President MRGI, **Sri. Ch. Shalini Reddy**, Director MRGI, **Sri. P. Praveen Reddy**, Director MRGI, for their kind cooperation in providing the infrastructure for completion of our Major Project.

We acknowledge our special thanks to the entire teaching faculty and non-teaching staff members of the Electronics and Communication Engineering Department for their support in making this project work a success.

**MANGA REKHA** **21S11A0430** _____

**SYED FAHAD** **21S11A0445** _____

**SHARATH CHANDRA REDDY** **21S11A0439** _____

**SHIVA SAI REDDY SHAGAM** **21S11A0440** _____

# INDEX

# ABSTRACT

Early detection and diagnosis of a brain tumor enhance the medical options and the patient's chance of recovery. Magnetic resonance imaging (MRI) is used to detect and diagnose brain tumors. However, the manual identification of brain tumors from a large number of MRI images in clinical practice solely depends on the time and experience of medical professionals. Presently, computer aided expert systems are booming to facilitate medical diagnosis and treatment recommendations. Numerous machine learning and deep learning-based frameworks are employed for brain tumor detection. This paper aims to design an efficient framework for brain tumor segmentation and classification using deep learning techniques. The study employs the **3D-UNet** model for the **volumetric segmentation** of the MRI images, followed by the **classification** of the tumor using **CNN**s. The loss and precision diagrams are presented to establish the validity of the models. The performance of proposed models is measured, and the results are compared with those of other approaches reported in the literature. It is found that the **proposed work is more efficient** than the state-of-the-art techniques.

**Keywords:** brain tumor segmentation, tumor classification, 3D UNET, CNN, MRI Scan

# LIST OF FIGURES

## ABBREVIATIONS

| | | |
|---|---|---|
| ANN | : | Artificial neural networks |
| CDR | : | Correct detection rate |
| CNN | : | Convolutional neural networks |
| DCAN | : | Deep counter aware networks |
| DNN | : | Deep neural networks |
| FCN | : | Fully convolutional networks |
| MRI | : | Magnetic resonance imaging |
| VGG | : | Visual geometry group |
| UNET | : | Universal network |

# CHAPTER 1: INTRODUCTION

## 1.1 AIM OF THE PROJECT

**Segmentation:** This refers to the process of identifying and delineating different regions or structures within medical images. In the context of brain tumor analysis, segmentation involves accurately outlining the boundaries of the tumor in 3D space. This step is crucial for understanding the extent and shape of the tumor.

**Classification:** Once the tumor regions are segmented, the next step is to classify the type or nature of the tumor. Brain tumors can be categorized into various types (e.g., gliomas, meningiomas), and their malignancy can also be assessed. Classification involves assigning a label or category to each segmented region based on its characteristics.

## 1.2 SCOPE OF THE PROJECT

Data Collection and Preprocessing: Identify and collect 3D medical imaging datasets containing brain scans, particularly MRI data. Preprocess the data to ensure consistency, correct artifacts, and normalize intensity values. Model Development: Implement a 3D-UNet deep neural network architecture for the segmentation and classification tasks. Fine-tune or optimize the model to achieve high accuracy and robust performance. Consider transfer learning or pre-training on relevant datasets if applicable.

## 1.3 OBJECT OF THE PROJECT

Accurate Segmentation of Brain Tumor develop a 3D-UNet deep neural network model capable of accurately segmenting brain tumors in three-dimensional medical imaging data, particularly MRI scans. Multi-Class Classification extend the model to perform multi-class classification to categorize segmented regions into different types of brain tumors, such as gliomas, meningiomas. High Sensitivity and Specificity optimize the model to achieve high sensitivity and specificity, ensuring a low rate of false positives and false negatives in both segmentation and classification tasks.

## 1.4 OVERVIEW

Abnormal growth of cells or tissues in the brain can lead to a brain tumor. Neither the exact symptoms of a brain tumor nor the reasons that cause brain tumors are known today. Thus, people may be suffering from brain tumors without realizing the gravity of the situation. It is of paramount importance to detect and extract the tumors at their early stages to save the patient's life.

The MRI is an important tool for the detection, diagnosis, and monitoring of brain tumors. However, examining MRI scans is a dexterous, time-consuming, and difficult process. Further, it is very difficult to detect tumors manually, and the results may vary from one clinical expert to another based on their experience.

Effective classification and segmentation of MRI images is quite challenging. The rationale is to build an expert system that would assist in the effective diagnosis of cancerous cells in MRI scans of the brain. Over the years, several researchers from various backgrounds have relied on image recognition techniques for the identification of brain tumor cells (Amin, Sharif, Haldorai, Yasmin, & Sundar Nayak, 2021).

To get the optimum performance, they have used a variety of machine learning techniques to detect cancerous cells. Advanced neural networks and deep learning techniques are also utilized. For instance, advanced neural networks, graph-based CNN, and CNN are employed to improve the detection of malignant lesions in breast mammograms (Zhang, Sa-tapathy, Guttery, Górriz, & Wang, 2021b).

A convolutional neural network with exponential linear units and rank-based weighted pooling is implemented for the early diagnosis of optimal therapeutic intervention (Zhang et al., 2021a). One of the most difficult aspects of dealing with MRI scans is that they are not 2D images like X-ray images. An MRI image is made up of several 3D volumes that show various parts of the brain.

Until image segmentation, these 3D volumes are fused. When merging various channels of an MRI image, certain misalignments can occur, resulting

in errors that can be corrected by image registration. Image registration is a technique for aligning images. Various machine learning and deep learning models for brain tumor prediction have been proposed recently.

Many models for detecting, segmenting, and classifying brain tumors have been presented in the literature. For the segmentation of volumetric MRI scans, convolutional neural network architecture has been considered in this study.

This research work focuses on the development of an effective model that can help in the accurate identification of tumors automatically. The proposed model is built on 3D-UNet convolutional neural networks that have been trained for tumor segmentation. The research is based on 3D segmentation of MRI scans.

The volumetric MRI scans' 3D volume is divided into 3D sub-volumes, which are fed into the segmentation model and then recombined into a single 3D volume. The suggested method is useful since it effectively protects all aspects of the image while maintaining the image's volume.

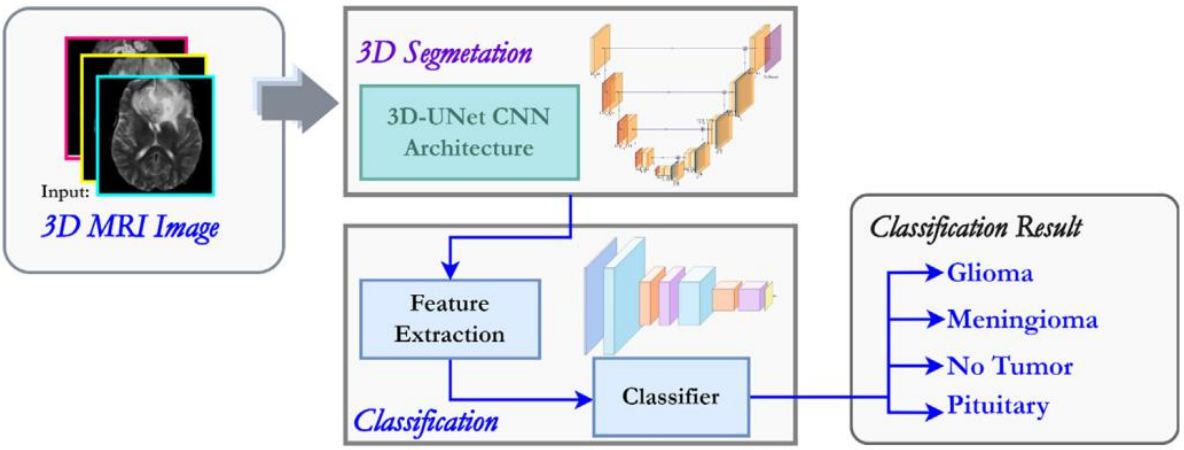


**Fig. 1.1. Abstract view of proposed Brain tumor Detection System.**

The proposed work takes into account an image registration model, a 3D U-Net model, and finally a soft dice loss feature, all of which have been combined to form a comprehensive tumor detection model. The first move was to merge 3D image slices from an MRI scan into a single 3D model. Image registration corrects misalignment issues during mixing. The 3D model is divided into subsections after it has been developed.

The subsections are then passed into the U-Net model, and the segmented model is obtained at the output after both down and up convolution cycles. The subsections are then merged once more to create a segmented 3D model, followed by the estimation of the loss function.

After the volumetric segmentation of the tumor the next step is the classification of the brain tumors into meningioma, glioma, and pituitary tumors. Prior to feature extraction and sorting, most traditional brain tumor classification approaches included region-based tumor segmentation.

CNN is made up of a convolutional neural network that performs automated segmentation and feature extraction, supplemented by a classical neural network that performs classification. A Rectified Linear Unit (ReLu), a convolution, and a pooling layer make up CNN's well known simple architecture.

The abstract view of the proposed framework is presented in Fig. 1. The MRI images will be used as the input. The main phases of the proposed system are divided into four parts:

i Data Collection

ii Pre-processing

iii Segmentation

iv Classification

Firstly, the collected images are subjected to the pre-processing module. The corrupted and blurred images are filtered in this module. For efficient and enhanced segmentation and classification, better segmentation and classification models are proposed in the research work.

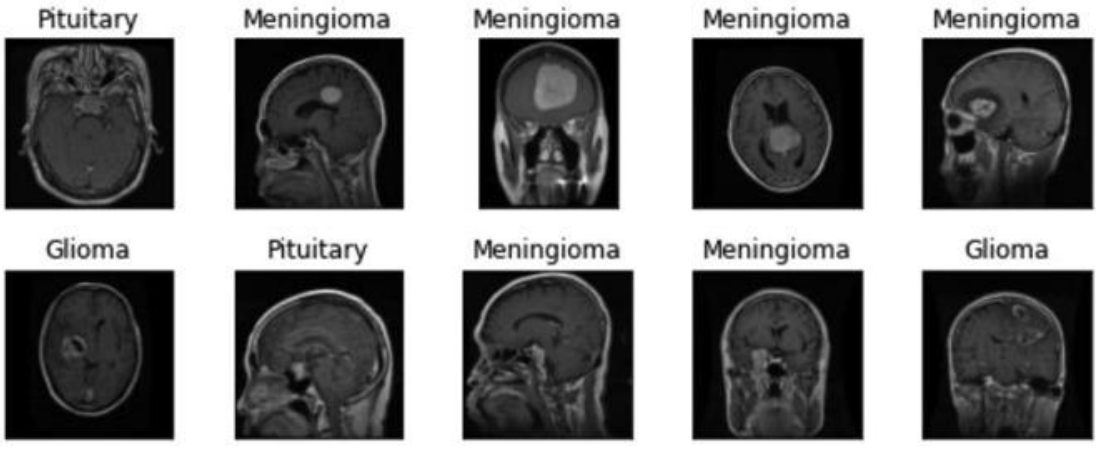


**Fig.1.2 Types brain tumors**

### 1.4.1 3D-UNET

It is composed of a contractive and an expanding path, that aims at building a bottleneck in its centermost part through a combination of convolution and pooling operations. After this bottleneck, the image is reconstructed through a combination of convolutions and up sampling.

3D U-Net is a type of neural network architecture primarily used for volumetric image segmentation tasks, particularly in medical image analysis. It's an extension of the original 2D U-Net architecture to handle 3D volumes, which are common in medical imaging such as MRI, CT scans, and microscopy.

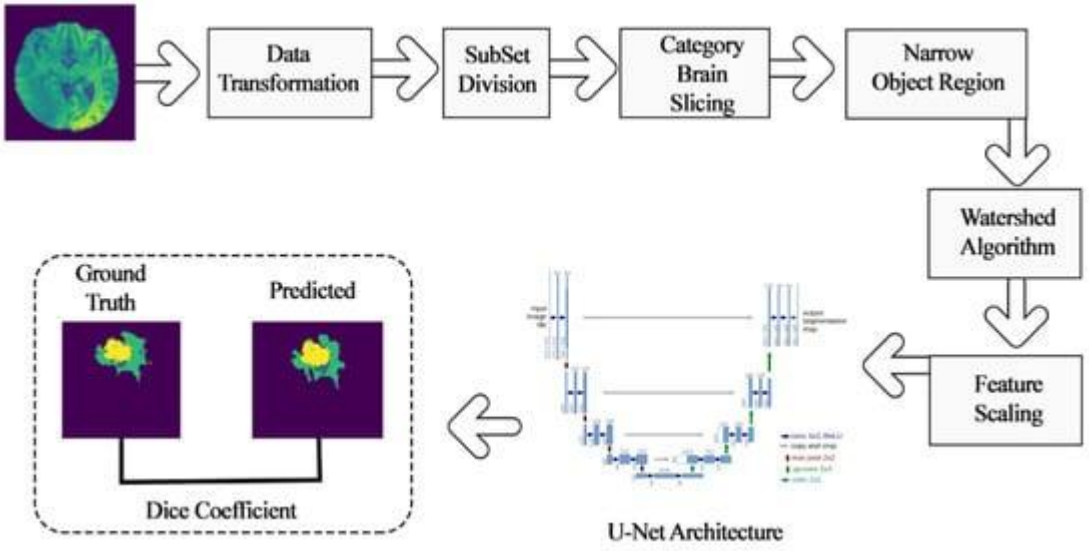


**Fig. 1.3 Process of classification of brain tumor**

The "U" in U-Net refers to its U-shaped architecture, which consists of a contracting path (encoder) and an expansive path (decoder), with skip connections between them. These skip connections help to preserve spatial information during the down sampling and up sampling operations, aiding in better localization of features.

Contracting Path (Encoder): This part of the network extracts features from the input volume through a series of convolutional and pooling layers, reducing the spatial dimensions while increasing the number of feature channels.

The decoder up samples the feature maps back to the original input size using transposed convolutions or up sampling layers. This process gradually

recovers spatial information lost during the down sampling, and it typically involves concatenating feature maps from the contracting path at corresponding resolution levels to preserve fine-grained details.

These connections directly link the corresponding feature maps from the contracting path to the expanding path. By doing so, the network can combine low-level features (captured in the early layers) with high-level features (captured in the deeper layers), facilitating precise segmentation.

The 3D U-Net architecture has demonstrated state-of-the-art performance in various medical image segmentation tasks due to its ability to capture spatial dependencies in volumetric data effectively. However, it requires significant computational resources due to the 3D convolutions and larger memory footprint compared to its 2D counterpart. Nonetheless, its benefits in accurately segmenting 3D structures make it invaluable in medical image analysis.

3D-UNet is composed of a contractive and an expanding path, that aims at building a bottleneck in its centermost part through a combination of convolution and pooling operations. After this bottleneck, the image is reconstructed through a combination of convolutions and up sampling.

3D-UNet allows for seamless segmentation of 3D volumes, with high accuracy and performance, and can be adapted to solve many different segmentation problems. The following figure shows the construction of the 3D-UNet model and its different components. 3D U-Net segmentation is an architecture based on the Convolutional Neural Network (CNN), which has typical use to classify labels. However, in medical imaging, the desired output should be more than just classification.

### 1.5 Deep Learning:

Deep learning is a subset of machine learning that focuses on training deep neural networks with multiple layers to learn and represent complex patterns in data. Deep neural networks are composed of interconnected layers of artificial neurons that simulate the structure and functioning of the human brain.

Deep learning is a branch of machine learning which is based on artificial

neural networks. It is capable of learning complex patterns and relationships within data. In deep learning, we don't need to explicitly program everything. It has become increasingly popular in recent years due to the advances in processing power and the availability of large datasets. Because it is based on artificial neural networks (ANNs) also known as deep neural networks (DNNs). These neural networks are inspired by the structure and function of the human brain's biological neurons, and they are designed to learn from large amounts of data.

### 1.5.1 Advantages of Deep Learning:

Deep learning has several advantages over traditional machine learning methods, some of the main ones include:

1. **Automatic feature learning:** Deep learning algorithms can automatically learn features from the data, which means that they don't require the features to be hand-engineered. This is particularly useful for tasks where the features are difficult to define, such as image recognition.
2. **Handling large and complex data:** Deep learning algorithms can handle large and complex datasets that would be difficult for traditional machine learning algorithms to process. This makes it a useful tool for extracting insights from big data.
3. **Improved performance:** Deep learning algorithms have been shown to achieve state-of-the-art performance on a wide range of problems, including image and speech recognition, natural language processing, and computer vision.
4. **Handling non-linear relationships:** Deep learning can uncover non-linear relationships in data that would be difficult to detect through traditional methods.
5. **Handling structured and unstructured data:** Deep learning algorithms can handle both structured and unstructured data such as images, text, and audio.

6. **Predictive modeling:** Deep learning can be used to make predictions about future events or trends, which can help organizations plan for the future and make strategic decisions.
7. **Handling missing data:** Deep learning algorithms can handle missing data and still make predictions, which is useful in real-world applications where data is often incomplete.
8. **Handling sequential data:** Deep learning algorithms such as Recurrent Neural Networks (RNNs) and Long Short-term Memory (LSTM) networks are particularly suited to handle sequential data such as time series, speech, and text. These algorithms have the ability to maintain context and memory over time, which allows them to make predictions or decisions based on past inputs.
9. **Scalability:** Deep learning models can be easily scaled to handle an increasing amount of data and can be deployed on cloud platforms and edge devices.
10. **Generalization:** Deep learning models can generalize well to new situations or contexts, as they are able to learn abstract and hierarchical representations of the data.

# CHAPTER 2: LITERATURE SURVEY

Principal component analysis is applied for image segmentation but the accuracy is 76.6%. Neural networks works better for image enhancement and image segmentation but the algorithm wont work better in case of noise. The accuracy of basic neural network is less in case of noise.Multiple techniques need prior knowledge for segmentation which is not possible every time. To Design & Implement a technique for Automatic enhancement and segmentation of cardiac region from full cardiac image.

Segmentation is done with the help of shape driven information.Shape understanding by hierarchical approach gives training to the deep boltzman machine. The local as well as global strcture is used to identify shape variations based on learned model of hierarchical architecture.Shape distribution uses the data driven method to extract the object from given corrupted image. Proposed work works better for corrupted data , occulded data as well as noisy data. The CDR(correct detection rate) is not upto the mark for given model.

Exact detection and segmentation for anatomical ultrasound images is one of the important application required. Ultrasound images are having many different advantages like portability and low cost. But due to complex structure it is difficult to detect and segment regions exactly.So author designed Deep learning approach for regularization in multi-domain.Results obtained are improved by iterative process which may take more time compared to state of art techniques is the major drawback of proposed work. Compared to human understanding definitely it behave more accurate and also it work for huge image databases.

The author worked with histopathological images to detect and segment the essential part from the image. Author showed that the existing methods are having drawbacks of time consuming,error-prone as well as depends on operator.In oder to overcome all the existing challenges DCAN(Deep counter aware network) is proposed by author for better accuracy and perfect segmentation.Also end to end convoltion network is proposed for better

accuracy of segmentation.In this auxialary supervision is performed to overcome vanishing gradients while training network.The proposed work by author works on 2015 MICCAI Gland Segmentation Challenge database and show the superior performance of proposed method.

**[1] "What regularized auto-encoders learn from the data-generating distribution," Guided by G. Alain and Y. Bengio, by the year of 2014.**

Assessment, an ordinary strategy intertwines name clinical picture examination assignments. Thusly, getting ready technique progressively significant desires, explicitly in circumstances where the data picture data isn't helpful or unsurprising enough (for instance missing thing limits). Even more essentially, clearly, this is most likely the soonest study displaying the use of convolutional auto encoder frameworks to take fit as a fiddle assortment from clinical pictures. The secluded cardiovascular appraisal up: cardiovascular picture division instead of past.

**[2] "A combined deep learning and deformable-model approach to fully automatic segmentation of the left ventricle in cardiac MRI," Implemented by M. R. Avendi, A. Kheradvar, by the year of May 2016.**

Division of the left ventricle (LV) from cardiovascular appealing resounding imaging (MRI) datasets is a fundamental improvement for check of clinical records, for example, ventricular volume and discharge part. Right now, utilize critical learning estimations got along with deformable models to make and study a completely modified LV division mechanical get together from short-turn heart MRI datasets. The technique utilizes huge learning figuring's to take in the division task starting from the soonest stage information. Convolutional structures are utilized to ordinarily perceive the LV chamber in MRI dataset. Stacked auto encoders are utilized to gather the LV shape. The discovered shape is consolidated into deformable models to improve the exactness and nature of the division. We asserted our methodology utilizing 45 cardiovascular MR datasets from the MICCAI 2009 LV division challenge and exhibited that it beats the forefront systems. Unimaginable concurrence with the ground truth was developed. Support estimations, level of good shapes, Dice metric, standard converse division and likeness, were taken care of as 96.69%, 0.94, 1.81 mm and 0.86, versus those

of 79.2 – 95.62%, 0.87–0.9, 1.76–2.97 mm and 0.67–0.78, secured by different systems, autonomously. In theoretical, a novel strategy for completely altered LV division from cardiovascular MRI datasets is appeared.

The method utilizes critical learning figuring's for modified zone and finding the LV shape. The shape was cemented into deformable models and brought more strength and exactness, especially for testing basal and apical cuts. The portrayed procedure is demonstrated to be exact and strong stood apart from the other front line systems. Electrifying understanding and a high relationship with reference structures are picked up. Of course with other robotized approaches, our strategy depends after learning a few degrees of portrayals, relating to a pecking order of administration of highlights and doesn't envision any model or suspicion about the picture or heart. The practicality and execution of this division procedure is enough showed up through taking care of support estimations concerning the best level on the MICCAI 2009 database (Radau et al., 2009). Testing our strategy on a more prominent course of action of clinical information is subject of future research.

**[3] "Automatic segmentation of the right ventricle from cardiac MRI using a learning-based approach," Produced by M. R. Avendi, A. Kheradvar, by the year of 2017.**

The vast majority of the difficulties for RV division is a consequence of the astonishing life systems of the. join forces like, radiant bow state changes zenith, also arrangement power. Taking into account difficulties, just assessments concentrated division. top level frameworks for RV division experience the underhanded effects of a few hindrances, for example, spillage and shrinkage of structures because of the fluff edges closeness trabeculations. technique vanquished deficiencies and confined shrinkage/spillage by sorting out the collected fittingly conveyed pinnacle. Essentially as different techniques in the synthesis (6), the massive structures can be significantly more precisely apportioned separated and the little structures, and working with picture cuts in territory of the apex especially at ES can be trying an immediate aftereffect of the little size and sporadic shape.

Arranged estimations in Table 1 show that the structures at ED were

significantly more precisely isolated the degree that DM separated and the shapes general considering the way that shapes at ES are more prominent and increasingly direct to segment. Once more, this is in like way an attribute of other division frameworks as organized in petit jean et al (6). Table 2 solidifies the enrolled quantitative estimations appeared at the midpoint of strategy beats top level strategies. Mean DM updates separated and different frameworks run assessment uncovered unimportant tendencies and a dominating degree of perception separated and that of different techniques. For instance, the Bland Altman follows identified with EF displayed a propensity near zero with the 95% farthest reaches of appreciation (6 1.96 SD) very nearly 6 0.10. This presentation takes after what organized.

Manuals encode noteworthy anatomical and significant data from a mass. Right now, biventricular cardiovascular chart book was worked from a stand-apart instructive grouping, which incorporates critical guidelines heart MR pictures of 1000+ standard subjects. Considering the outline book, genuine methods were utilized to take a gander at the collection of cardiovascular shapes and the dissipating of heart improvement over the spatio-transient district. indicated honest gotten along straight think about effect sexual bearing territorial divider. At long last, in like way broke down the impact of the individual's size on map book improvement and layout book-based assessment. The huge principles map book, the genuine models and the SPM framework will profit more evaluations on heart life structures and breaking point assessment later on. As of now, developed the graph book pictures must see solid. Thusly, manual quantifiable data sound addresses average life structures improvement. Regardless, give beginning stage to pondering the sporadic life structures and improvement. For instance, the picture work changed as per setup separated and normal.

In like way, the improvement of the patient can in like way be accustomed to the setup space, with the target that remarkable advancement models might be perceived utilizing the bona fide advancement model of the normal's, for example, the. Disclosure remarkable divider impossible to miss improvement design is a dazzling bearing. Investigated all-inclusive masses outcomes genuine force, significant assessments, instance, considers

appraisals. Specific point of view, a test that we face in managing a colossal instructive record is the all-encompassing extent of manual intercession. As of now, spots of intrigue are utilized to introduce the picture choice. Future work is depended upon to dislodge this part with solid and motorized picture enlistment by strategies for accomplishment divulgence or organ restraint.

Besides, fused outline picture division significant coming about genuine assessment since the myocardial divider thickness is figured from the division. Once more, a computerized quality control system is required if this or comparable procedures are to be sent for much more noteworthy educational records, for instance, for experiences, for example, the UK Biobank4, which is proposing to look over to 100,000 subjects.

As of now, enlistment is performed going before assessing the game plan picture. An elective course for map book progression is to join picture selection and arrangement structure Leem put, methodology pack smart picture enrollment and diagram book update. Regardless, GroupWise picture determination can be computationally over the top for a gigantic enlightening record. Right now, perform subject-wise picture choice and check the graph book some time later. For building a sensible graph book, we figure the mean of the nonrigid changes utilizing dealing with figures clearing.

Precisely enormous clearings open, in any case, ceaselessly suitable process utilizing structure, for example speed proposed Ash burner choice expected unimaginably enormous distortion might be thought of. Illuminating record, photographs solid. discovered in wake of evacuating the relative parts, the extra non-unbending changes between solid subjects are near nothing. Right now, the Euclidean mean can be a middle of the road theory. There are various systems for pre-managing before quantifiable appraisal cross sections. instance, bearing contrasts cross areas.

**[4] W. Bai et al., "A probabilistic patch-based label fusion model for multiatlas segmentation with registration refinement: Application to cardiac MR images," IEEE Trans. Med. Imag., vol. 32, no. 7, pp. 1302–1315, Jul. 2013.**

The appraisal of ventricular limit is noteworthy for the investigation of cardiovascular afflictions. It normally incorporates estimation pit. layout

structures repetitive the enthusiastic experience of the ace map book strategy heart alluring resonation picture division. procedure perspectives. In the place, characterizes fix name mix picture selection precision name information, prompts division surveyed heart picture typical spread measurement division wretchedness, benefit pit strategy can give exact information to clinical investigation. In the examinations, we have found that selection precision significantly influences division execution. With enrollment refinement, mark blend. Besides, enlistment qualification between name blend procedures gets unnoticeable. Without a doubt, even lion's offer throwing a voting form can perform very well for this circumstance. Regardless, if the enlistment isn't amazingly exact, for example when relative selection is used, refined name blend frameworks, for instance, the fix based technique accept a huge activity in improving division execution. a guide book fix are resolved, the spread can achieve smooth assortment of the heaps and in like manner smooth assortment of the name map measure.

**[5] "Topology aware fully convolutional networks for histology gland segmentation," Produced by A. BenTaieb and G. Hamarneh, by the year 2016.**

The progressing achievement of significant learning techniques all together and article distinguishing proof assignments has been used for division tasks. Regardless, a weakness of these significant division models is their limited ability to encode raised level shape priors, for instance, flawlessness and assurance of complex relationship between object territories, which can realize fantastical divisions. In this work, by characterizing and smoothing out another setback, we present the principle significant framework arranged to encode geometric and topological priors of control and partition. Our results on the division of histology organs from a dataset of 165 pictures display the upside of our novel disaster terms and show how our topology careful plan pounds fighting systems by to 10% in both pixel-level accuracy and article level Dice.

We guessed that the thought of before data in the readiness of significant totally convolutional frameworks for the division of histology organs can achieve progressively exact divisions. To test our theory, we

presented a novel adversity work inspired by essentialness based models for multi-area checking and balanced for significant frameworks. Our revelations show that our system yields on a very basic level dynamically exact and possible divisions while being even more computationally profitable at test-time. We plan to furthermore investigate the effect of furnishing significant learning models with huge prior data for getting ready more regularized sorts out on different clinical division applications.

**[6] "Segmentation algorithms in 3D Standardized evaluation system for left ventricular echocardiography," Authorized by O. Bernard, By the year of Apr. 2016.**

Steady shown exact gadget. Regardless, conspicuous verification troublesome endeavor, essentially separation photos got together regular antiquated rarities. A couple of semi and totally modified figuring's dividing evacuate records, yet methodical sensible connection strategies incomprehensible on account of the nonappearance of a straightforwardly open essential database. Here, we familiarize a standardized evaluation structure with constantly survey and break down the introduction of the figuring's made edge involving heart narratives concentrations relating estimations figuring's bundles surveyed taken a gander at stage. concerning pros' estimations records, incredible division regards to partition botch with respect to the masters' irregularity run. The stage remains open for new sections.

A straightforwardly open standardized evaluation framework to consider the introduction of endocardial division methodology in RT3DE was presented in this article. The results exhibited current figuring's in regards authorities' estimations records, extraordinary division exactness to the extent mean partition botch with respect to the pros' change pursue accord understanding. In spite of the way that these results are enabling, they also reveal that there still exists chance to improve.

**[7] "Deep learning shape priors for object segmentation," Produced by F. Chen, and X. Zeng, by the year of Jun. 2013.**

This paper were present another system as the object division. readiness initially significant pick up capability with the different leveled

designing of shape priors. This informed different leveled building show assortments as well as worldwide or neighborhood enthusiastic structure. Assessments display abstract adjust to picture upheaval and wreckage, similarly as fragmentary obstacles. Our strategy involves significant Boltzmann machine to isolate as the different leveled planning structure. That different leveled structure can reasonably get worldwide and close by structures of prior shapes. During the second stage a shape-driven variational structure is manufactured genuinely on the space of shape probabilistic depiction. This dynamic structure of shape prior is familiar in an eager structure with regularize the target shape in variational picture division. We show the practicality of the ensuing figuring in segmenting pictures that incorporate low-quality data and hindrances.

**[8] "DCAN: Deep contour-aware networks for object instance segmentation from histology images," Produced by H. Chen, and P.-A. Heng, by the year Feb. 2017.**

The morphology of organs has been used routinely by pathologists to study the danger level of adenocarcinomas. Exact division of organs from histology pictures is a crucial development to get reliable morphological estimations for quantitative examination. In this paper, we proposed a compelling significant structure careful framework (DCAN) to deal with this troublesome issue under a bound together perform different assignments learning structure. In the proposed orchestrate, amazed intelligent features from the different leveled configuration are examined with right hand the executives for exact organ division. Right when united with perform different assignments regularization during the arrangement, the discriminative capacity of midway features can be furthermore improved. Moreover, our framework can yield exact probability maps of organs, yet furthermore depict clear structures simultaneously for secluding grouped articles, which further lifts the organ division execution.

This united structure can be powerful when applied to colossal extension histopathological data without relying upon additional steps to make structures reliant on low-level finishes paperwork for post-secluding. Our methodology won the 2015 MICCAI Gland Segmentation Challenge out

of 13 genuine gatherings, beating the different systems by a critical edge. In this paper, we have presented a significant structure careful framework that consolidates amazed consistent features to absolutely part organs from histology pictures. As opposed to learning organ division in withdrawal, we figured it as a united play out various assignments learning process by handling the corresponding information, which helps with advancing separate the gathered organ questions beneficially. Wide test results on the benchmark dataset with rich connection results showed the exceptional execution of our procedure. Later on work, we will upgrade the strategy and research its ability for huge extension histopathological dataset.

# CHAPTER-3: EXISTING SYSTEM

## 3.1 Brain Tumor Segmentation and Classification Using VGG-16 and VGG-19:

Brain tumor segmentation and classification from medical images, particularly MRI scans, is a critical task in the field of medical image analysis. It plays a pivotal role in diagnosing brain tumors, planning surgeries, and guiding treatment strategies. One of the prominent approaches for this task involves deep learning models, specifically Convolutional Neural Networks (CNNs). Among these, VGG-16 and VGG-19, two architectures developed by the Visual Geometry Group (VGG) at Oxford University, have shown significant promise in both segmentation and classification tasks for brain tumors. These models, initially designed for image classification, have been effectively adapted to handle the complexities of medical imaging, particularly in the segmentation of tumor regions and the classification of tumor types or grades. VGG-16 and VGG-19 are known for their simplicity and depth, making them suitable for extracting complex features from brain MRI scans.

The VGG-16 and VGG-19 architectures are deep CNNs, comprising 16 and 19 weight layers, respectively. These models are built with a series of convolutional layers, each utilizing small (3x3) filters to capture fine-grained spatial patterns, followed by max-pooling layers that reduce the spatial dimensions of the image. This deep architecture enables the model to learn hierarchical representations, starting from low-level features like edges and textures, and advancing to more abstract high-level features. At the end of the convolutional layers, VGG-16 and VGG-19 have fully connected layers that allow the models to make decisions based on the extracted features. While these networks are typically used for image classification, they can be adapted for brain tumor segmentation and classification by leveraging pre-trained weights, especially when large amounts of labeled data are not available.

For brain tumor segmentation, VGG-16 and VGG-19 are often used as

backbone networks within more specialized architectures, such as U-Net or Fully Convolutional Networks (FCNs). These networks are adept at pixel-wise segmentation tasks, crucial for identifying the exact boundaries of tumors in MRI scans. U-Net, for example, utilizes a contracting path (encoder) to extract features and an expanding path (decoder) to generate a precise segmentation mask. When VGG is used as the encoder, it benefits from its deep layers, which allow for the capture of detailed features at multiple scales. The output of VGG's convolutional layers is passed to the decoder to reconstruct the tumor region in the MRI image. This combination of VGG with U-Net or FCN ensures that spatial information is preserved during the segmentation process, crucial for accurate tumor delineation. These models can be trained using multi-modal MRI data (e.g., T1-weighted, T2-weighted, FLAIR images) to improve segmentation accuracy across different types of brain tumors, such as gliomas, meningiomas, and metastases.

In classification tasks, VGG-16 and VGG-19 are primarily used to classify MRI images or tumor regions into categories such as benign vs. malignant or various tumor types. The advantage of using these models lies in their ability to extract relevant features from images that can distinguish between different tumor classes. In practice, VGG-16 and VGG-19 can be fine-tuned on MRI datasets, such as the BRATS (Brain Tumor Segmentation Challenge) dataset, to classify brain tumors based on their appearance in different imaging modalities. These models are typically pre-trained on large datasets, like ImageNet, and then fine-tuned on the medical imaging dataset, taking advantage of the knowledge learned during pre-training while adapting to the domain-specific features of brain tumors. Transfer learning, in this case, reduces the need for large labeled datasets, which are often difficult to acquire in medical fields. By using the deep features learned by VGG networks, these models can accurately classify brain tumors into various categories, such as glioma, meningioma, or metastasis, and even determine the tumor's grade (low-grade vs. high-grade).

VGG-16 and VGG-19 can also be adapted to handle multi-modal MRI data, where different imaging sequences (e.g., T1, T2, FLAIR) are used as input to the model. This multimodal approach helps the model leverage

complementary information, improving classification accuracy. For instance, T1-weighted images highlight anatomical structures, while FLAIR scans provide better delineation of tumors and edema. By combining these diverse sources of information, VGG-based models can provide a more comprehensive understanding of the tumor's characteristics, aiding in more precise classification. Furthermore, data augmentation techniques, such as rotation, flipping, and zooming, are commonly applied to overcome the challenge of limited data in medical imaging. These techniques artificially increase the size of the dataset, improving model robustness and generalization.

The workflow for brain tumor segmentation and classification using VGG models typically follows several steps. First, preprocessing of MRI images is performed to normalize the data, ensuring consistency across different scans. Techniques such as resizing, intensity normalization, and data augmentation are applied. For segmentation, a VGG-16 or VGG-19 model is used as part of a U-Net or FCN architecture, which segments the tumor region from the surrounding brain tissue. The segmented tumor region can then be classified using the same VGG model or a separate classification network, which outputs the tumor’s type or grade. The results of both segmentation and classification are crucial for diagnosing brain tumors, planning treatments such as surgery or radiation therapy, and predicting patient outcomes. Postprocessing techniques, such as morphological operations, are often used to refine the segmentation output, ensuring that the tumor boundaries are smooth and accurate.

While VGG-16 and VGG-19 are powerful tools for both segmentation and classification tasks, there are challenges to overcome. One of the primary challenges is data imbalance, as many datasets, such as those for brain tumor classification, contain far fewer examples of rare tumor types. This imbalance can lead to biased predictions, with the model favoring the more common classes. Techniques like oversampling, under sampling, and synthetic data generation can mitigate this issue. Additionally, the generalization of these models to unseen data is another challenge, especially when MRI images come from different scanners or hospitals. To address this,

techniques like cross-validation, transfer learning, and regularization are often employed to ensure the models perform well on new, unseen data. Another challenge lies in the interpretability of deep learning models, including VGG-16 and VGG-19. Medical practitioners often require an understanding of how the model arrived at a specific diagnosis or segmentation. Efforts to improve the explainability of deep learning models, such as visualization techniques like Grad-CAM, are essential to make these models more trustworthy in clinical settings.

In conclusion, the use of VGG-16 and VGG-19 for brain tumor segmentation and classification has proven to be a highly effective approach in the field of medical image analysis. These deep learning architectures, with their deep layers and ability to capture complex features, are particularly well-suited to the task of analyzing brain MRI scans. By leveraging pre-trained VGG models, medical professionals can efficiently classify and segment brain tumors, assisting in diagnosis and treatment planning. However, challenges such as data imbalance, generalization to new data, and model interpretability must be addressed to ensure these models can be applied effectively in clinical practice. Nonetheless, the ongoing advancement of deep learning techniques and their integration into medical imaging hold significant potential for improving brain tumor diagnosis and patient outcomes.

## 3.2 Disadvantages of VGG-16 AND VGG-19:

- **Data Imbalance –** Many datasets contain fewer examples of rare tumor types, leading to biased predictions favoring more common classes.
- **Generalization Issues –** Models may struggle with unseen data, especially when MRI images come from different scanners or hospitals.
- **Interpretability Challenges –** Deep learning models like VGG-16 and VGG-19 are often considered “black boxes,” making it difficult for medical professionals to understand how a diagnosis or segmentation

was made.

- **Need for Large Datasets –** While transfer learning helps, these models still require significant labeled medical data for fine-tuning.
- **Computational Complexity –** VGG models are deep networks, making them computationally expensive and requiring high processing power.
- **Domain Adaptation Issues –** MRI images from different sources may vary in quality and distribution, requiring additional techniques like cross-validation and regularization for better performance

# CHAPTER 4: PROPOSED METHOD

## 4.1. Segmentation model

3D-Unet U-Net is one of the most popular architectures used for segmentation. It was designed for image segmentation in the biomedical field. It produced great results for cell tracking. It can work with hundreds of examples and produce good results. As it is U-shape so it is called the U-net model.

It consists of two paths: the contracting path and the expanding path. Both paths perform opposite results. The contracting path involves down sampling and down convolution. Expanding paths involves up-sampling and up-convolution. In contracting path feature maps get spatially smaller, whereas in an expanding path, the feature maps are expanded back to their original size.

This model was basically built for 2D images, but by replacing 2D convolutional networks with 3D networks the model can be used for 3D convolution as well. Fig 4.1 shows the architecture of the 3D-Unet deep neural network architecture.

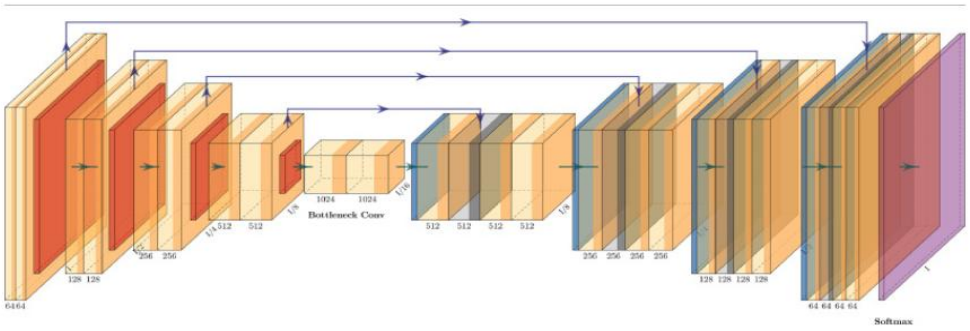


**Fig. 4.1 3D-UNet structure**

The 3D U-Net model is the model introduced in this paper. The models that make up the full tumor detection platform are an image registry model, a 3D U-Net model, and soft dice failure. The first step was to combine 3D image slices from an MRI scan into a single 3D model. Image registration is used to solve misalignment issues during combination.

Following the formation of the 3D model, the 3D model is divided into subsections, each of which is coded in the appendix. The subsections are then

fed into the U-Net model, which produces the segmented model after all of the down and up convolution cycles. The subsections are then merged once more to create a segmented 3D model. The next move is to calculate the damage. 3.2. Classification model.

### 4.1.1. Dataset

The classification model is based on the Brain tumor Classification (MRI) Kaggle dataset. This dataset is split into training and research sets, accumulating 3264 files categorized as glioma, meningioma, pituitary, and no tumor photographs. Since this is a classification model, this dataset aids in the accurate and precise training and testing of the model.

### 4.1.2. Convolutional neural network

Neural network architecture is inspired by the biological human brain. Neural networks are primarily used to quantify vectors, approximate data, cluster data, align patterns, optimize, and classify functions.

Based on their links, the neural network is categorized into three groups, viz., (a) feedback, (b) feedforward, and (c) recurrent networks. Further, a neural network can be classified as a single-layer network or a multilayer neural network.

The picture cannot be scaled in the standard neural network. However, in the convolution of the neural network, pictures can be scaled (i.e. in length, width, and height). The Convolution Neural Network (CNN) consists of an input layer, a convolution layer, and a rectified linear unit (ReLu).

The provided input picture is divided into several small regions of the convolution sheet. In the ReLu layer, element-wise feature activation is performed, and an optional pooling layer could be used. The pooling layer is used primarily for sampling purposes.

A class score or mark score value dependent on chance between 0 and 1, is used in the last layer (i.e., to produce the completely connected layer). Fig 4.2 shows the block diagram of the grouping of brain tumor based on the neuronal network. The classification of brain tumor based on CNN is split into two stages: (a) preparation and (b) research.

The number of photographs is categorized by naming the marks (tumor,

non-tumor images, etc.) into various categories. In the training step, pre-processing, functional extraction and loss function classification are carried out to produce a prediction model.

First, the picture collection is marked for the instruction, and then the image is resized to adjust the image size in the pre-processing process. Finally, for the automated detection of brain tumor the neural convolution network is used.

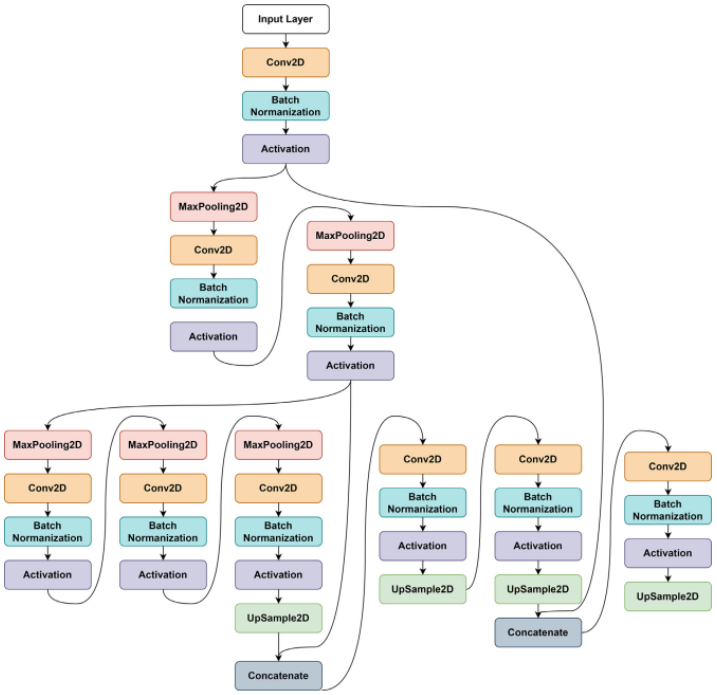


**Fig. 4.2 Segmentation and Detection model**

The brain image dataset used for this model is taken from Kaggle. To use the untrained dataset, the model is trained from layer one until the end layer. This can be very time-consuming and will also affect the outcome. So, for classification measures, a pre-trained model-based brain dataset is used to prevent this issue.

In the proposed model, only the last layer is trained during implementation. As a result, the proposed model has a short computing period with higher efficiency. The loss function is determined by the gradient descent algorithm. The raw pixel image is mapped using a score feature to achieve class results. Quality is calculated by the loss function of a particular set of parameters.

It is dependent on the way induced results are accepted in the training

data with the ground truth marks. In order to increase the precision, calculating the loss function is extremely necessary. When there is a high loss function, the precision will be very poor. Similarly, when the loss function is minimal, the precision will be high. The value for the loss function is determined to estimate the downward gradient algorithm, and it accesses the gradient value to calculate the loss function gradient repeatedly.

### 4.1.3. Proposed Classification model:

The proposed model in this paper is a newly developed CNN architecture. The proposed architecture is novel because it is updated. The design has 16 layers to enable the classifier to efficiently classify the brain tumor images. The configuration of the implemented CNN architecture is presented in Fig. 4.3

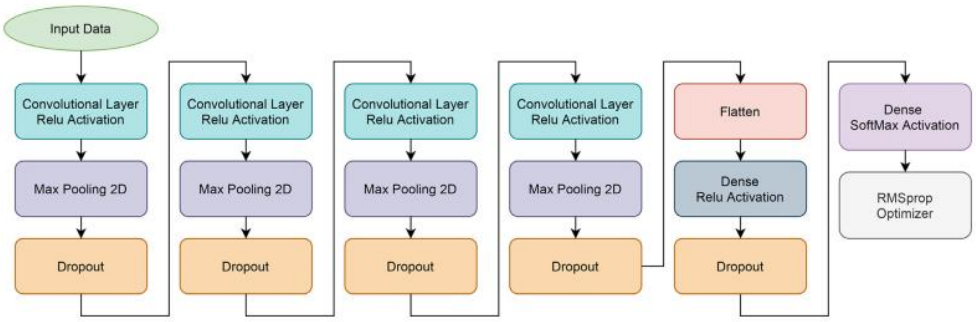


**Fig 4.3 Proposed Classification Model**

The given diagram represents a Convolutional Neural Network (CNN) architecture for brain tumor classification using multiple convolutional layers followed by fully connected layers. Here's how the classification process works:

Step-by-Step Classification Method:

1. **Input Data**

   • The model takes MRI scan images of the brain as input.

2. **Feature Extraction Using Convolutional Layers**

   • The model has multiple Convolutional Layers with ReLU activation, which extract features such as edges, textures, and tumor structures from the image.

   • Each convolutional layer applies a set of filters to detect patterns in the image.

3. **Max Pooling for Dimensionality Reduction**

• After each convolutional layer, a Max Pooling (2D) operation is applied.

• This reduces the spatial dimensions while keeping important features, making the model more efficient.

4. **Dropout for Regularization**

• Dropout layers are used after each pooling layer to prevent overfitting by randomly disabling some neurons during training.

5. **Flattening the Feature Maps**

• After the final convolutional layer, the output is flattened into a one-dimensional vector, preparing it for classification.

6. **Fully Connected (Dense) Layers for Classification**

• A Dense layer with ReLU activation processes the extracted features.

• Another Dense layer with Softmax activation assigns probabilities to different classes (e.g., normal, benign tumor, malignant tumor).

7. **Optimization Using RMSprop**

• The RMSprop optimizer is used to adjust the model's weights, ensuring efficient learning.

## 4.2 Flowchart for proposed method

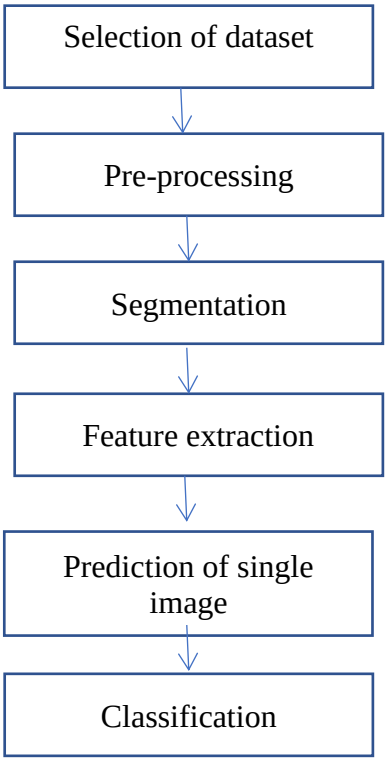


**Fig 4.4 Flowchart of proposed method**

The flowchart represents the step-by-step process for brain tumor segmentation and classification using deep learning techniques. Here's a breakdown of each step:

1. **Selection of Dataset**

• The process begins with choosing a suitable dataset containing brain MRI images.

• Common datasets include BRATS (Brain Tumor Segmentation Challenge), Harvard Brain Dataset, and Kaggle brain tumor datasets.

• These datasets contain labeled MRI scans with tumor annotations.

2. **Pre-processing**

• MRI images are preprocessed to enhance quality and improve model performance.

• Preprocessing steps include:

• Resizing: Standardizing image size for uniform input.

• Normalization: Adjusting pixel intensity values.

• Noise removal: Using filtering techniques to reduce noise.

• Data augmentation: Applying transformations (rotation, flipping) to increase dataset size.

3. **Segmentation**

• Segmentation isolates the tumor region from the MRI scan.

• Techniques used:

Thresholding: Identifying regions with intensity variations.

Edge detection: Using filters like Sobel or Canny to find boundaries.

Deep learning models: U-Net for precise tumor segmentation.

• The output is a binary or multi-class segmentation mask highlighting the tumor.

4. **Feature Extraction**

• Extracts key features from the segmented tumor region for classification.

• Commonly extracted features:

• Shape features: Tumor size, boundary, symmetry.

• Texture features: Edge sharpness, contrast.
• Deep features: Extracted using CNN models like VGG-16, VGG-19, or ResNet.

5. **Prediction of Single Image**
• The trained model predicts the type of tumor for a given MRI scan.
• Transfer learning with pre-trained CNN models helps classify tumors based on extracted features.

6. **Classification**
• The final step classifies the tumor into different categories:
• Benign (non-cancerous)
• Malignant (cancerous)
• Tumor subtypes (glioma, meningioma, metastasis, etc.)
• SoftMax activation function in CNN outputs probability scores for different classes.
• The class with the highest probability is chosen as the final classification.

**4.3 Advantages of proposed system:**

- **Accurate Segmentation** – Captures fine details of brain tumors in 3D space.
- **Handles Complex Structures** – Works well with irregular tumor shapes.
- **Fully Convolutional** – Efficient learning with fewer parameters.
- **Skip Connections** – Preserves spatial information for better segmentation.
- **Automated & Fast** – Reduces manual effort in medical imaging.
- **High Sensitivity** – Detects even small tumors effectively.
- **Better Feature Extraction** – Captures depth information using 3D convolutions.
- **Improved Diagnosis** – Assists radiologists in decision-making.

# CHAPTER 5: SYSTEM REQUIREMENT

## 5.1 HARDWARE REQUIREMENTS:

- System : Pentium Dual Core.
- Hard Disk : 120 GB.
- Monitor : 15" LED
- Input Devices : Keyboard, Mouse
- Ram : 4 GB

## 5.2 SOFTWARE REQUIREMENTS:

- Operating system : Windows 10
- Coding Language : python
- Tool : Python
- Database : Brain MRI scan dataset
- Frontend : GUI (**tkinter** library in python)

# CHAPTER 6: RESULTS AND DISCUSSION

Brain tumor is a deadly disease which causes death to millions every year and timely detection of such tumor can help in reducing risk of losing life. In the past many deep learning algorithms were introduced which can detect tumor and perform classification but its detection rate is low and work only 2-dimension MRI images. Latest technology generating MRI in 3D format and existing UNET segmentation cannot work on 3D MRI images and to solve this issue author of this paper employing 3D-UNET algorithm which will segment out tumor part from brain MRI and then employing 16-layer CNN algorithm to classify or damage brain tumor.

3D-UNET algorithm trained on BRATS2020 dataset to segment out tumor data and then propose 16-layer CNN algorithm trained on 'Brain Tumor MRI Dataset' which consists of 4 different classes listed below

**'Glioma', 'Meningioma', 'No Tumor', 'Pituitary'**

Above dataset can be download from below KAGGLE repository dataset

https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset

Above dataset trained on VGG16 pre-trained model and propose 16 layers CNN model and in both algorithms propose CNN 16-layer algorithm is giving best accuracy. Propose algorithm consist of CNN layer to filter MRI features and to efficiently extract tumor and then MaxPool2d layer will collect filtered features from CNN and then apply Dropout layer to remove irrelevant features. This filtration make propose CNN algorithm to detect and classify tumor 90% accurately.

3D-UNET algorithm can able to train and segment tumor part from 3D images and by seeing this segmented tumor output doctors can easily identify tumor region and based on region they can perform suitable treatment to reduce risk of patient life.

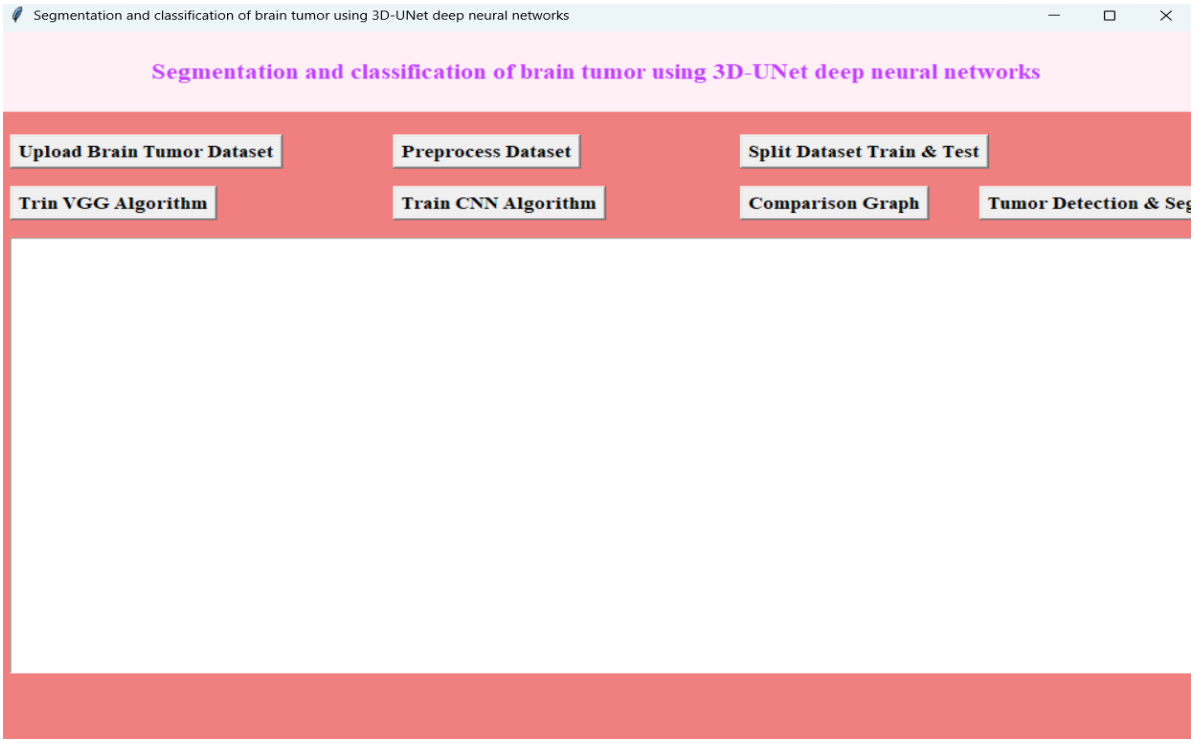


**FIG 6.1: GUI Interface of brain tumor classification**

The above fig 6.1 shows the graphical user interface (GUI) of segmentation and classification of brain tumor using 3D UNET deep neural networks. This interface represents the overall structure of it as shown. And here we click on upload brain tumor dataset.

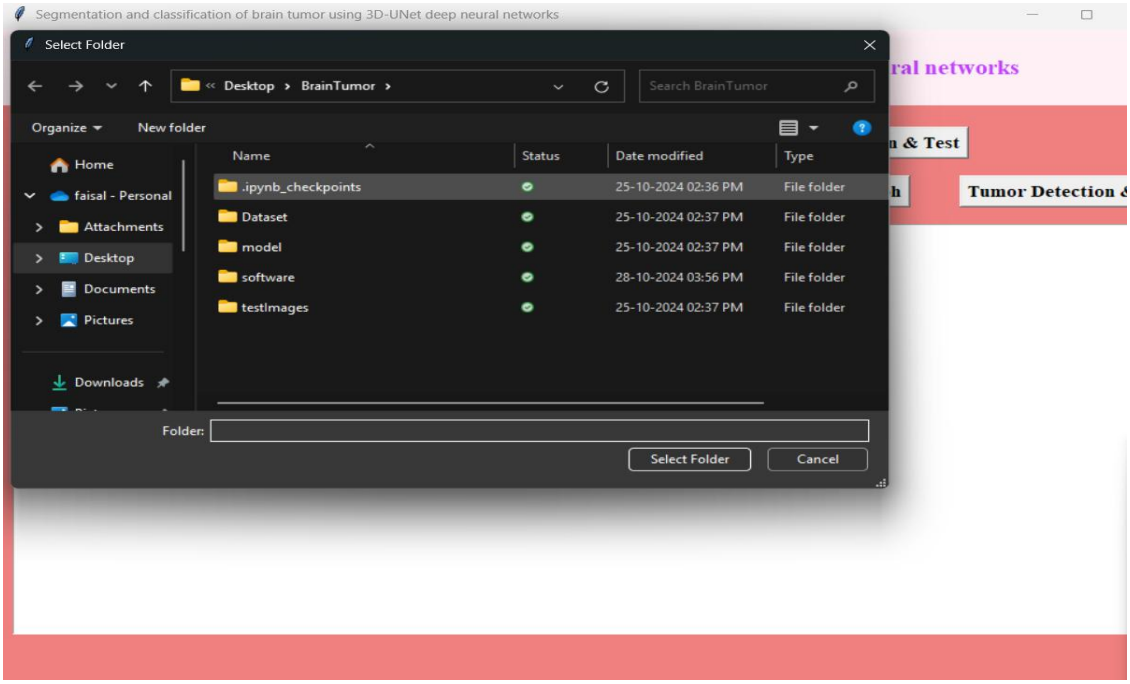


**fig 6.2: Uploading the Brain tumor dataset**

By clicking on the button we get a new window to select the dataset of brain tumor to be uploaded as shown in the fig 6.2


**Fig 6.3: Brain tumor classes and the labelled graph**

The above fig 6.3 shows the brain tumor classes found in the dataset. There are four types of classes are found and they are Glioma, Meningioma, Pituitary and No tumor. And the graph represents the count of dataset in each particular classes. Now click on the preprocess Dataset button for normalization and shuffling of the Dataset.


**Fig 6.4: Normalization and shuffling of the Dataset**

Fig 6.4 shows the completion of normalization and shuffling of the brain tumor dataset which was uploaded and now we have to split the dataset for training and testing purpose by clicking on split dataset train & test button.

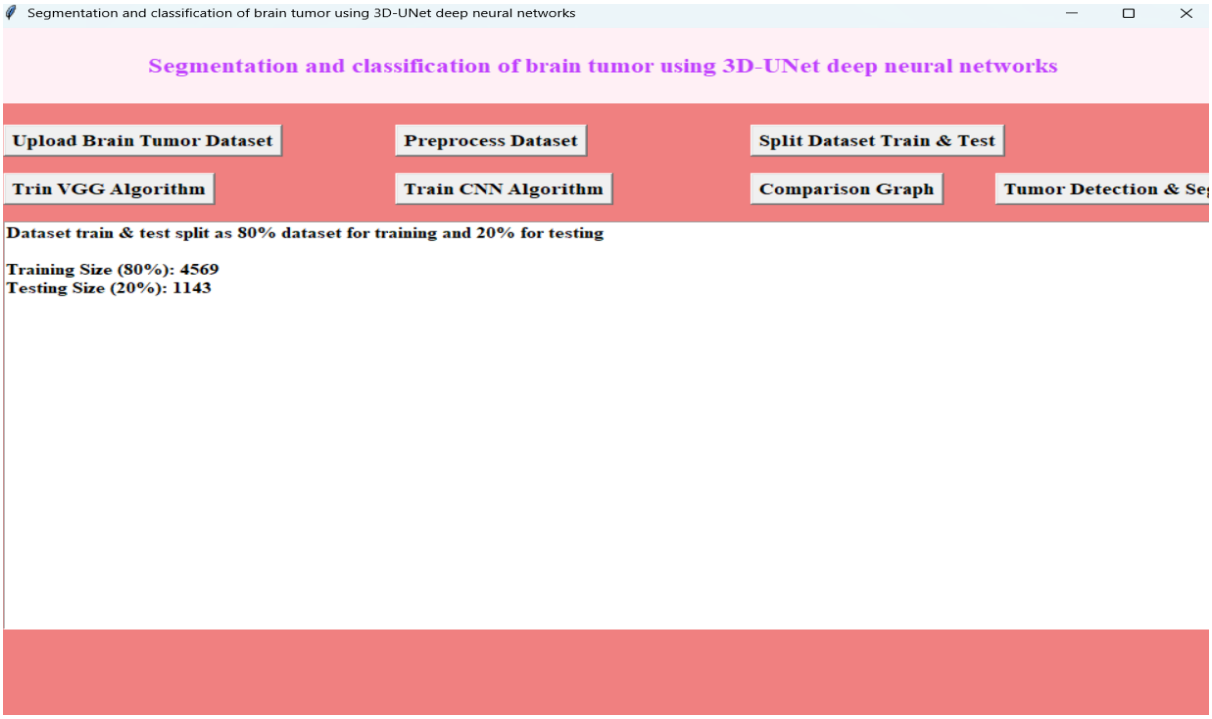


**Fig 6.5: Splitting of Dataset for training and testing**

Fig 6.5 shows the dataset train and test split as 80% dataset for training purpose and 20% for testing purpose as shown above. Now we need to test the accuracy of the train VGG algorithm by clicking on train VGG algorithm button.

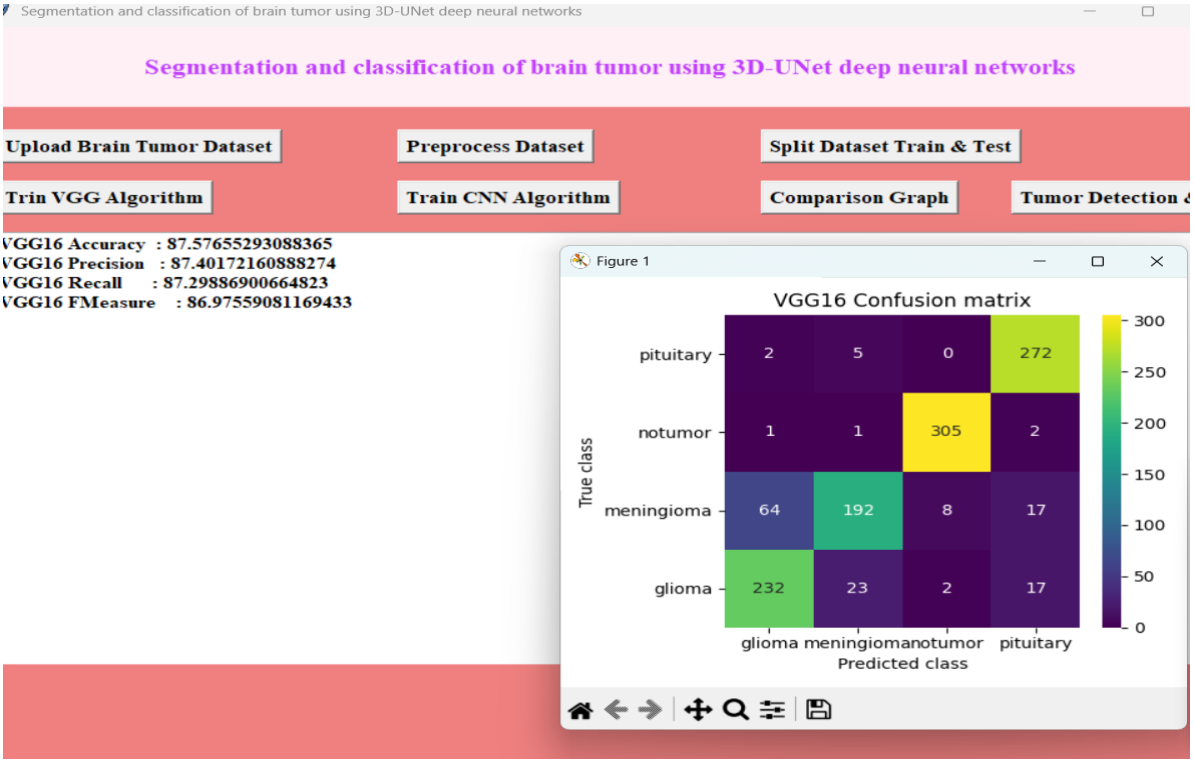


**Fig 6.6: VGG16 confusion matrix**

Fig 6.6 shows the accuracy, precision, recall and Fmeasure of VGG16 algorithm with the confusion matrix graph of the predicted class shown. Now click on train CNN algorithm which is the proposed algorithm for classification of the brain tumor.

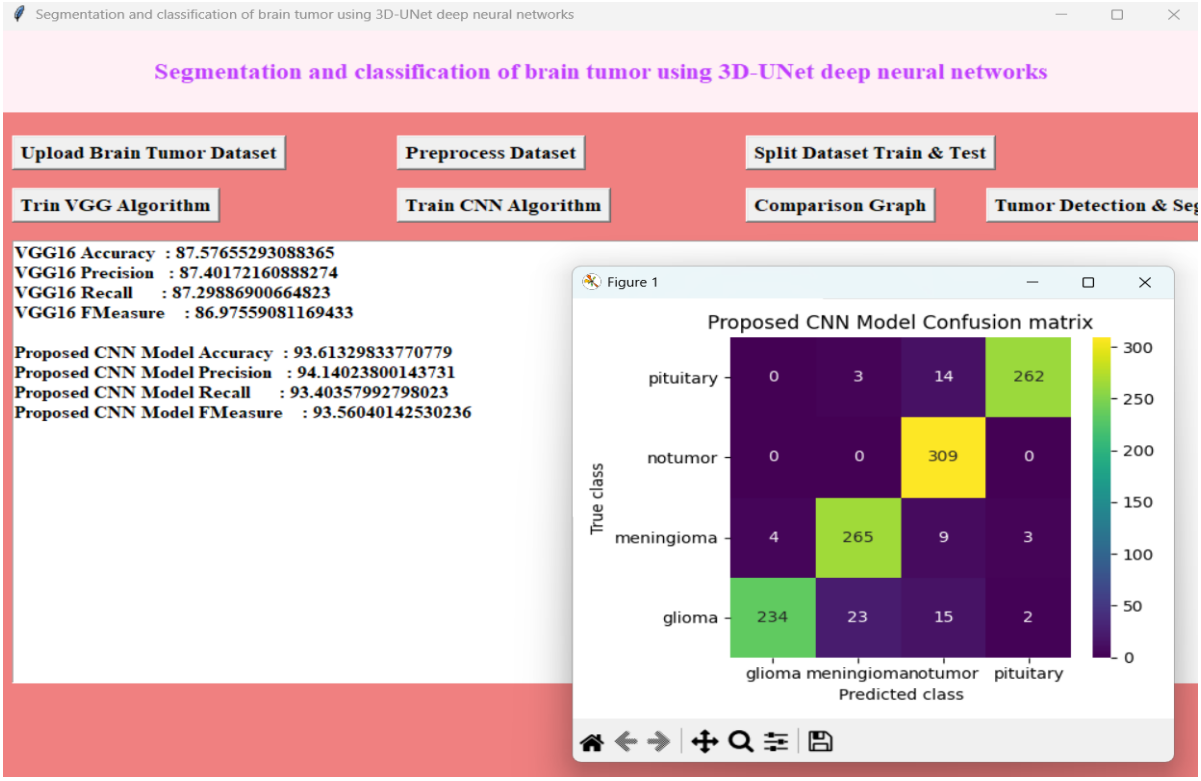


**Fig 6.7: Proposed CNN model confusion matrix**

This is the comparition of accuracy and various parameters between the VGG16 and proposed CNN model algorithm.and it has been observed that CNN model algorithm is more accurate than the VGG16 algorithm. And the proposed CNN model confusion matrix shows the less errors when compared to the VGG16 algorithm.

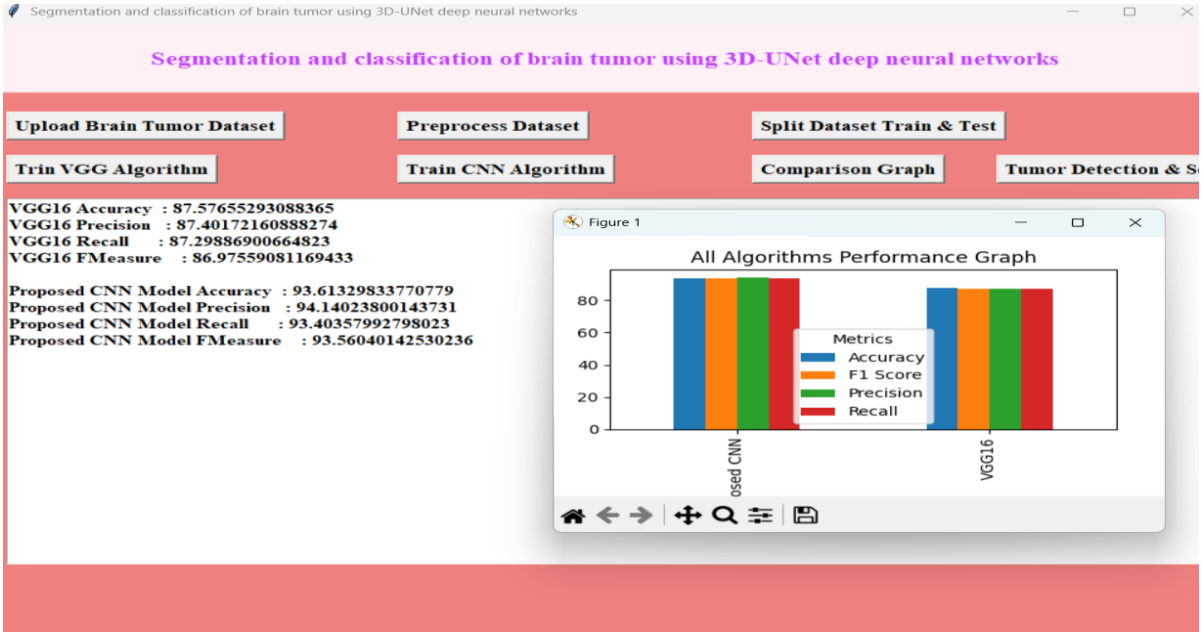


**Fig 6.8: Performance graph of Algorithms.**

Fig 6.8 shows the all algorithms performance graph based on various parameters such as Accuracy, f1 score, precision, recall as shown.

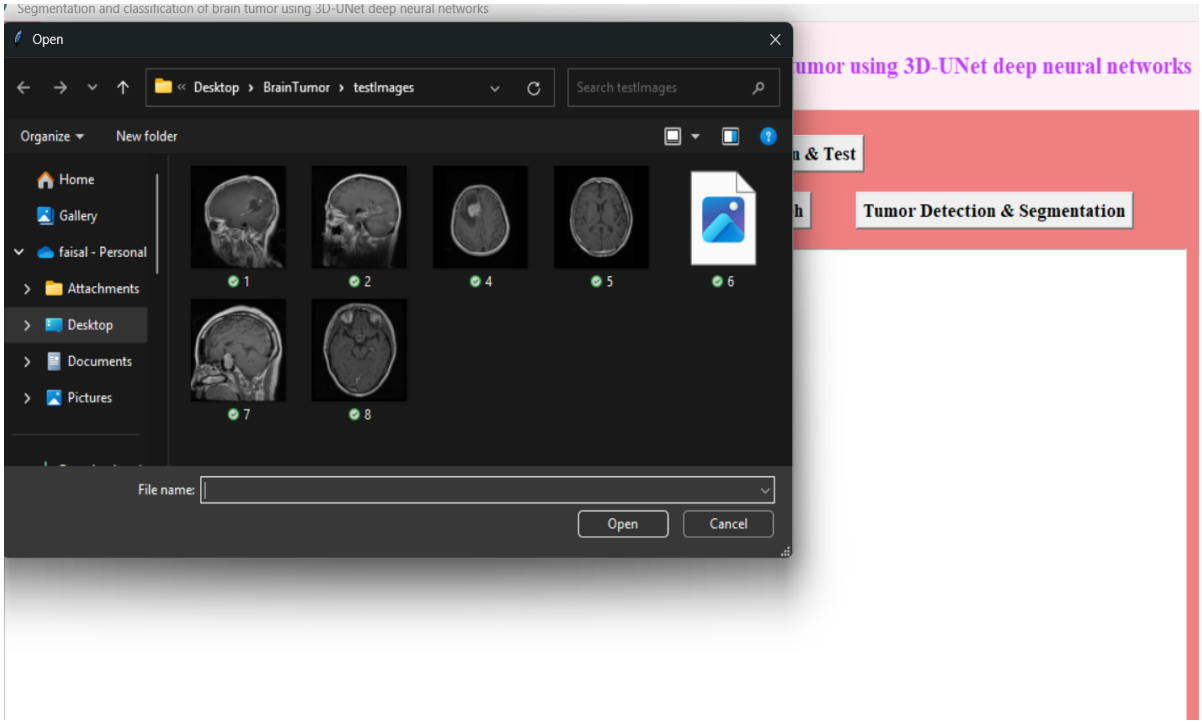


**Fig 6.9: Uploading of MRI image for testing purpose**

Fig 6.9 shows the test of the dataset by selecting a particular MRI image for detection and classification.

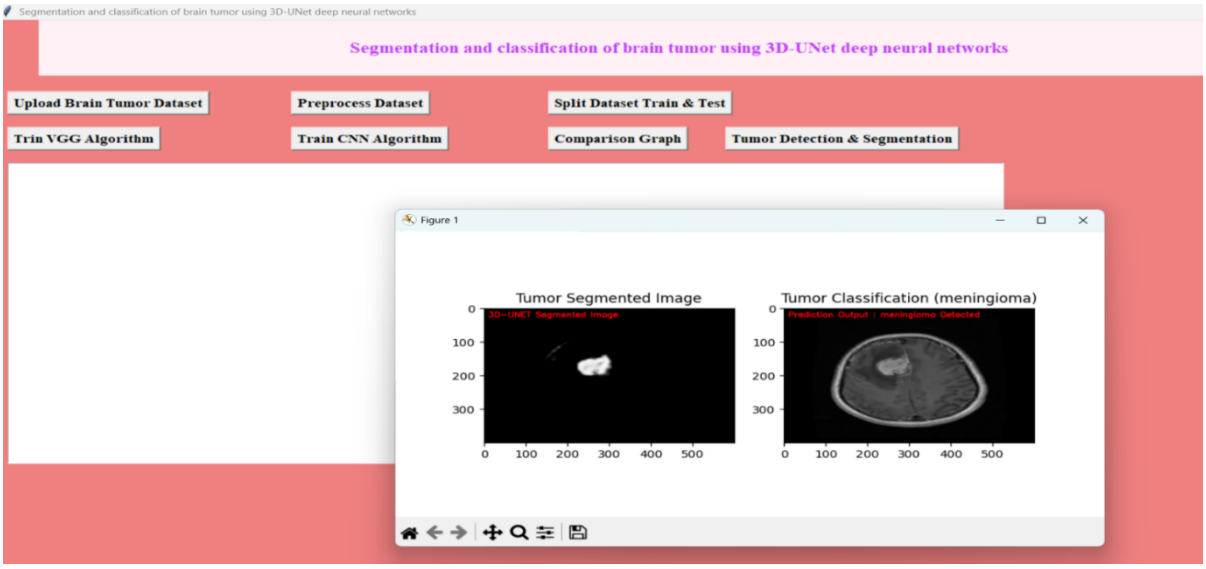


**Fig 6.10: Segmentation and Classification of Meningioma**


**Fig 6.11: Segmentation and Classification of Glioma**

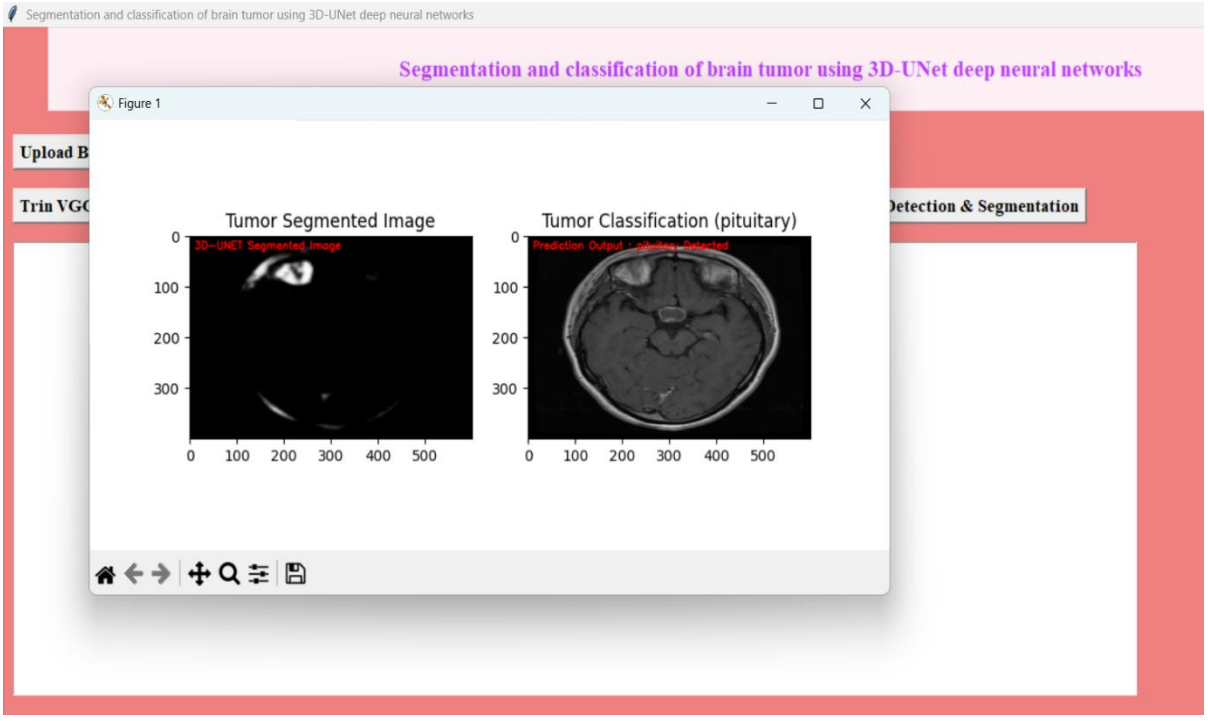


**Fig 6.12: Segmentation and Classification of Pituitary**

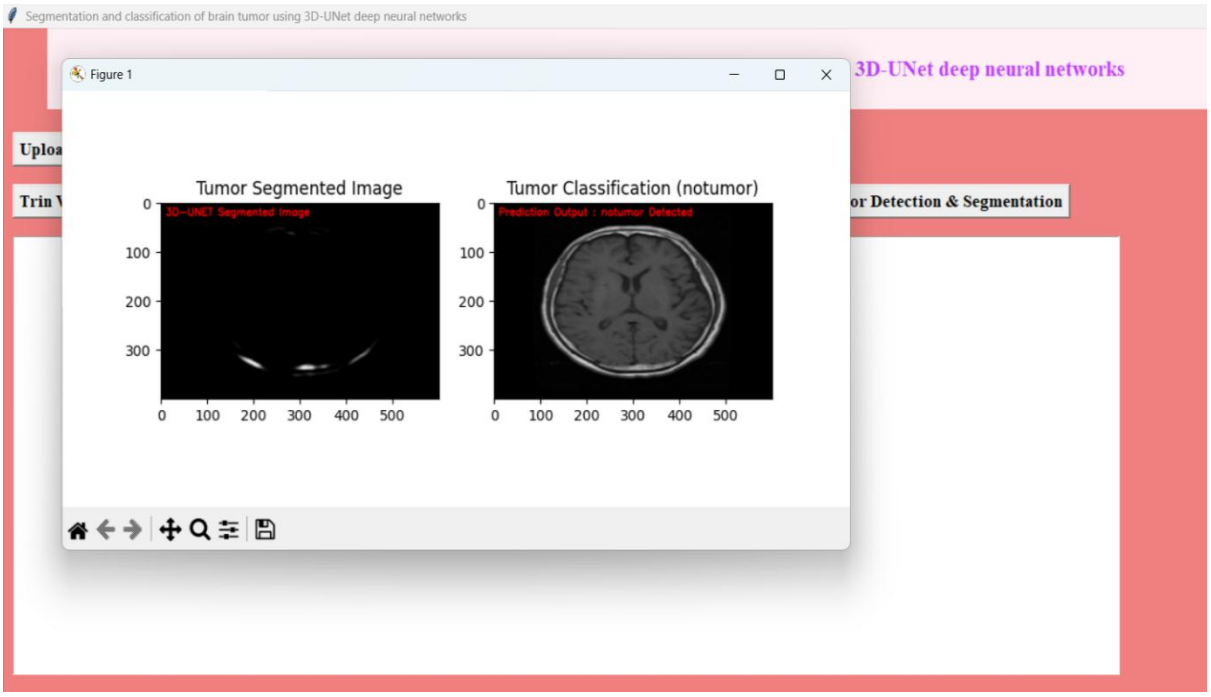


**Fig 6.13: Segmentation and Classification of Notumor**

The fig 6.10, 6.11, 6.12, 6.13 shows the final output of the brain tumor as shown. Here we had segmented and classified the Brain tumor in four different classes and they are Glioma, Meningioma, pituitary and no tumor. Here the tumor has been observed and segmented the region where the tumor is present in the brain with classification of type of the brain tumor.

# CHAPTER 7: CONCLUSION AND FUTURE SCOPE

## 7.1 CONCLUSION:

The segmentation and classification of brain tumors using 3D U-Net deep neural networks have demonstrated significant improvements in medical image analysis. By leveraging deep learning, 3D U-Net effectively captures spatial and contextual information, leading to accurate tumor detection and segmentation from MRI scans. Compared to traditional methods, this approach enhances precision, reduces manual intervention, and minimizes human error, making it highly valuable for clinical applications. However, challenges such as data imbalance, limited annotated datasets, and high computational requirements still exist. Addressing these issues can further improve model performance and reliability.

## 7.2 FUTURE SCOPE:

In the future, advancements in deep learning architectures, such as integrating attention mechanisms or hybrid models, can enhance accuracy and robustness. Expanding datasets with multi-center and multi-modality MRI scans can improve generalization across diverse patient populations. Optimizing computational efficiency will enable real-time tumor detection, making it feasible for clinical deployment. Additionally, integrating the model into hospital workflows with user-friendly interfaces can assist radiologists in faster and more reliable diagnoses. Further research in explainability and interpretability of deep learning models can also increase trust among medical professionals. With continuous improvements, 3D U-Net-based brain tumor segmentation has the potential to revolutionize early detection and treatment planning in neuro-oncology.

# CHAPTER 8: BIBLIOGRAPHY


[1]. G. Alain and Y. Bengio, "What regularized auto-encoders gain from the data making allotment," J. Mach. Learn. Res., vol. 15, no. 1, pp. 3563–3593, 2014.

[2]. M. R. Avendi, A. Kheradvar, and H. Jafarkhani, "A solidified significant learning and deformable-model approach to manage totally customized division of the left ventricle in heart MRI," MedIA Image Anal., vol. 30, pp. 108–119, May 2016.

[3]. M. R. Avendi, A. Kheradvar, and H. Jafarkhani, "Modified division of the right ventricle from heart MRI using a learning-based system," in Magnetic Resonance in Medicine. New York, NY, USA: Wiley, Feb. 2017, doi: 10.1002/mrm.26631.

[4]. W. Bai et al., "A bi-ventricular cardiovascular guide book worked from 1000+ significant standards MR pictures of sound subjects and an examination of shape and development," MedIA Image Anal., vol. 26, no. 1, pp. 133–145, 2015.

[5]. W. Bai et al., "A probabilistic fix based name blend model for multiatlas division with selection refinement: Application to cardiovascular MR pictures," IEEE Trans. Medication. Picture. vol. 32, no. 7, pp. 1302–1315, Jul. 2013.

[6]. D. Barbosa et al., "Speedy and totally customized 3-D echocardiographic division using B-spline unequivocal unique surfaces: Feasibility study and endorsement in a clinical setting," Ultrasound Med. Biol., vol. 39, no. 1, pp. 89–101, 2013. BenTaieb and G. Hamarneh, "Topology careful totally convolutional frameworks for histology organ division," in Proc. Int. Conf. MICCAI, 2016, pp. 460–468.

[7]. O. Bernard et al., "Standardized evaluation system for left ventricular division computations in 3D echocardiography," IEEE Trans. Prescriptions. Picture., vol. 35, no. 4, pp. 967–977, Apr. 2016.

[8]. F. Chen, H. Yu, R. Hu, and X. Zeng, "Significant learning shape priors for object division," in Proc. IEEE CVPR, Jun. 2013, pp. 1870–1877.

[9]. H. Chen, X. Qi, L. Yu, Q. Dou, J. Qin, and P.- A. Heng, "DCAN: Deep structure careful frameworks for object model division from histology pictures," MedIA Image Anal., vol. 36, pp. 135–146, Feb. 2017.

[10]. H. Chen, Y. Zheng, J.- H. Park, P.- A. Heng, and S. K. Zhou, "Iterative multi-territory regularized significant learning for anatomical structure acknowledgment and division from ultrasound pictures," in Proc. Int. Conf. MICCAI, 2016, pp. 487–495.

[11]. Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning thick volumetric division from lacking remark," in Proc. MICCAI, 2016, pp. 424–432.

A. de Marvao et al., "People based examinations of myocardial hypertrophy: High objectives cardiovascular alluring resonation map books improve quantifiable power," J. Cardiovascular Magn. Reson.vol. 16, no. 1, p. 16, 2014.

[12]. S. Eslami and C. Williams, "A generative model for parts-based thing division," in Proc. NIPS, 2012, pp. 100–107.

[13]. S. M. A. Eslami, N. Heess, and J. Winn, "The shape Boltzmann machine: A strong model of thing shape," in Proc. IEEE Conf. Comput. Vis. Model Recognit., Jun. 2012, pp. 406–413.

[14]. R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta, "Learning an envisioned and generative vector depiction for objects," in Proc. ECCV, 2016, pp. 484–499.

[15]. H. Greenspan, "Super-objectives in clinical imaging," Compute. J., vol. 52, no. 1, pp. 43–63, 2009.

[16]. J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual disasters for consistent style move and super-objectives," in Proc. ECCV, 2016, pp. 694–711.

[17]. K. Kamnitsas et al., "Gainful multi-scale 3D CNN with totally related CRF for precise brain sore division," Media Image Anal., vol. 36, pp. 61–78, Feb. 2017.

[18]. J. Lafferty, A. McCallum, and F. C. N. Pereira, "Prohibitive unpredictable fields: Probabilistic models for separating and naming progression data," in Proc. ICML, vol. 1. 2001, pp. 282–289.

[19]. R. M. Lang et al., "EAE/ASE proposition for picture making sure about

and show using three-dimensional echocardiography," J. Amer. Soc. Echocardiogr., vol. 25, no. 1, pp. 1–46, 2012.

[20]. C. Ledig et al., (Sep. 2016). "Photo handy single picture super objectives using a generative poorly arranged framework."[Online]. Available: https://arxiv.org/abs/1609.04802

[21]. K. Li, B. Hariharan, and J. Malik, "Iterative event division," in Proc. CVPR, 2016, pp. 3659–3667.

[22]. G. Litjens et al., (2017). "An audit on significant learning in clinical picture assessment."[Online]. Open: https://arxiv.org/abs/1702.05747

[23]. J. Long, E. Shelhamer, and T. Darrell, "Totally convolutional frameworks for semantic division," in Proc. CVPR, 2015, pp. 3431–3440.

[24]. P. Luc, C. Couprie, S. Chintala, and J. Verbeek. (Dec. 2016). "Semantic division using poorly arranged frameworks." [Online]. Open: https://arXiv preprint arXiv: 1611.08408

[25]. J. Masci, U. Meier, D. Cire¸san, and J. Schmidhuber, "Stacked convolutional auto-encoders for dynamic segment extraction," in Proc. Int. Conf. Workmanship if. Neural Netw. 2011, pp. 52–59.

[26]. M. S. Nosrati and G. Hamarneh. (2016). "Merging prior data in clinical picture division: An examination." [Online]. Open: https://arxiv.org/abs/1607.01092

[27]. O. Oktay et al., "Isolated decision forests for definite anatomical achievement limitation in heart pictures," IEEE Trans. Drugs. Picture., vol. 36, no. 1, pp. 332–342, Jan. 201

# APPENDIX

## Introduction

One of the most popular languages is Python. Guido van Rossum released this language in 1991. Python is available on the Mac, Windows, and Raspberry Pi operating systems. The syntax of Python is simple and identical to that of English. When compared to Python, it was seen that the other language requires a few extra lines.

It is an interpreter-based language because code may be run line by line after it has been written. This implies that rapid prototyping is possible across all platforms. Python is a big language with a free, binary-distributed interpreter standard library.

It is inferior to maintenance that is conducted and is straightforward to learn. It is an object-oriented, interpreted programming language. It supports several different programming paradigms in addition to object-oriented programming, including functional and procedural programming.

It supports several different programming paradigms in addition to object-oriented programming, including practical and procedural programming. Python is mighty while maintaining a relatively straightforward syntax. Classes, highly dynamic data types, modules, and exceptions are covered. Python can also be utilized by programs that require programmable interfaces as an external language.

Here are some key features and characteristics of Python:

- Readability: Python emphasizes code readability with its clean and intuitive syntax. It uses indentation and whitespace to structure code blocks, making it easy to understand and maintain.
- Easy to Learn: Python's simplicity and readability make it an excellent choice for beginners. Its straightforward syntax and extensive documentation make it accessible for newcomers to programming.
- Interpreted Language: Python is an interpreted language, meaning that it doesn't need to be compiled before running. The Python interpreter

reads and executes the code directly, making the development process faster and more interactive.

- Cross-platform Compatibility: Python is available for major operating systems like Windows, macOS, and Linux. This cross-platform compatibility allows developers to write code once and run it on different platforms without modifications.
- Large Standard Library: Python comes with a vast standard library that provides ready-to-use modules and functions for various tasks. It covers areas such as file I/O, networking, regular expressions, databases, and more, saving developers time and effort.
- Extensible and Modular: Python supports modular programming, enabling developers to organize code into reusable modules and packages. Additionally, Python allows integrating modules written in other languages, such as C or C++, providing flexibility and performance optimizations.
- Wide Range of Libraries and Frameworks: Python has a vibrant ecosystem with numerous third-party libraries and frameworks. These libraries, such as NumPy, pandas, TensorFlow, and Django, extend Python's capabilities for specific domains, making it a powerful tool for diverse applications.
- Object-Oriented: Python supports object-oriented programming (OOP) principles, allowing developers to create and work with classes and objects. OOP provides a structured approach to code organization, promoting code reuse and modularity.
- Dynamic Typing: Python is dynamically typed, meaning variable types are determined at runtime. Developers do not need to declare variable types explicitly, which enhances flexibility and simplifies code writing.

## Installation

To install Python on your computer, follow these basic steps:

- Step 1: Visit the Python website Go to the official Python website at https://www.python.org/.

- Step 2: Select the operating system Choose the appropriate installer for your operating system. Python supports Windows, macOS, and various Linux distributions. Make sure to select the correct version that matches your operating system.
- Step 3: Check which version of Python is installed; if the 3.7.0 version is not there, uninstall it through the control panel and
- Step 4: Install Python 3.7.0 using Cmd.
- Step 5: Install the all libraries that required to run the project
- Step 6: Run

## Python Features:

1) **Easy:** Because Python is a more accessible and straightforward language, Python programming is easier to learn.
2) **Interpreted language:** Python is an interpreted language, therefore it can be used to examine the code line by line and provide results.
3) **Open Source:** Python is a free online programming language since it is open-source.
4) **Portable:** Python is portable because the same code may be used on several computer standard
5) **libraries:** Python offers a sizable library that we may utilize to create applications quickly.
6) **GUI:** It stands for GUI (Graphical User Interface)
7) **Dynamical typed:** Python is a dynamically typed language, therefore the type of the value will be determined at runtime.

## Libraries :

pip install h5py==3.10.0
pip install ipykernel==6.5.0
pip install ipython==7.34.0
pip install ipython-genutils==0.2.0
pip install ipywidgets==8.1.1
pip install joblib==1.3.2
pip install jupyter==1.0.0

pip install jupyter-console==6.4.0
pip install jupyter-events==0.9.0
pip install jupyter-lsp==2.2.1
pip install jupyter_client==7.4.9
pip install jupyter_core==5.5.0
pip install jupyter_server==2.11.1
pip install jupyter_server_terminals==0.4.4
pip install jupyterlab==4.0.9
pip install jupyterlab-widgets==1.0.0
pip install jupyterlab_pygments==0.3.0
pip install jupyterlab_server==2.25.2
pip install keras==2.14.0
pip install matplotlib==3.7.1
pip install nbclient==0.9.0
pip install nbconvert==7.11.0
pip install nbformat==5.9.2
pip install nibabel==5.2.0
pip install notebook==7.0.6
pip install notebook_shim==0.2.3
pip install numpy==1.23.5
pip install opencv-contrib-python==4.8.0.76
pip install opencv-python==4.8.0.76
pip install pandas==2.1.3
pip install Pillow==10.1.0
pip install protobuf==4.25.1
pip install scikit-learn==1.2.2
pip install scipy==1.11.3
pip install seaborn==0.10.1
pip install tensorflow==2.14.0

## Source code:

```
#import required libraries files
import os
import numpy as np
from keras.utils import to_categorical
from keras.layers import  MaxPooling2D
from keras.layers import Dense, Dropout, Activation, Flatten, GlobalAveragePooling2D, BatchNormalization
from keras.layers import Convolution2D
from keras.models import Sequential, load_model, Model
import pickle
from keras.applications import VGG16
from sklearn.model_selection import train_test_split
from keras.callbacks import ModelCheckpoint
import keras
from sklearn.metrics import accuracy_score
from keras.layers import Conv2D, MaxPool2D, Flatten, Dense, InputLayer, BatchNormalization, Dropout
from sklearn.metrics import confusion_matrix #class to calculate accuracy and other metrics
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
import seaborn as sns
import nibabel as nib
import cv2
import numpy as np
import keras
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score
from tensorflow.keras.models import *
from tensorflow.keras.layers import *
from tensorflow.keras.optimizers import *
```

```
from tensorflow.keras.callbacks import ModelCheckpoint, ReduceLROnPlateau, EarlyStopping, TensorBoard
dataset_path = "Dataset"
def getID(name): #function to get ID of the MRI view as label
    index = 0
    for i in range(len(labels)):
        if labels[i] == name:
            index = i
            break
    return index
#function to read labels from dataset
labels = []
for root, dirs, directory in os.walk(dataset_path):#now loop all files and get labels and then display all birds names
    for j in range(len(directory)):
        name = os.path.basename(root)
        if name not in labels:
            labels.append(name)
print("Brain Tumor Classes Found in Dataset")
print(labels)
#now load dataset images
if os.path.exists('model/X.txt.npy'):#if dataset already process then load load it
    X = np.load('model/X.txt.npy')
    Y = np.load('model/Y.txt.npy')
else: #if not process the loop all images from dataset
    X = []
    Y = []
    for root, dirs, directory in os.walk(dataset_path):#loop all images from dataset
        for j in range(len(directory)):
            name = os.path.basename(root)
            if 'Thumbs.db' not in directory[j]:
```

```
            img = cv2.imread(root+"/"+directory[j])#read images from looping
path
            img = cv2.resize(img, (32,32))#resize images
            X.append(img)#add image features to X
            label = getID(name)#get Image ID
            Y.append(label) #add image id as label
    X = np.asarray(X)
    Y = np.asarray(Y)
    np.save('model/X.txt',X)
    np.save('model/Y.txt',Y)
print("Dataset MRI Images Loading Completed")
print("Total images found in dataset : "+str(X.shape[0]))
#plot graph of different labels found in dataset
unique, count = np.unique(Y, return_counts = True)
height = count
bars = labels
y_pos = np.arange(len(bars))
plt.figure(figsize=(6,3))
plt.bar(y_pos, height)
plt.xticks(y_pos, bars)
plt.xlabel("Alzheimer  Names")
plt.ylabel("Count")
plt.title("Dataset Class Label Graph")
plt.show()
#defining layers for 3D-UNET CNN model
def build_unet(inputs, ker_init, dropout):
    #defining conv3d cnn layer to build 3dunet
    conv1 = Conv3D(32, (3, 3, 3), activation = 'relu', padding = 'same',
kernel_initializer = ker_init)(inputs)
    conv1 = Conv3D(32, (3, 3, 3), activation = 'relu', padding = 'same',
kernel_initializer = ker_init)(conv1)

    pool = MaxPooling3D(pool_size=(2, 2, 2))(conv1)
```

```
    conv = Conv3D(64, (3, 3, 3), activation = 'relu', padding = 'same', kernel_initializer = ker_init)(pool)
    conv = Conv3D(64, (3, 3, 3), activation = 'relu', padding = 'same', kernel_initializer = ker_init)(conv)

    pool1 = MaxPooling3D(pool_size=(2, 2, 2))(conv)
    conv2 = Conv3D(128, (3, 3, 3), activation = 'relu', padding = 'same', kernel_initializer = ker_init)(pool1)
    conv2 = Conv3D(128, (3, 3, 3), activation = 'relu', padding = 'same', kernel_initializer = ker_init)(conv2)

    pool2 = MaxPooling3D(pool_size=(2, 2, 2))(conv2)
    conv3 = Conv3D(256, (3, 3, 3), activation = 'relu', padding = 'same', kernel_initializer = ker_init)(pool2)
    conv3 = Conv3D(256, (3, 3, 3), activation = 'relu', padding = 'same', kernel_initializer = ker_init)(conv3)


    pool4 = MaxPooling3D(pool_size=(2, 2, 2))(conv3)
    conv5 = Conv3D(512, (3, 3, 3), activation = 'relu', padding = 'same', kernel_initializer = ker_init)(pool4)
    conv5 = Conv3D(512, (3, 3, 3), activation = 'relu', padding = 'same', kernel_initializer = ker_init)(conv5)
    drop5 = Dropout(dropout)(conv5)

    up7 = Conv3D(256, (2,2,2), activation = 'relu', padding = 'same', kernel_initializer = ker_init)(UpSampling3D(size = (2,2,2))(drop5))
    merge7 = concatenate([conv3,up7])
    conv7 = Conv3D(256, (3, 3, 3), activation = 'relu', padding = 'same', kernel_initializer = ker_init)(merge7)
    conv7 = Conv3D(256, (3, 3, 3), activation = 'relu', padding = 'same', kernel_initializer = ker_init)(conv7)
```

```
    up8 = Conv3D(128, (2,2,2), activation = 'relu', padding = 'same', kernel_initializer = ker_init)(UpSampling3D(size = (2,2,2))(conv7))
    merge8 = concatenate([conv2,up8])
    conv8 = Conv3D(128, (3, 3, 3), activation = 'relu', padding = 'same', kernel_initializer = ker_init)(merge8)
    conv8 = Conv3D(128, (3, 3, 3), activation = 'relu', padding = 'same', kernel_initializer = ker_init)(conv8)

    up9 = Conv3D(64, (2,2,2), activation = 'relu', padding = 'same', kernel_initializer = ker_init)(UpSampling3D(size = (2,2,2))(conv8))
    merge9 = concatenate([conv,up9])
    conv9 = Conv3D(64, (3, 3, 3), activation = 'relu', padding = 'same', kernel_initializer = ker_init)(merge9)
    conv9 = Conv3D(64, (3, 3, 3), activation = 'relu', padding = 'same', kernel_initializer = ker_init)(conv9)

    up = Conv3D(32, (2,2,2), activation = 'relu', padding = 'same', kernel_initializer = ker_init)(UpSampling3D(size = (2,2,2))(conv9))
    merge = concatenate([conv1,up])
    conv = Conv3D(32, (3, 3, 3), activation = 'relu', padding = 'same', kernel_initializer = ker_init)(merge)
    conv = Conv3D(32, (3, 3, 3), activation = 'relu', padding = 'same', kernel_initializer = ker_init)(conv)

    conv10 = Conv3D(4, (1,1,1), activation = 'softmax')(conv)

    return Model(inputs = inputs, outputs = conv10)

# dice loss as defined above for 4 classes
def dice_coef(y_true, y_pred, smooth=1.0):
    class_num = 4
    for i in range(class_num):
        y_true_f = K.flatten(y_true[:,:,:,i])
```

```
        y_pred_f = K.flatten(y_pred[:,:,:,i])
        intersection = K.sum(y_true_f * y_pred_f)
        loss = ((2. * intersection + smooth) / (K.sum(y_true_f) + K.sum(y_pred_f)
+ smooth))
    #                K.print_tensor(loss, message='loss value for class {} :
'.format(SEGMENT_CLASSES[i]))
        if i == 0:
            total_loss = loss
        else:
            total_loss = total_loss + loss
    total_loss = total_loss / class_num
#    K.print_tensor(total_loss, message=' total dice coef: ')
    return total_loss

# define per class evaluation of dice coef
def dice_coef_necrotic(y_true, y_pred, epsilon=1e-6):
    intersection = K.sum(K.abs(y_true[:,:,:,1] * y_pred[:,:,:,1]))
    return (2. * intersection) / (K.sum(K.square(y_true[:,:,:,1])) +
K.sum(K.square(y_pred[:,:,:,1])) + epsilon)

def dice_coef_edema(y_true, y_pred, epsilon=1e-6):
    intersection = K.sum(K.abs(y_true[:,:,:,2] * y_pred[:,:,:,2]))
    return (2. * intersection) / (K.sum(K.square(y_true[:,:,:,2])) +
K.sum(K.square(y_pred[:,:,:,2])) + epsilon)

def dice_coef_enhancing(y_true, y_pred, epsilon=1e-6):
    intersection = K.sum(K.abs(y_true[:,:,:,3] * y_pred[:,:,:,3]))
    return (2. * intersection) / (K.sum(K.square(y_true[:,:,:,3])) +
K.sum(K.square(y_pred[:,:,:,3])) + epsilon)

# Computing Precision
def precision(y_true, y_pred):
        true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
```

```
        predicted_positives = K.sum(K.round(K.clip(y_pred, 0, 1)))
        precision = true_positives / (predicted_positives + K.epsilon())
        return precision

# Computing Sensitivity
def sensitivity(y_true, y_pred):
    true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
    possible_positives = K.sum(K.round(K.clip(y_true, 0, 1)))
    return true_positives / (possible_positives + K.epsilon())

# Computing Specificity
def specificity(y_true, y_pred):
    true_negatives = K.sum(K.round(K.clip((1-y_true) * (1-y_pred), 0, 1)))
    possible_negatives = K.sum(K.round(K.clip(1-y_true, 0, 1)))
    return true_negatives / (possible_negatives + K.epsilon())
#create & load unet 3d model
input_layer = Input((128, 128, 128, 3))
unet_model = build_unet(input_layer, 'he_normal', 0.2)
unet_model                                                                    =
keras.models.load_model('model/model_per_class.h5',custom_objects={
'accuracy' : keras.metrics.MeanIoU(num_classes=4),
                                    "dice_coef": dice_coef,
                                    "precision": precision,
                                    "sensitivity":sensitivity,
                                    "specificity":specificity,
                                    "dice_coef_necrotic": dice_coef_necrotic,
                                    "dice_coef_edema": dice_coef_edema,
                                    "dice_coef_enhancing": dice_coef_enhancing
                                   }, compile=False)
unet_model.summary()
#dataset preprocessing such as shuffling and normalization
X = X.astype('float32')
X = X/255 #normalizing images
```

```
indices = np.arange(X.shape[0])
np.random.shuffle(indices)#shuffling images
X = X[indices]
Y = Y[indices]
Y = to_categorical(Y)
print("Dataset Normalization & Shuffling Process completed")
#now splitting dataset into train & test
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2) #split dataset into train and test
print()
print("Dataset train & test split as 80% dataset for training and 20% for testing")
print("Training Size (80%): "+str(X_train.shape[0])) #print training and test size
print("Testing Size (20%): "+str(X_test.shape[0]))
print()
#define global variables to calculate and store accuracy and other metrics
precision = []
recall = []
fscore = []
accuracy = []
#function to calculate various metrics such as accuracy, precision etc
def calculateMetrics(algorithm, predict, testY):
    p = precision_score(testY, predict,average='macro') * 100
    r = recall_score(testY, predict,average='macro') * 100
    f = f1_score(testY, predict,average='macro') * 100
    a = accuracy_score(testY,predict)*100
    print(algorithm+' Accuracy  : '+str(a))
    print(algorithm+' Precision   : '+str(p))
    print(algorithm+' Recall      : '+str(r))
    print(algorithm+' FMeasure    : '+str(f))
    accuracy.append(a)
    precision.append(p)
```

```
    recall.append(r)
    fscore.append(f)
    conf_matrix = confusion_matrix(testY, predict)
    plt.figure(figsize =(5, 4))
    ax = sns.heatmap(conf_matrix, xticklabels = labels, yticklabels = labels, annot = True, cmap="viridis" ,fmt ="g");
    ax.set_ylim([0,len(labels)])
    plt.title(algorithm+" Confusion matrix")
    plt.ylabel('True class')
    plt.xlabel('Predicted class')
    plt.show()
#train VGG16 on processed traion images
vgg16 = VGG16(input_shape=(X_train.shape[1], X_train.shape[2], X_train.shape[3]), include_top=False, weights='imagenet')
for layer in vgg16.layers:
    layer.trainable = False
vgg16_model = Sequential()
vgg16_model.add(vgg16)
vgg16_model.add(Convolution2D(32, (1 , 1), input_shape = (X_train.shape[1], X_train.shape[2], X_train.shape[3]), activation = 'relu'))
vgg16_model.add(MaxPooling2D(pool_size = (1, 1)))
vgg16_model.add(Convolution2D(32, (1, 1), activation = 'relu'))
vgg16_model.add(MaxPooling2D(pool_size = (1, 1)))
vgg16_model.add(Flatten())
vgg16_model.add(Dense(units = 256, activation = 'relu'))
vgg16_model.add(Dense(units = y_train.shape[1], activation = 'softmax'))
vgg16_model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])
if os.path.exists("model/vgg16_weights.hdf5") == False:
    model_check_point = ModelCheckpoint(filepath='model/vgg16_weights.hdf5', verbose = 1, save_best_only = True)
    hist = vgg16_model.fit(X, Y, batch_size = 32, epochs = 10,
```

```
validation_data=(X_test, y_test), callbacks=[model_check_point], verbose=1)
    f = open('model/vgg16_history.pckl', 'wb')
    pickle.dump(hist.history, f)
    f.close()
else:
    vgg16_model = load_model("model/vgg16_weights.hdf5")
#perform prediction on test images and then calculate accuracy and other metrics
predict = vgg16_model.predict(X_test)
predict = np.argmax(predict, axis=1)
y_test1 = np.argmax(y_test, axis=1)
calculateMetrics("VGG16", predict, y_test1)#call function to calculate accuracy and other metrics
#training tensorflow, keras cnn proposed model
cnn_model = Sequential()
cnn_model.add(InputLayer(input_shape=(X_train.shape[1], X_train.shape[2], X_train.shape[3])))
cnn_model.add(Conv2D(64, (5, 5), activation='relu', strides=(1, 1), padding='same'))
cnn_model.add(MaxPool2D(pool_size=(2, 2), padding='same'))
cnn_model.add(Conv2D(50, (5, 5), activation='relu', strides=(2, 2), padding='same'))
cnn_model.add(MaxPool2D(pool_size=(2, 2), padding='same'))
cnn_model.add(BatchNormalization())
cnn_model.add(Conv2D(70, (3, 3), activation='relu', strides=(2, 2), padding='same'))
cnn_model.add(MaxPool2D(pool_size=(2, 2), padding='valid'))
cnn_model.add(BatchNormalization())
cnn_model.add(Flatten())
cnn_model.add(Dense(units=100, activation='relu'))
cnn_model.add(Dense(units=100, activation='relu'))
cnn_model.add(Dropout(0.2))
cnn_model.add(Dense(units=y_train.shape[1], activation='softmax'))
```

```
cnn_model.compile(loss='categorical_crossentropy', optimizer="adam", metrics=['accuracy'])
if os.path.exists("model/cnn_weights.hdf5") == False:
    model_check_point = ModelCheckpoint(filepath='model/cnn_weights.hdf5', verbose = 1, save_best_only = True)
    hist = cnn_model.fit(X_train, y_train, batch_size = 32, epochs = 10, validation_data=(X_test, y_test), callbacks=[model_check_point], verbose=1)
    f = open('model/cnn_history.pckl', 'wb')
    pickle.dump(hist.history, f)
    f.close()
else:
    cnn_model.load_weights("model/cnn_weights.hdf5")
#perform prediction on test images and then calculate accuracy and other metrics
predict = cnn_model.predict(X_test)
predict = np.argmax(predict, axis=1)
y_test1 = np.argmax(y_test, axis=1)
calculateMetrics("Proposed CNN Model", predict, y_test1)#call function to calculate accuracy and other metrics
#classification report for each class
from sklearn.metrics import classification_report
print(classification_report(y_test1, predict, target_names=labels))
#comparison graph between all algorithms
import pandas as pd
df = pd.DataFrame([['VGG16','Precision',precision[0]],['VGG16','Recall',recall[0]],['VGG16','F1 Score',fscore[0]],['VGG16','Accuracy',accuracy[0]],
                   ['Proposed CNN','Precision',precision[1]],['Proposed CNN','Recall',recall[1]],['Proposed CNN','F1 Score',fscore[1]],['Proposed CNN','Accuracy',accuracy[1]],
                  ],columns=['Algorithms','Metrics','Value'])
df.pivot_table(index="Algorithms", columns="Metrics",
```

```
values="Value").plot(kind='bar', figsize=(5, 3))
plt.title("All Algorithms Performance Graph")
plt.show()
#displaying all algorithms performance in tabular format
columns = ["Algorithm Name","Precison","Recall","FScore","Accuracy"]
values = []
algorithm_names = ["VGG16", "Proposed CNN"]
for i in range(len(algorithm_names)):

values.append([algorithm_names[i],precision[i],recall[i],fscore[i],accuracy[i]])
temp = pd.DataFrame(values,columns=columns)
temp
f = open('model/cnn_history.pckl', 'rb')
data = pickle.load(f)
f.close()
cnn_accuracy = data['accuracy']
cnn_loss = data['loss']
f = open('model/vgg16_history.pckl', 'rb')
data = pickle.load(f)
f.close()
vgg_accuracy = data['accuracy']
vgg_loss = data['loss']
plt.figure(figsize=(8,4))
plt.grid(True)
plt.xlabel('EPOCH')
plt.ylabel('Accuracy/Loss Rate')
plt.plot(cnn_accuracy)
plt.plot(cnn_loss)
plt.plot(vgg_accuracy)
plt.plot(vgg_loss)
plt.legend(['Proposed CNN Accuracy', 'Proposed CNN Loss', 'VGG16 Accuracy',
'VGG16 Loss'], loc='upper left')
plt.title('Proposed CNN & VGG16 Training Accuracy & Loss Graph')
```

```
plt.show()
#function to convert image gto 3d format
def cv2_to_nibabel(image):
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    image = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)
    image = cv2.resize(image, (128, 128))
    image = np.array(image)
    image = nib.Nifti1Image(image, affine=np.eye(4))
    return image
def getSegmentation(img_path):
    img = cv2.imread(img_path)
    img = cv2_to_nibabel(img)
    img.to_filename('image.nii')
    img = nib.load('image.nii')
    data = img.get_fdata()
    X = np.empty((1, 128, 128, 2))
    flair = data
    ce = data
    X[0,:,:,0] = flair
    X[0,:,:,1] = ce
    data = unet_model.predict(X/np.max(X), verbose=1)
    core = data[:,:,:,1]
    edema= data[:,:,:,2]
    enhancing = data[:,:,:,3]
    core = core[0]
    edema = edema[0]
    segment= enhancing[0]
    cv2.imwrite("segment.jpg", segment*255)
    return cv2.imread("segment.jpg")
    def classifyTumor(test_image, image):
    img = cv2.imread(test_image)
    img = cv2.resize(img, (32,32))#resize image
    im2arr = np.array(img)
```

```
    im2arr = im2arr.reshape(1,32,32,3)
    img = np.asarray(im2arr)
    img = img.astype('float32')
    img = img/255 #normalizing test image
    predict = cnn_model.predict(img)#now using  extension CNN + GRU to predict wild animals
    predict = np.argmax(predict)
    img = cv2.imread(test_image)
    img = cv2.resize(img, (600,400))
    image = cv2.resize(image, (600,400))
    cv2.putText(img, 'Prediction Output : '+labels[predict]+" Detected", (10, 25), cv2.FONT_HERSHEY_SIMPLEX,0.7, (255, 0, 0), 2)
    cv2.putText(image, '3D-UNET Segmented Image', (10, 25), cv2.FONT_HERSHEY_SIMPLEX,0.7, (255, 0, 0), 2)
    return img, image, labels[predict]
img = getSegmentation("testImages/6.jpg")
classify_img, segment, label = classifyTumor("testImages/6.jpg", img)
plt.figure()
f, axarr = plt.subplots(1,2, figsize=(8,4))
axarr[0].imshow(classify_img, cmap="gray")
axarr[0].title.set_text('Tumor Classification ('+label+")")
axarr[1].imshow(segment, cmap="gray")
axarr[1].title.set_text('Tumor Segmented Image')
plt.show()
img = getSegmentation("testImages/4.jpg")
classify_img, segment, label = classifyTumor("testImages/4.jpg", img)
plt.figure()
f, axarr = plt.subplots(1,2, figsize=(8,4))
axarr[0].imshow(classify_img, cmap="gray")
axarr[0].title.set_text('Tumor Classification ('+label+")")
axarr[1].imshow(segment, cmap="gray")
axarr[1].title.set_text('Tumor Segmented Image')
plt.show()
```

```
img = getSegmentation("testImages/8.jpg")
classify_img, segment, label = classifyTumor("testImages/8.jpg", img)
plt.figure()
f, axarr = plt.subplots(1,2, figsize=(8,4))
axarr[0].imshow(classify_img, cmap="gray")
axarr[0].title.set_text('Tumor Classification ('+label+")")
axarr[1].imshow(segment, cmap="gray")
axarr[1].title.set_text('Tumor Segmented Image')
plt.show()
img = getSegmentation("testImages/1.jpg")
classify_img, segment, label = classifyTumor("testImages/1.jpg", img)
plt.figure()
f, axarr = plt.subplots(1,2, figsize=(8,4))
axarr[0].imshow(classify_img, cmap="gray")
axarr[0].title.set_text('Tumor Classification ('+label+")")
axarr[1].imshow(segment, cmap="gray")
axarr[1].title.set_text('Tumor Segmented Image')
plt.show()
```

# INSTITUTION'S INNOVATION COUNCIL
# MOE'S INNOVATION CELL

**Institute Name:**
**Malla Reddy Institute of Technology & Science**

**Title of the Innovation/Prototype:**
**Segmentation And Classification Of Brain Tumor Using 3D-UNET Deep Neural Networks**

| Team Lead Name: | Team Lead Email: | Team Lead Phone: | Team Lead Gender: |
|---|---|---|---|
| Rekha | mangarekha146@gmail.com | 9989481217 | Female |

**Website (if any):**
**null**

**Startup/Venture Registered as:**
**Not Yet Registered as an entity**

**Does your Startup/Venture Recognized by DPIIT, Startup India?:**
**No**

**Name a Key Innovation which is Core to the Startup /Venture:**
**Segmentation And Classification Of Brain Tumor Using 3D-UNET Deep Neural Networks**

| Year of Started Receiving Pre-incubation/IncubationSupport for the Development of Innovation-Startup from the Institute (FY): | The Key Innovation which is Core to your Startup /Venture was Developed as: | Choose the Type of Innovation: | TRL LEVEL: |
|---|---|---|---|
| 2024-25 | Academic Requirement/Study Project | Product,Service | 6 |

**The Sector/Domain of Focus of the Innovation/Startup / Venture:**
**Healthcare & Biomedical devices.,Education,**

**Define the problem and its relevance to today's market / sociaty / industry need:**
**Startups leveraging 3D U-Net deep networks for brain tumor segmentation and classification are revolutionizing medical imaging with innovative AI-driven solutions. These models provide highly accurate and automated tumor detection, reducing the need for manual segmentation and enabling faster, more more precise diagnoses. By analy zing MRI or CT scans in three dimensions, they help in early tumor detec tion and classification, leading to personalized treatment plans. Their integration into hosp ital systems enhances efficiency and accessibility, particularly in remote areas. Additionally, continuous learning improves accuracy over time, making AI-powered diagnosis a cost-effective, scala ble, and reliable solution for modern healthcare.**

**Explain the uniqueness and distinctive features of the (product / process / service) solution:**

**How your proposed / developed (product / process / service) solution is different from similiar kind of product by the competitors if any:**

**Is there any IP or Patentable Component associated with the Solution?:**
**No**

**Copy of IP/Patent Applied or Obtained:** View File

**Did the venture/startup receive any innovation grant from the Institute?**
**Yes**

**Mention the Pre-Incubation / Incubation Unit Name: MRIT**

**Total Grant Amount Received (Rs.) in FY 2019-20: 0**

**Total Grant Amount Received (Rs.) in FY 2020-21: 0**

**Total Grant Amount Received (Rs.) in FY 2021-22: 0**

**Did the venture/startup receive any innovation grant from any external sources, so far?**
**No**

**Did the venture/startup raise any Angel/Venture Capital Investment so far?**
**No**

**Are there any recognitions/awards received by the venture/startup for the innovation in National/International Competitions?:**
**No**

**Upload the Audited copy of the financial Statement clearly indicating the FY and Annual turnover amount of Rs. 50 Lakhs or above:**
**No**

**Define the Problem – Solution fit achieved/to be achieved by the Startup: Briefly explain the relevance of the innovative solutions are being offered by the startup and what/whose problem (Industry/Society/Market) these are solving:**

**Startups leveraging 3D U-Net deep networks for brain tumor segmentation and classification are revolutionizing medical imaging with innovative AI-driven solutions. These models provide highly accurate and automated tumor detection, reducing the need for manual segmentation and enabling faster, more more precise diagnoses. By analy zing MRI or CT scans in three dimensions, they help in early tumor detec tion and classification, leading to personalized treatment plans. Their integration into hosp ital systems enhances efficiency and accessibility, particularly in remote areas. Additionally, continuous learning improves accuracy over time, making AI-powered diagnosis a cost-effective, scala ble, and reliable solution for modern healthcare.**

**Define the Product-Market fit achieved/ to be achieved by the Startup: Briefly explain the readiness levels (Technology Readiness Level and Manufacturing Readiness Level) of innovations/solutions offered by the startup to meet the customer need/requirement.**

**The readiness levels of innovation for brain tumor segmentation and classification using 3D U-Net deep networks progress through multiple stages. Initially, research focuses on algorithm development and validation using medical imaging datasets. As models mature, they und ergo clinical testing to assess accuracy, reliability, and real-world applicability. Regula tory approvals ensure compliance with healthcare standards, mov ing the technology toward integration into hospitals and diagnostic centers. Wider adoption depends on scalability, and cost-effectiveness, and real-world performance. With ongo ing advancements in AI and medical imaging, these solutions are transitioning from experimental research to practical deployment, aiming for widespread clinical acceptance.**

**Detail the potential market size and target customers/segment (Total Available Market -TAM, Serviceable Available Market - SAM, Serviceable Obtainable Market - SOM):**

**The potential market size for brain tumor segmentation and classification using 3D U-Net deep networks is significant, driven by the rising global burden of brain tumors and advancements in AI-powered diagnostics. The market includes hospitals, diagnostic centers, research institutions, and AI-driven healthcare startups. Key target custo mers are radiologists, neurologists, oncologists, and healthcare providers seeking accurate, automated tumor analysis. With increasing adoption of AI in medical imaging, the global AI healthcare market, valued in billions, is expected to grow rapidly. The demand for early detection, personalized treatment, and cost-effective solutions positions this technology as a game-changer in neuroscience and oncology.**

**Detail the Business fit achieved/ to be achieved by the Startup: Briefly explain the business model readiness level of innovations to be commercialized. Business Tractions Achieved for the innovation if any, briefly explain the customer tractions achieved for the innovations or solutions offered by the Startup as an attempt to commercialization:**

**A startup developing brain tumor segmentation and classification using 3D U-Net deep networks must achieve a strong business fit by aligning innovation with healthcare needs. This includes ensuring high diagnostic accuracy, and regulatory compliance, and seamless integration with existing medical imaging systems. Partnerships with hospitals, diagnostic centers, and AI healthcare firms will drive adoption. A scalable, cost-effective, and user-friendly solution enhances market penetration. Demonstrating clinical reliability and securing approvals (FDA, CE) and will build trust. A subscription-based or per-scan revenue model can ensure sustainability. Ultimately, success depends on delivering real-world impact by improving tumor detection, patient outcomes, and workflow efficiency.**

**Highlight any competitive advantages such as Intellectual property (IP) or any Unique Selling Proposition (USP) etc. associate with the product/service/business model/startup:**
**The competitive advantages of a startup using 3D U-Net deep networks for brain tumor segmentation and classification lie in its intellectual property (IP), and proprietary algorithms, and AI-driven automation. A unique deep learning model trained on diverse medical datasets ensures higher accuracy and reliability than traditional methods. Integration with cloud-based platforms enables remote diagnostics, mak ing it accessible globally. Regulatory approvals and (FDA, CE) and clinical validation build trust among healthcare providers. The unique selling proposition (USP) includes faster diagnosis, reduced radiologist workload, and personalized treatment insights. These innovations position of the startup ahead in the AI-driven medical imaging market.**

**Video URL:**
**https://docs.google.com/presentation/d/1AbSZNxPU17iFNdU53Bxef0lFRp-3kIzo/edit?usp=sharing&ouid=105279094658403936023&rtpof=true&sd=true**

**Innovation Photograph:**
**View File**