BRAIN TUMOR DETECTION
USING DEEP LEARNING

A Project Report submitted in the partial fulfillment of the requirements for the award of the degree

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by

M.Venkata Sai Pavan Kumar (20471A0596)

S.Gopi Krishna (20471A05B0)

Under the esteemed guidance of

N.Vijaya Kumar, M.E.

Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE : NARASARAOPET

(AUTONOMOUS)

Accredited by NAAC with A+ grade and NBA under cycle-1

NIRF rank in the brand of 251-320 and an ISO 9001:2015 Certified

Approved by AICTE, New Delhi, Permanently Affiliated to J.N.T.U,Kakinada

KOTAPPAKONDA ROAD, YALAMANDA VILLAGE,NARASARAOPET-522601

2023-2024

NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPET
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the main project entitled "BRAIN TUMOR DETECTION USING DEEP LEARNING" is a bonafide work done by M.Venkata Sai Pavan Kumar (20471A0596),S.Gopi Krishna (20471A05B0) in partial fulfillment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in the Department of COMPUTER SCIENCE AND ENGINEERING during 2023-2024.

PROJECT GUIDE

Vijaya Kumar Nukala, M.E.
Associate Professor

PROJECT CO-ORDINATOR

Dr. Sireesha Moturi, M.Tech., Ph.D.
Associate Professor

HEAD OF THE DEPARTMENT

Dr. S. N. TirumalaRao, M.Tech.,Ph.D
Professor

EXTERNAL EXAMINER

DECLARATION

We declare that this project work titled " BRAIN TUMOUR DETECTION USING DEEP LEARNING " is composed by ourselves that the work contain here is our own except where explicitly stated otherwise in the text and that this work has been submitted for any other degree or professional qualification except as specified.

M.Venkata Sai Pavan Kumar (20471A0596)

S.Gopi Krishna (20471A05B0)

ACKNOWLEDGEMENT

We wish to express our thanks to carious personalities who are responsible for the completion of the project. We are extremely thankful to our beloved chairman sir **M.V. Koteswara Rao, B.sc** who took keen interest in us in every effort throughout this course.We owe our gratitude to our principal sir **M. Sreenivasa Kumar, M.Tech.,Ph.D(U.K), MISTE.,FIE(I),** for his kind attention and valuable guidance throughout the course.

We express our deep felt gratitude to **Dr. S. N. Tirumala Rao, M.Tech., Ph.D** Professor and H.O.D. of CSE department and our guide **N.Vijaya Kumar, M.E.** Associate Professor CSE department whose valuable guidance and unstinting encouragement enable me to accomplish my project successfully in time.

We extend our sincere thanks to **Dr.M.Sireesha,M.Tech., Ph.D** Associate Professor and coordinator of the project for extending her encouragement. Their profound knowledge and willingness have been a constant source of inspiration for me through out this project work.

We extend our sincere thanks to all other teaching and non-teaching staff to department for their cooperation and encouragement during my B. Tech degree. We have no words to acknowledge the warm affection, constant inspiration and encouragement that we receive from my parents.

We affectionately acknowledge the encouragement received from my friends and those who involved in giving valuable suggestions had clarifying our doubts which had reallyhelped me in successfully completing our project.

By

M.Venkata Sai Pavan Kumar (20471A0596)

S.Gopi Krishna (20471A05B0)

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community,

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering andresearch

M2: Build a passionate and a determined team of faculty with student centric teaching, imbibing experiential, innovative skills

M3: Imbibe lifelong learning skills, entrepreneurial skills and ethical values in students for addressing societal problems

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE ANDENGINEERING

VISION OF THE DEPARTMENT

To become a centre of excellence in nurturing the quality Computer Science & Engineering professionals embedded with software knowledge, aptitude for researchand ethical values to cater to the needs of industry and society.

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

M1: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M2: Impart high quality professional training to get expertize in modern softwaretools and technologies to cater to the real time requirements of the Industry.

M3: Inculcate team work and lifelong learning among students with a sense ofsocietal and ethical responsibilities.

NARASARAOPETA
NEC ENGINEERING COLLEGE
(AUTONOMOUS)

Program Specific Outcomes (PSO's)

PSO1: Apply mathematical and scientific skills in numerous areas of ComputerScience and Engineering to design and develop software-based systems.

PSO2: Acquaint module knowledge on emerging trends of the modern era inComputer Science and Engineering

PSO3: Promote novel applications that meet the needs of entrepreneur, environmental and social issues.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Educational Objectives (PEO's)

The graduates of the programme are able to:

PEO1: Apply the knowledge of Mathematics, Science and Engineering fundamentals to identify and solve Computer Science and Engineering problems.

PEO2: Use various software tools and technologies to solve problems relatedto academia, industry and society.

PEO3: Work with ethical and moral values in the multi-disciplinary teams and can communicate effectively among team members with continuous learning.

PEO4: Pursue higher studies and develop their career in software industry.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Outcomes

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receiveclear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Project Course Outcomes (CO'S):

CO421.1: Analyse the System of Examinations and identify the problem.

CO421.2: Identify and classify the requirements.

CO421.3: Review the Related Literature

CO421.4: Design and Modularize the project

CO421.5: Construct, Integrate, Test and Implement the Project.

CO421.6: Prepare the project Documentation and present the Report using appropriate method.

	PO1	PO 2	PO 3	PO 4	PO 5	PO 6	PO7	PO8	PO 9	PO 10	PO 11	PO 12	PSO1	PSO2	PSO3
C421.1		√											√		
C421.2	√		√		√								√		
C421.3				√		√	√	√					√		
C421.4			√			√	√	√					√	√	
C421.5					√	√	√	√	√	√	√	√	√	√	√
C421.6									√	√	√		√	√	

Course Outcomes – Program Outcome correlation

	PO1	PO2	PO3	PO 4	PO 5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
C421.1	2	3											2		
C421.2			2		3								2		
C421.3				2		2	3	3					2		
C421.4			2			1	1	2					3	2	
C421.5					3	3	3	2	3	2	2	1	3	2	1
C421.6									3	2	1		2	3	

Note: The values in the above table represent the level of correlation between CO's and PO's:

1. **Low level**

2. **Medium level**

3. **High level**

Project mapping with various courses of Curriculum with Attained PO's:

Name of the course from which principles are applied in this project	Description of the device	Attained PO
C2204.2,C22L3.2	Gathering the requirements and defining the problem, plan to develop a **Brain Tumor Detection using Deep Learning**	PO1, PO3
CC421.1,C2204.3,C22L3.2	Each and every requirement is critically analyzed, the process model is identified and divided into **five modules**	PO2, PO3
CC421.2,C2204.2,C22L3.3	Logical design is done by using the unified modelling language which involves individual team work	PO3, PO5, PO9
CC421.3,C2204.3,C22L3.2	Each and every module is tested, integrated, and evaluated in our project	PO1, PO5
CC421.5,C2204.2,C22L3.3	Documentation is done by all our four members in the form of a group	PO10
CC421.5,C2204.2,C22L3.3	Each and every phase of the work in group is presented periodically	PO10, PO11
C2202.2,C2203.3,C1206.3, C3204.3,C4110.2	Implementation is done and the project will be handled by the **users using this project to detect the brain tumors through MRI scans.**	PO4, PO7
C32SC4.3	**The physical design includes the website to classify the MRI scanned images whether it is tumor or not.**	PO5, PO6

ABSTRACT

The accurate and timely detection of brain tumors plays a crucial role in the effective management of neurological disorders. This abstract presents a comprehensive overview of a novel approach for brain tumor detection leveraging deep learning techniques. Our proposed method utilizes convolutional neural networks (CNNs) and recurrent neural networks (VGG16s) to analyze medical imaging data, specifically magnetic resonance imaging (MRI) scans. The workflow involves pre processing of MRI images to enhance features, followed by a CNN based feature extraction stage that learns hierarchical representations. The extracted features are then fed into an VGG16 to capture temporal dependencies and spatial relationships within the data. This hybrid deep learning architecture enhances the model's ability to discern subtle patterns indicative of brain tumors, improving both sensitivity and specificity. To facilitate model training and evaluation, a curated dataset of diverse brain tumor cases is utilized. The model demonstrates promising results in terms of accuracy, sensitivity, and specificity during rigorous validation and testing phases. The proposed approach not only outperforms traditional methods but also exhibits robustness against variations in imaging conditions and tumor characteristics. In conclusion, our deep learning- based approach showcases the potential for significantly improving the accuracy and efficiency of brain tumor detection in clinical settings. The integration of CNNs and VGG16s in a unified framework demonstrates the synergy of spatial and temporal information, making it a promising avenue for enhancing the capabilities of automated brain tumor diagnosis.

INDEX

LIST OF FIGURES

1. Introduction

1.1 Introduction

The human body is made up of many organs and brain is the most critical and vital organ of them all. One of the common reasons for dysfunction of brain is brain tumor. A tumor is nothing but excess cells growing in an uncontrolled manner. Brain tumor cells growin a way that they eventually take up all the nutrients meant for the healthy cells and tissues,which results in brain failure. Currently, doctors locate the position and the area of brain tumor by looking at the MR Images of the brain of the patient manually. This results in inaccurate detection of the tumor and is considered very time consuming. A Brain Cancer isvery critical disease which causes deaths of many individuals. The brain tumor detection andclassification system is available so that it can be diagnosed at early stages. Cancer classification is the most challenging tasks in clinical diagnosis. This project deals with sucha system, which uses computer, based procedures to detect tumor blocks and classify the typeof tumor using Convolution Neural Network Algorithm for MRI images of different patients.Different types of image processing techniques like image segmentation, image enhancementand feature extraction are used for the brain tumor detection in the MRI images of the cancer-affected patients. Detecting Brain tumor using Image Processing techniques its involves the four stages is Image Pre-Processing, Image segmentation, Feature Extraction, and Classification. Image processing and neural network techniques are used for improve the performance of detecting and classifying brain tumor in MRI images.

1.2 Existing System

Brain Tumor is a cancerous or non-cancerous mass or growth of abnormal cells in the brain. Tumors can start in the brain, or cancer elsewhere in the body can spread to the brain.Imaging tests can help doctors find out if the tumor is a primary brain tumor or if it is cancer that has spread to the brain from elsewhere in the body. Imaging tests show pictures of the inside of the body. Your doctor may consider these factors when choosing a diagnostic test.

- The type of tumor suspected

- Your signs and symptoms

- Your age and general health

In general, diagnosing a brain tumor usually begins with magnetic resonance imaging(MRI). Once MRI shows that there is a tumor in the brain, the most common way to determine the type of brain tumor is to look at the results from a sample of tissue after a biopsy or surgery. After diagnostic tests are done, doctor will review the test results with you. If the diagnosis is a brain tumor, additional tests will be done to learn more about the tumor. The resultshelp the doctor describe the tumor and plan your treatment. Hospitals maintain all the patient records. Even though, those records are not used in an efficient manner for diagnosis. To maintain the records in an efficient error free manner, the new proposed system is introduced.

Disadvantages:

1. Doesn't generate accurate and efficient results
2. Computation time is very high
3. Difficulty in maintenance of patient records
4. Lacking of accuracy may result in lack of efficient further treatment.

1.3 Proposed System

We proposed to develop a system which will help practitioners to predict brain tumor based on images of mammography test. So, there is a need for developing a decision system which will help practitioners to predict brain tumor in an easy way, which can offer prediction about the tumor condition of the patient so that further treatment can be made effectively.

This proposed system not only accurately predicts brain tumors but also reduces time for prediction. Deep learning algorithms like CNN, VGG19 are used to develop this system.

The proposed system has mainly five modules. Dataset, Pre-processing, Split the data, Build CNN model train, Deep Neural network for epochs, and classification. In the dataset we can take multiple MRI images and take one as input image. In pre-processing image to encoded the label and resize the image. In split the data we set the image as 80% Training Data and 20% Testing Data. Then build CNN model train deep neural network for epochs. Then classified the image as yes or no if tumor is positive then it returns yes and the tumor is negative the it returns no.

Advantages:

1. Generates accurate and efficient results
2. Computation time is greatly reduced
3. Reduces manual work
4. Automated prediction

1.4 Deep Learning

Deep learning is an artificial intelligence(AI) function that imitates the working of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence that has networks capable oflearning unsupervised from data that is unstructured or unlabeled. Also known as deep neurallearning or deep neural network. **Deep Learning** is a subfield of machine learning concernedwith algorithms inspired by the structure and function of the brain called **artificial neural networks**.

1.5 Deep Learning Methods

Deep learning uses artificial neural networks to perform sophisticated computations on large amounts of data. It is a type of machine learning that works based on the structure and function of the human brain. Deep learning algorithms train machines by learning from examples. Industries such as health care, ecommerce, entertainment, and advertising commonly use deep learning.. Few examples are: LSTM,VGG16,CNN etc. Deep learning has evolved over the past five years, and deep learning algorithms have become widely popular in many industries. Deep learning algorithms train machines by learning from examples. Industries such as health care, eCommerce, entertainment, and advertising commonly use deep learning.

While deep learning algorithms feature self-learning representations, they depend upon ANNs that mirror the way the brain computes information. During the trainingprocess, algorithms use unknown elements in the input distribution to extract features, group objects,and discover useful data patterns. Much like training machines for self- learning, this occursat multiple levels, using the algorithms to build the models.

Deep learning models make use of several algorithms. While no one network is considered perfect, some algorithms are better suited to perform specific tasks. To choose the right ones, it's good to gain a solid understanding of all primary algorithms.

Defining Neural Networks

A neural network is structured like the human brain and consists of the artificial neurons, also known as nodes. These nodes are stacked next to each other in three layers:

- The input layer
- The hidden layer(s)
- The output layer

Data provides each node with information in the form of inputs.The node multiplies the inputs with random weights, calculates them, and adds a bias. Finally, the non linear functions, also known as activation functions, are applied to determine which neuron to fire.**LSTM:**

Long Short Term Memory Networks are a type of Recurrent Neural Network(VGG16) that can learn and memorize long-term dependencies. Recalling past information for long periods is the default behavior.

LSTMs retain information over time. They are useful in time-series prediction becausethey remember previous inputs. LSTMs have a chain-like structure where four interacting layers communicate in a unique way. Besides time-series predictions, LSTMs are typically used for speech recognition, music composition, and pharmaceutical development.

How Do LSTMs Work?

- First, they forget irrelevant parts of the previous state
- Next, they selectively update the cell-state values
- Finally, the output of certain parts of the cell state

Autoencoders:

Auto encoders are a specific type of feed forward neural network in which the input and output are identical. Geoffrey Hinton designed autoencoders in the 1980s to solve unsupervised learning problems. They are trained neural networks that replicate the data from the input layer to the output layer. Autoencoders are used for purposes such as pharmaceutical discovery, popularity prediction, and image processing.

How Do Autoencoders Work?

An auto encoder consists of three main components: the encoder, the code, and the decoder.

- Autoencoders are structured to receive input and transform it into a different representation. They then attempt to reconstruct the original input as accurately as possible.
- When an image of a digit is not clearly visible, it feeds to an autoencoder neural network.
- Autoencoders first encode the image, then reduce the size of the input into a smaller representation.
- Finally, the autoencoder decodes the image to generate the reconstructed image.

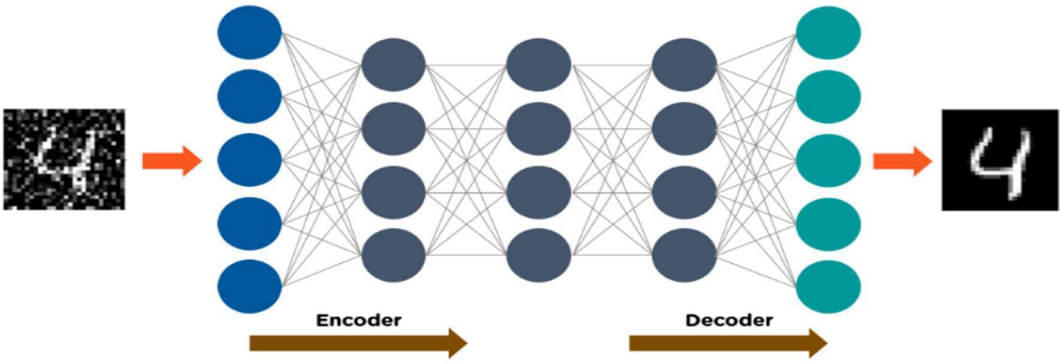

Fig:1.1 Autoencoders

RBMs:

RBMs are stochastic neural networks that can learn from a probability distribution over a set of inputs.

RBMs consist of two layers:

- Visible units

Each visible unit is connected to all hidden units. RBMs have a bias unit that isconnected to all the visible units and the hidden units, and they have no output nodes.

How Do RBMs Work?

RBMs have two phases: forward pass and backward pass.

- RBMs accept the inputs and translate them into a set of numbers that encodes the inputsin the forward pass.
- RBMs combine every input with individual weight and one overall bias.The algorithmpasses the output to the hidden layer.
- In the backward pass , RBMs take that set of numbers and translate them to formThe reconstructed inputs.
- RBMs combine each activation with individual weight and overall bias and pass theoutput to the visible layer for reconstruction.
- At the visible layer,the RBM compares the reconstruction with the original input toAnalyze the quality of the result.

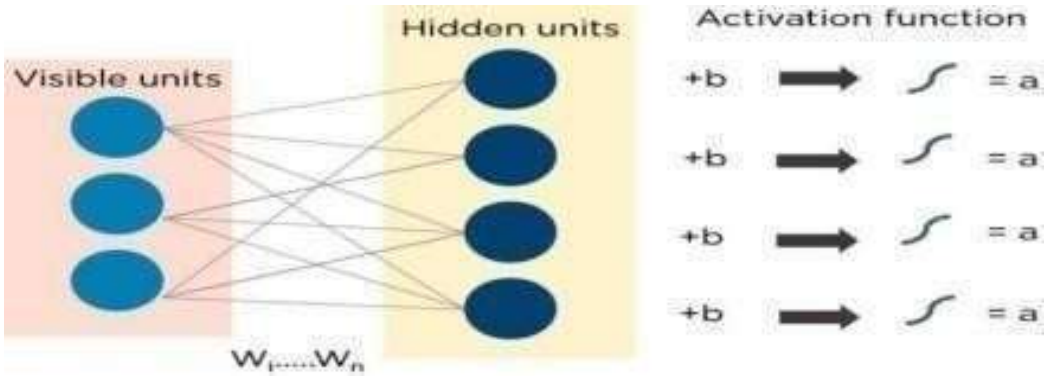

Fig:1.2 RBM

Recurrent Neural Networks (VGG16s):

VGG16s have connections that form directed cycles, which allow the outputs from the LSTM to be fed as inputs to the current phase.

The output from the LSTM becomes an input to the current phase and can memorize previous inputs due to its internal memory. VGG16s are commonly used for image captioning, time-series analysis, natural-language processing, handwriting recognition, and machine translation.

How Do VGG16s work?

- The output at time t-1 feeds into the input at time t.
- Similarly, the output at time t feeds into the input at time t+1.
- VGG16s can process inputs of any length.
- The computation accounts for historical information, and the model size does not increasewith the input size.

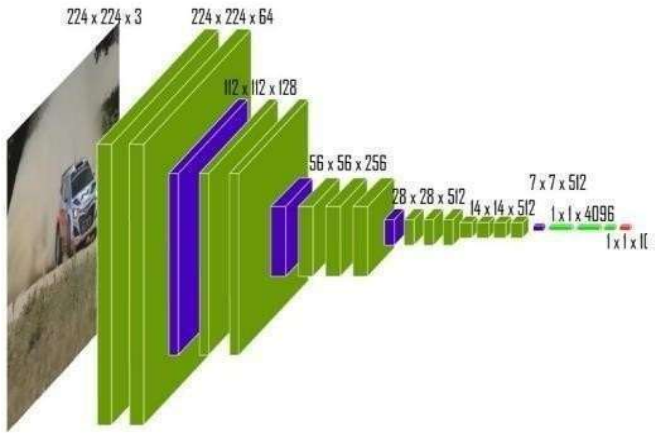

Fig:1.3 VGG16

CNN:

A **Convolutional Neural Network (ConvNet/CNN)** is a Deep Learning algorithm which can take in an input image, assign importance to various aspects / objects in the image and be able to differentiate one from the other. Sequential model in CNN allows to build a model layer by layer. There are three types of layers in a convolutional neural network: convolutional layer, pooling layer, and fully connected layer.

Convolutional layer:

Convolutional layer is the first layer that is used to extract the various features from theinput images. This layer separates and identifies the various features of the image for analysis in aprocess called as **Feature Extraction**. Various parameters such as filter, kernel size,activation,padding and input shape are used. The Activation function used in this layer Rectified Linear Unit(ReLU) and it returns 0 if it receives zero input but for any positive valuex it returns the x value.

Rectified Linear Unit (ReLU):

• CNN's have a ReLU layer to perform operations on elements. The output is a rectified featurefeature map.

Pooling Layer

• The rectified feature map next feeds into a pooling layer . Pooling is a down sampling operation that reduces the dimensions of the feature map.

• The pooling layer then converts the resulting two - dimensional arrays from the Pooledfeature map into a single , long , continuous , linear vector by flattening it.

Fully Connected Layer

•A fully connected layer forms when the flattened matrix from the pooling layer is fed as aninput, which classifies and identifies the images.

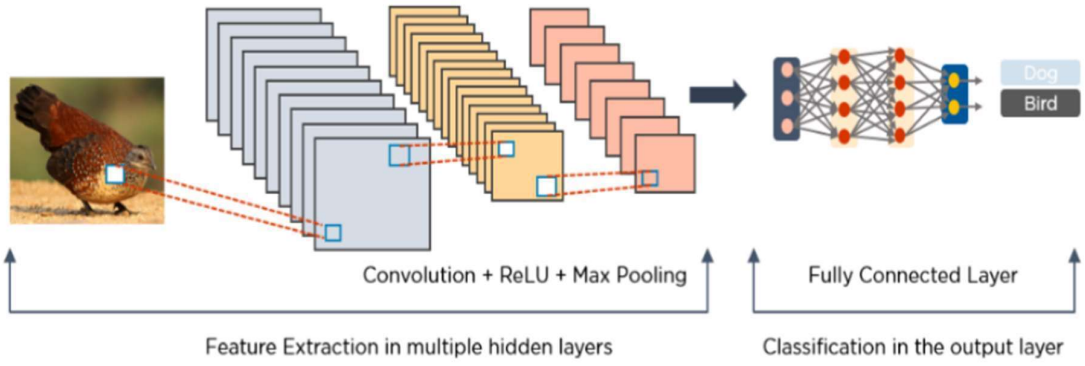

Fig:1.4 CNN

1.6 Applications of Deep Learning:

1. Self-Driving Cars
2. Visual Recognition
3. Fraud Detection
4. Healthcare
5. Personalisations
6. Detecting Developmental Delay in Children
7. Colorization of Black and White Images
8. Adding Sounds to Silent Movies
9. Facial Expression Recognition
10. Breast cancer prediction

1.7 Importance of Deep Learning:

Deep Learning is a branch of artificial intelligence that uses data to enable machinesto learn to perform tasks on their own. This technology is already live and used in automatic email reply predictions, virtual assistants, tumor recognition systems[17], and self driving cars. Break throughs in this technology are also making an impact in the health care sector.Using these types of advanced analytics, we can provide better information to health care.The ability to process large numbers of features makes deep learning very powerful When dealing with unstructured data. However, deep learning algorithms can be overkill for less complex problems because they require access to a vast amount of data to be effective.If the data is too simple or incomplete, it is very easy for a deep learning model to become overfitted and fail to generalize well to new data. As a result, deep learning models are not as effective as other techniques (such as boosted decision trees or linear models) for most practical business problems such as understanding customer churn, detecting fraudulent transactions and other cases with smaller datasets and fewer features. In certain cases like multiclass classification, deep learning can work for smaller, structured datasets.

1.8 Implementation of Deep Learning using Python

Python is a popular programming language. It was created in 1991 by Guido van Rossum. It is used for:

- Web development (server-side)
- software development,
- mathematics and
- System scripting.

The most recent major version of Python is Python 3.However, Python 2,although not not being updated with anything other than security updates, is still quite popular.

It is possible to write Python in an Integrated Development Environment, such as Thonny, PyCharm, NetBeans or Eclipse, Anaconda which are particularly useful when managinglarger collections of Python files.

Python was designed for its readability. Python uses new lines to complete a command,as opposed to other programming languages which often use semicolons or parentheses.

Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

In the older days, people used to perform Deep Learning tasks manually by coding all the algorithms and mathematical and statistical formula. This made the process time consuming, tedious and inefficient. But in the modern days, it is become very much easy and efficient compared to the olden days by various python libraries, frameworks, and modules. Today, Python is one of the most popular programming languages for this task and it has replaced manylanguages in the industry, one of the reason is its vast collection of libraries.

Python libraries that used in Deep Learning are:

1. Pandas
2. Numpy
3. Matplotlib
4. Tensor flow
5. Keras

Pandas: It is a popular Python library for data analysis. It is not directly related to Deep Learning.As we know, the dataset must be prepared before training. In this case, Pandas comeshandy as itwas developed specifically for data extraction and preparation. It provides high- level data structures and a wide variety of tools for data analysis. It provides many inbuilt methods for groping, combining and filtering data.

NumPy: It is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical functions. It is very useful for fundamental scientific computations in Deep Learning. It is particularly useful for linear algebra, Fourier transform, and random number capabilities. High-end libraries like TensorFlow uses NumPy internally for manipulation of Tensors.

Matplotlib: It is a very popular Python library for data visualization. Like Pandas, it is not directly related to Deep Learning. It particularly comes in handy when a programmer wants to visualize the patterns in the data. It is a 2D plotting library used for creating 2D graphs and plots[9]. A module named pyplot makes it easy for programmers for plotting as it provides features to control line styles, font properties, formatting axes, etc. It provides various kinds of graphs and plots for data visualization, histogram, error charts, bar chats, etc.

TensorFlow: It is an open-source library developed by Google primarily for deep learning applications[12]. It also supports traditional machine learning. TensorFlow was originally developed for large numerical computations without keeping deep learning in mind.

Keras : It is a powerful and easy-to-use free open source Python library for developing and evaluating deep learning models[13]. It wraps the efficient numerical computation libraries Theano and TensorFlow and allows to define and train neural network models in just a few linesof code.

1.9 Deep Learning Products

1. Yuehao Pan, Weimin Huang et.al "Brain Tumor Grading based on Neural networks and theConvolutional Neural Networks",IEEE.
2. Darko Zikic, Yani Ioannou et.al " Segmentation of Brain Tumor tissues with convolutional neural networks", MiCCAI,2014,Brats challenge.
3. Karen, Zisserman "Very deep convolutional networks for large scale image recognition",ICLR.[Published in 2014].
4.

2. Literature Survey

Grampurohit et al. (2020) introduced a brain tumor detection system leveraging CNN and VGG-16 models applied to MRI images. While CNN achieved an accuracy of 93%, VGG-16 demonstrated superior performance with 97.16% training and 97.42% validation accuracy. However, VGG-16 necessitates higher computational resources and time compared to CNN. Limitations of this approach include the demand for extensive data and computational power. Further exploration of optimization techniques is warranted to mitigate these challenges and enhance overall performance. Possible avenues for improvement include refining model architectures, exploring lightweight alternatives to VGG-16, and implementing advanced optimization strategies to reduce computational overhead. By addressing these limitations, the efficiency and scalability of brain tumor detection systems based on deep learning can be significantly enhanced, potentially facilitating more accessible and accurate diagnostic tools for medical practitioners [1].

Saleh et al. (2020) introduced a pioneering approach for brain tumor classification using deep learning and convolutional neural networks (CNN). They trained five pre-trained CNN models – Xception, ResNet50, InceptionV3, VGG16, and MobileNet – on a dataset comprising 4480 MRI images. Through rigorous validation, these models demonstrated impressive F1-scores ranging from 97.25% to 98.75%. Particularly, the Xception model outperformed others with an accuracy of 98.75%. This groundbreaking research aims to optimizeMRI machine efficiency in tumor classification, potentially facilitating early detection and minimizing physical side effects. However, it's essential to acknowledge certain limitations, suchas imbalanced data distribution and the reliance on pre-trained models, which may impede generalization across diverse datasets. Addressing these limitations through further research could enhance the applicability and robustness of deep learning-based tumor classification systems in clinical settings [2].

G.N.V. Pushpalatha et al. (2023) presented a system for brain tumor identification and categorization using deep learning, showcased at the 2023 Winter Summit onSmart Computing and Networks. The study highlighted the effectiveness of deep learning techniques in accurately identifying and categorizing brain tumors. However, potential limitations include the necessity for extensive and diverse datasets to ensure model robustness, as well as significant computational resources for training complex deep learning architectures.

Further research is needed to validate the approach's performance across diverse patient cohortsand clinical environments, thereby enhancing its practical utility in real-world applications. [3].Sravya et al. (2021) conducted a survey on brain tumor detection methods using machine learning and deep learning approaches, presented at the 2021 International Conference on Computer Communication and Informatics. The study highlighted various techniques, emphasizing both machine learning and deep learning methodologies. However, limitations include the lack of standardized datasets and benchmarks for fair comparisons among different techniques. Additionally, challenges such as the interpretability of deep learning models and theirscalability to large datasets remain areas requiring further investigation and enhancement in brain tumor detection research.[4]

Sinha et al. (2021) propose a deep learning-based method for brain tumor detection from MRI images, employing convolutional neural network (CNN) classification, segmentation, feature extraction, and picture preprocessing. The study attains 98% accuracy on test data, surpassing prior methods. Limitations include dataset size constraints and segmentation challenges in distinguishing tumor regions from healthy tissue. Future work may focus on enhancing the system's interface, expanding disease detection capabilities, and refining density estimation techniques. Addressing these aspects could further improve the efficacy and applicability of the proposed method in clinical settings, potentially advancing early diagnosis and treatment of brain tumors.[6]

Hemanth et al. (2019) propose a brain tumor detection system utilizing machine learning techniques and convolutional neural networks (CNNs) for both segmentation and classification. Their methodology involves data collection, pre-processing, average filtering, segmentation, feature extraction, and CNN-based classification. Despite promising results, the study lacks thorough exploration of dataset diversity and evaluation metrics. It also neglects potential challenges like data imbalance and generalizability across diverse patient demographics. Further research is necessary to enhance performance robustness and ensure applicability across variousclinical scenarios.[9]

Sankara Narayanan et al. present a methodology employing deep learning to improve glioma brain tumor identification from MRI scans. Their approach includes data preprocessing, feature extraction, and classification using deep neural networks (DNNs). While yielding promising results, the study lacks exploration of alternative deep learning architectures and comprehensive comparison with traditional machine learning methods. Moreover, the generalization of the

proposed technique across diverse MRI datasets and its performance in real-world clinical settings necessitate further investigation. Addressing these limitations in future research can enhance the effectiveness and applicability of the proposed method, potentially advancing the field of brain tumor identification and diagnosis.[12]

Prakram et al. (2023) introduce a brain tumor detection system employing deep learning andimage classification techniques to enhance accuracy. Their methodology encompasses preprocessing, feature extraction, and classification utilizing convolutional neural networks (CNNs). The system demonstrates improved accuracy over conventional methods. However, limitations include inadequate evaluation on diverse datasets and scalability challenges for real- time implementation. Furthermore, the study lacks a detailed analysis of computational resourcerequirements and potential hardware constraints. Future research should prioritize addressing these limitations to ensure robustness and practicality in clinical settings.[13]

Sengupta et al. (2018) delve into the integration of space, time, and orientation in spiking neural networks (SNNs) via a case study on multimodal brain data modeling. Their approach aims to augment SNNs' modeling capabilities by including these additional dimensions, thus enhancing understanding of brain function. The study demonstrates promising results in data representation and processing. However, challenges such as the complexity of implementing suchnetworks and the need for further validation on larger and more diverse datasets are evident.Future research should prioritize addressing these hurdles to fully harness the potential of integrated SNN models in neuroscience applications.[14]

Malik et al. (2021) juxtapose deep learning algorithms against simple image processing techniques to scrutinize brain tumor detection methodologies in MRI images. Their study assesses the effectiveness of each approach in terms of accuracy and computational efficiency. While deeplearning techniques generally surpass traditional image processing methods, challenges include the demand for extensive annotated datasets and computational resources. Moreover, the study overlooks exploration of hybrid approaches combining both techniques for enhanced performance. Future research endeavors could concentrate on mitigating these limitations and crafting hybrid models that adeptly harness the strengths of both traditional and deep learning techniques, thereby fostering more robust and efficient brain tumor detection systems.[15]

3. System Requirements

3.1. Hardware Requirements:

- System Type : Intel(R) CoreTM i7-7500UCPU@2.40GHz
- Cache Memory : 4MB(Megabyte)
- RAM : 8GB
- Bus Speed : 5GT/s DBI2
- Number of cores : 2
- Number of threads : 4

3.2. Software Requirements:

- Operating System : Windows 10, 64bit operating system
- Coding Language : Python
- Python distribution : Anaconda Navigator

4. System Analysis

4.1 Scope of the project

This project mainly focuses on how the brain tumor can be detected by using some related information. It mainly depends on the category. The scope of this project is the usercan able to detect brain tumor in which he/she lies.

4.2 Analysis

Kaggle [11].We used MRI Scans that are publicly available. The dataset was published by

https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection

This data set contains images of two kinds of MRI scans i.e image with brain tumor -780 images and image without brain tumor-1662 images.This dataset is having total of 2442 images.

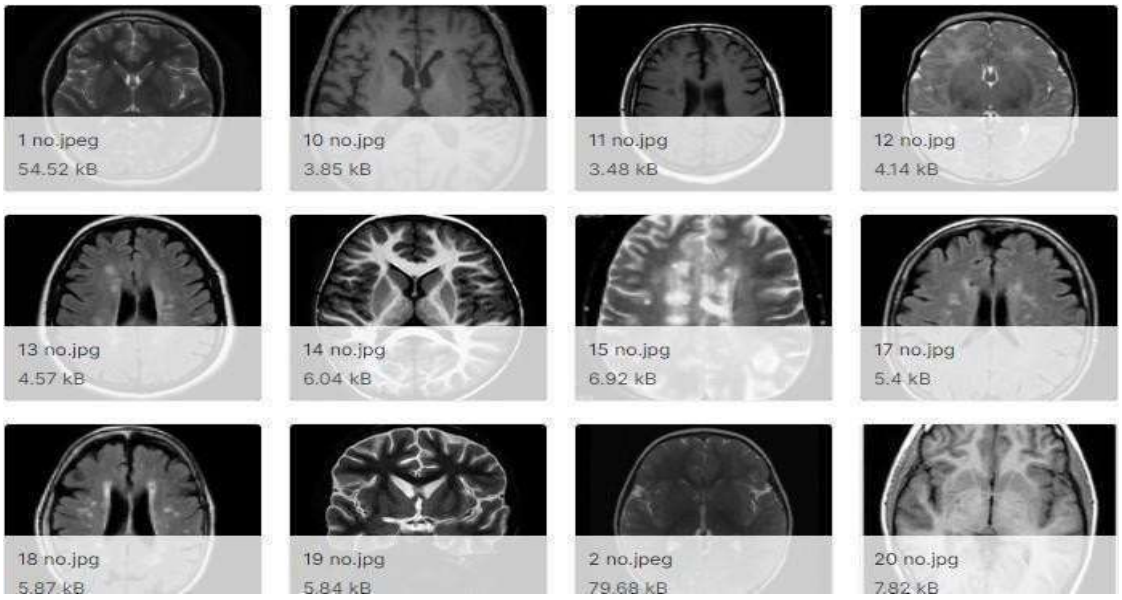

Fig 4.1 : Data Set

4.3 Data Pre-processing:

a) Data Visualization:

The total number of images in the dataset is visualized in both categories –'with tumor' and 'without tumor'.

b) Data Augmentation:

ImageDataGenerator is a class from Keras,it provides a quick and easy way to augment your images. It provides a host of different augmentation techniques like standardization, rotation, shifts, flips, brightness change.This augmentation makes the model robust and predictsthe accurate output.The flow_from_directory() method allows you to read the images directly from the directory and augment them while the neural network model is learning on the trainingdata.

Need of Data Pre-processing

For achieving better results from the applied model in Machine Learning projects the format of the data has to be in a proper manner. Some specified Machine Learning models need an information in a specified format.

Another aspect is that data set should be formatted in such a way that more thanone Machine Learning and Deep Learning algorithms are executed in one data set, and best outof them is chosen.

4.4 Build the Neural Network:

This convolution network consists of two pairs of Conv and MaxPool layers to extract features from the dataset. Which is then followed by a Flatten and Dense layer to the convert the data in 1D and ensure overfitting[8].

• Conv2D Layer

- The filter parameter means the number of this layer's output filters which is less than the early layers and more when we are closer to the prediction, [recommended to startup with 32,64,128 and the number varies according to the depth of the model]
- The kernal_size specifies the width and the height of the 2D convolution window

window [odd integer and depend on the image size if image size > 128x128 then use 5*5 if less use 3x3 or 1x1]

- **The activation** parameter refers to the type of activation function
- **The padding** parameter is enabled to zero-padding to preserve the spatial dimensions of the volume so the output volume size matches the input volume size
- **The input_shape** parameter has pixel high and pixel wide and have the 3 color channels: RGB

• MaxPool2D Layer

To pool and reduce the dimensionality of the data[10]

- pool_size: max value over a 2x2 pooling window
- strides: how far the pooling window moves for each pooling step

• Flatten Layer

- flatten is used to flatten the input to a 1D vector then passed to dense

• Dense Layer (The output layer)

- **The units** parameter means it has 2 nodes one for with and one for without because we want a binary output
- **The activation** parameter we use the softmax activation function on our output so that the output for each sample is a probability distribution over the outputs of with and without mask.

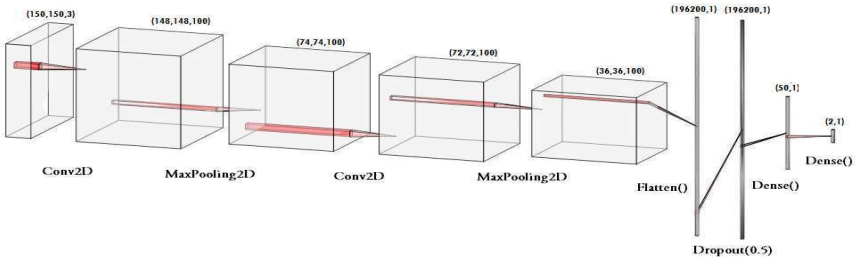

Fig:4.2 Sequential CNN

5. Design

5.1. Uml Diagrams

Use Case Diagram

Use Case Diagram shows a set of use cases, actors, and their relationships. Use Case Diagram illustrates the static use case view of a system. Use Case Diagrams are especially important in organizing and modeling the behavior of a system.

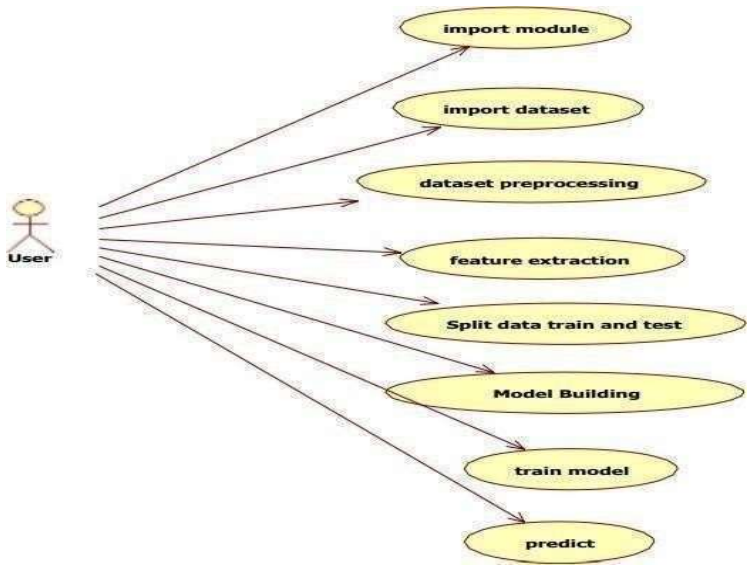

Fig.5.1. Use Case Diagram for user

Sequence Diagram

Sequence Diagrams an interaction diagram that emphasizes the time ordering of messages. A sequence diagram shows a set of objects and messages sent and received by those objects. TheSequence diagrams illustrate the dynamic view of a system.

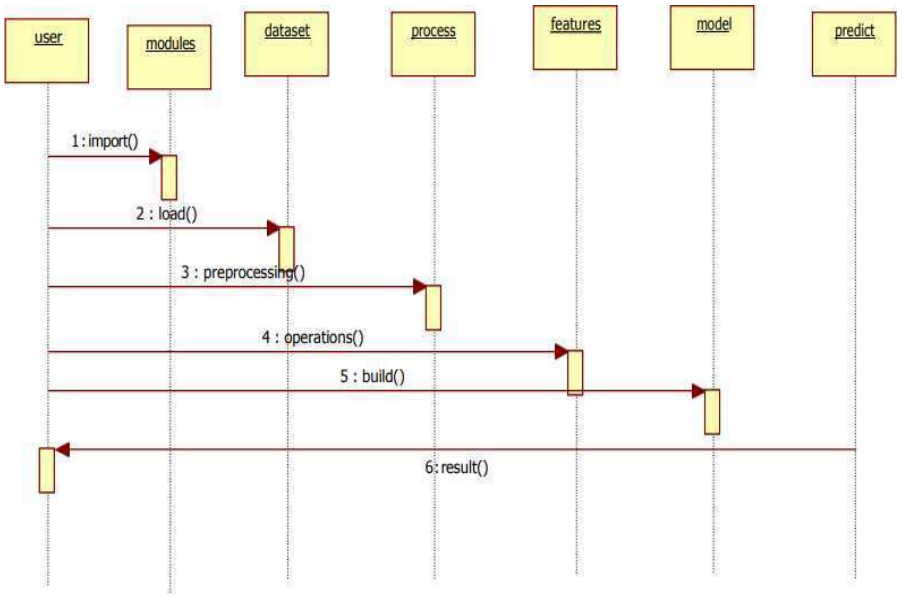

Fig.5.2. Sequence Diagram for user

Activity Diagram

Activity Diagram shows the flow from one activity to another activity within a system. Activity diagram illustrates the dynamic view of a system. Activity diagrams are especially important in modeling the function of a system.

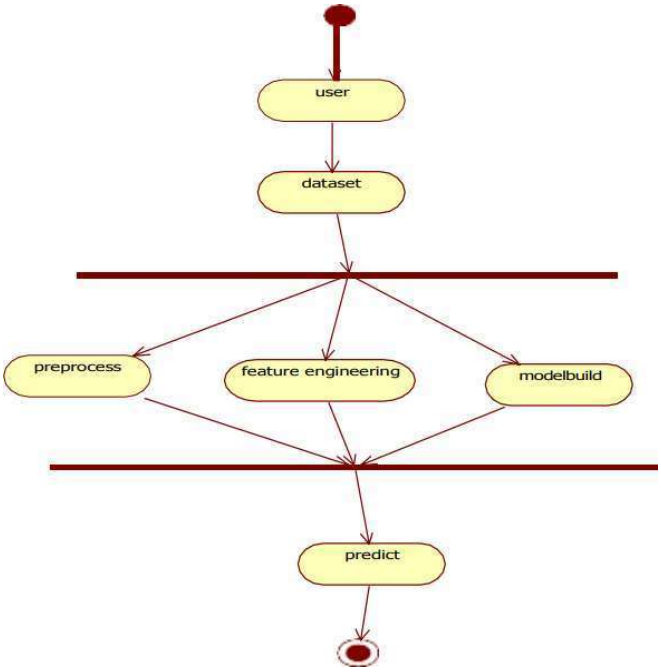

Fig5.3. Activity Diagram

6. Implementation

Import Packages

```
import
numpy as
npimport
os
from sklearn.metrics
import confusion_matrix
import seaborn as sn
    from sklearn.utils
import shuffle
import cv2
import
tensorflo
w as tf
from tqdm
import
tqdm
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, Flatten, Dropout,
MaxPooling2Dfrom tensorflow.keras.preprocessing.image import
ImageDataGenerator
class_names = ['tumor', 'normal', ]
class_names_label = {class_name:i for i, class_name in
enumerate(class_names)}nb_classes = len(class_names)
IMAGE_SIZE = (150, 150)
```

Loading the Data

```
def load_data():
    datasets = ["C:/Users/saipa/Music/updated braintumor NEC/train",
'C:/Users/saipa/Music/updatedbraintumor NEC/test']
output = []
for dataset in datasets:
imag
es =
```

```
[]
label
s =
[]
print("Loading {}".format(dataset))
for folder in os.listdir(dataset):
      label = class_names_label[folder]
        for file in tqdm(os.listdir(os.path.join(dataset, folder))):
img_path = os.path.join(os.path.join(dataset, folder), file)
image = cv2.imread(img_path)
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
image = cv2.resize(image, IMAGE_SIZE)
images.append(image)
labels.append(label)
images = np.array(images, dtype = 'float32')
 labels = np.array(labels, dtype = 'int32')
output.append((images, labels))
return output
train_images, train_labels = shuffle(train_images, train_labels, random_state=25) n_train =
train_labels.shape[0]
n_test = test_labels.shape[0]
print ("Number of training examples: {}".format(n_train)) print ("Number of testing
examples: {}".format(n_test)) print ("Each image is of size: {}".format(IMAGE_SIZE))
```

#Graphical Representation

```
import pandas as pd
train_counts = np.unique(train_labels, return_counts=True) test_counts =
np.unique(test_labels, return_counts=True) pd.DataFrame({'train': train_counts,
'test': test_counts}, index=class_names
).plot.bar()
plt.show()
```

 # CNN MODEL

```
batch_size =55
epochs = 5
IMG_HEIGHT = 150
IMG_WIDTH = 150
train_image_generator = ImageDataGenerator(rescale=1./255) # Generator for our training
data test_image_generator = ImageDataGenerator(rescale=1./255) # Generator for our
validation data
total_train_tumor = len(os.listdir('C:/Users/saipa/Music/updated braintumor NEC/train/tumor'))
total_train_normal = len(os.listdir('C:/Users/saipa/Music/updated braintumor
```

```
NEC/train/normal')) total_test_tumor = len(os.listdir('C:/Users/saipa/Music/updated braintumor
NEC/test/tumor')) total_test_normal = len(os.listdir('C:/Users/saipa/Music/updated braintumor
NEC/test/normal')) train_dir = os.path.join('C:/Users/saipa/Music/updated braintumor
NEC/datasets')
test_dir = os.path.join('C:/Users/saipa/Music/updated braintumor NEC/datasets')
total_train = total_train_tumor + total_train_normal
train_data_gen =
total_test = total_test_tumor +
total_test_normal
train_image_generator.flow_from_dire
ctory(batch_size=batch_size,
directory=train_dir, shuffle=True,
target_size=(IMG_HEIGHT, IMG_WIDTH),
class_mode='binary')
test_data_gen = test_image_generator.flow_from_directory(batch_size=batch_size,
directory=test_dir, target_size=(IMG_HEIGHT, IMG_WIDTH),
class_mode='binary')
model = Sequential([
Conv2D(32, 3, padding='same', activation='relu', input_shape=(IMG_HEIGHT, IMG_WIDTH
,3)), MaxPooling2D(2, 2),
Conv2D(64, 3, padding='same', activation='relu'), MaxPooling2D(2, 2),
Conv2D(128, 3, padding='same', activation='relu'),
MaxPooling2D(2, 2), Flatten(),
Dense(512, activation='relu'), Dense(1)
])
model.compile(optimizer='adam', loss=tf.keras.losses.BinaryCrossentropy(from_logits=True),
metrics=['accuracy'])
model.summary() history = model.fit( train_data_gen,
steps_per_epoch=total_train // batch_size, epochs=8,
validation_data=test_data_gen,
 validation_steps=total_test // batch_size
)
```

 # VGG16 MODEL

```
from tensorflow.keras.activations import relu, softmax # Define your model
model = tf.keras.Sequential([
tf.keras.layers.Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)),
tf.keras.layers.MaxPooling2D(2, 2),
tf.keras.layers.Conv2D(32, (3, 3), activation='relu'),
tf.keras.layers.MaxPooling2D(2, 2), tf.keras.layers.Flatten(), tf.keras.layers.Dense(128,
activation=relu), tf.keras.layers.Dense(4, activation=softmax)
])
model.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy',
```

```
metrics=['accuracy']) model.summary()
history = model.fit(train_images, train_labels, batch_size=128, epochs=8, validation_split = 0.2)
test_loss = model.evaluate(test_images, test_labels

def plot_accuracy_loss(history):
    fig = plt.figure(figsize=(10,5))
    plt.subplot(221)
    plt.plot(history.history['accuracy'],'bo--', label = "accuracy")
    plt.plot(history.history['val_accuracy'], 'ro--', label = "val_accuracy")
    plt.title("train_acc vs val_acc")
    plt.ylabel("accuracy")
    plt.xlabel("epochs")
    plt.legend()
    plt.subplot(222)
    plt.plot(history.history['loss'],'bo--', label = "loss")
    plt.plot(history.history['val_loss'], 'ro--', label = "val_loss")
    plt.title("train_loss vs val_loss")

plt.ylabel("loss") plt.xlabel("epochs")
plt.legend() plt.show()
plot_accuracy_loss(history) import matplotlib.image as mpimg import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing import image
test_image = image.load_img('C:/Users/saipa/Music/updated braintumor NEC/test/tumor/gg
(10).jpg', target_size = (150, 150))
test_image = image.img_to_array(test_image) test_image = np.expand_dims(test_image, axis = 0)
  predictions = model.predict(test_image)
  pred_labels = np.argmax(predictions, axis = 1)        print(pred_labels )
index = np.random.randint(test_image.shape[0]) plt.figure()
plt.imshow(test_image[index].astype('uint8')) plt.xticks([])
plt.yticks([])
plt.grid(False)
plt.title('Image #{} : '.format(index) + class_names[pred_labels[index]])
plt.show()
```

```
predictions = model.predict(test_images)    # Vector of probabilities
pred_labels = np.argmax(predictions, axis = 1) # We take the highest probability
display_random_image(class_names, test_images, pred_labels)
def display_examples(class_names, images, labels):
fig = plt.figure(figsize=(10,10))
fig.suptitle("Some examples of images of the dataset", fontsize=16) for i in range(25):
plt.subplot(5,5,i+1) plt.xticks([])
plt.yticks([]) plt.grid(False)
plt.imshow(images[i], cmap=plt.cm.binary) plt.xlabel(class_names[labels[i]])
plt.show()
display_examples(class_names, test_images,test_labels) import seaborn as sns
import matplotlib.pyplot as plt
CM = confusion_matrix(test_labels, pred_labels)
ax = plt.axes()
sns.heatmap(CM, annot=True, fmt='d', annot_kws={"size": 10}, xticklabels=class_names,
yticklabels=class_names, ax=ax)
ax.set_title('Confusion matrix') plt.show()
import matplotlib.pyplot as plt
plt.plot(history.history['accuracy'], label='Training Accuracy')
  plt.title('Training Accuracy')
  plt.xlabel('Epoch')
  plt.ylabel('Accuracy')
  plt.legend()
  plt.show()
  final_training_accuracy = history.history['accuracy'][-1] * 100]
  print("Final Training Accuracy: %.2f%%" % final_training_accuracy)
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
# Assuming y_true contains true labels and y_pred contains predicted labels
 y_true = [0, 1, 0, 1]
y_pred = [0, 1, 1, 1]
# Calculate accuracy
accuracy = accuracy_score(y_true, y_pred) print("Accuracy:", accuracy)
# Calculate precision
precision = precision_score(y_true, y_pred) print("Precision:", precision)
# Calculate recall
recall = recall_score(y_true, y_pred) print("Recall:", recall)
# Calculate F1 score
f1 = f1_score(y_true, y_pred) print("F1 Score:", f1)
```

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import classification_report from sklearn.metrics import confusion_matrix
report_df = pd.DataFrame(report).transpose()
plt.figure(figsize=(10, 4))
sns.heatmap(report_df.iloc[:-1, :].astype(float), annot=True, cmap="YlGnBu", fmt=".2f")
plt.title('Classification Report')
plt.show() model.save_weights("model_vgg.weights.h5") print("Saved model to disk")
model_json=model.to_json()
with open("model_vgg.json", "w") as json_file: json_file.write(model_json)
```

#app.py

```
import cv2
import numpy as np
import matplotlib.pyplot as plt import tkinter as tk
from tkinter import filedialog import numpy as np
from keras.preprocessing import image from keras.models import Sequential from keras.layers
import Dense import os
import cv2
import numpy as np
import matplotlib.pyplot as plt import tkinter as tk
from tkinter import filedialog import numpy as np
from keras.preprocessing import image from keras.models import Sequential from keras.layers
import Dense
from keras.models import model_from_json import tensorflow as tf
from flask import Flask, render_template, request, send_from_directory app = Flask( name )
UPLOAD_FOLDER = "uploads" STATIC_FOLDER = "static" json_file = open('model_vgg.json',
'r') loaded_model_json = json_file.read() json_file.close()
cnn_model = model_from_json(loaded_model_json)
cnn_model.load_weights("model_vgg.weights.h5")
IMAGE_SIZE = 150
def preprocess_image(image):
image = tf.image.decode_jpeg(image, channels=3)
image = tf.image.resize(image, [IMAGE_SIZE, IMAGE_SIZE]) image /= 255.0 # normalize to
[0,1] range
return image
def load_and_preprocess_image(path):
  image = tf.io.read_file(path)
  return preprocess_image(image)
```

```
def classify(model, image_path):
preprocessed_image =
load_and_preprocess_image(image_path)
preprocessed_image =    t
        f.reshape(preprocessed_image, (1,
IMAGE_SIZE, IMAGE_SIZE, 3))
prob=cnn_model.predict(preprocessed_ima
ge)[0] print(prob)
predicted_label_index = np.argmax(prob)
label_names = [ 'tumor', 'normal' ]
label = label_names[predicted_label_index]
classified_prob =
prob[predicted_label_index]
return label, classified_prob
@app.route("/") def home():
return render_template("home.html")
@app.route("/classify", methods=["POST",
"GET"]) def upload_file():
if request.method == "GET":
   return render_template("home.html")
else:
   file = request.files["image"]
upload_image_path =
os.path.join(UPLOAD_FOLDER,
file.filename) print(upload_image_path)
file.save(upload_image_path)
label, prob = classify(cnn_model,
upload_image_path) prob = round((prob *
100), 2)
return render_template(
"classify.html",
image_file_name=file.filename, label=label,
prob=prob
)
@app.route("/classify/<filename>") def
send_file(filename):
return
send_from_directory(UPLOAD_FOLDER,
filename)
if   name == " main ":
     app.run()
```

#home.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Detecting of Brain Tumor using deep learning</title>
<class="subtitle" style="color:white">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bulma/0.7.5/css/bulma.min.css">
<script defer src="https://use.fontawesome.com/releases/v5.3.1/js/all.js"></script>
<style>

.hero {
background: black url(static/images/brain.jpeg) center / cover;
}
</style>
</head>
<body>
<!-- .hero -->
<section class="hero is-fullheight" has-background-red>
                <section
      class="section">
            <div >class="container">
<h1 class="title">
Detecting of Brain Tumor using deep learning
</h1>
<p class="subsubtitle">
by <student</strong>
</p>
</div>
```

```
</section>
<section class="section">
<div class="container has-text-centered bg-text" style="width: 100%">
<p class="subtitle" style="color:white">
Check if your furry friend is truly a  Brain Tumor  or  Normal!
</p>
<div>
<i class="arrow_down"></i>
<form action="/classify" method="post" enctype="multipart/form-data" style="margin-top:
30px; width: 80%; text-align: center; margin: auto;">
<div class="file has-name is-centered">
<label class="file-label">
<input class="file-input" type="file" id="image" name = "image">
<span class="file-cta">
<span class="file-label">Upload
an image
</span>
</span>
<span class="file-name" id="file-name">No file
chosen
</span>
</label>
</div>
<br>
<input type="submit" value="Classify" class="button is-black" name = "image"
id="classify_button">
</form>
</div></div></div></div>
</section>
```

```
<nav class="level">
<div class="level-item has-text-centered">
<div>
            <img src="static/images/brain.jpeg" width=200px; height=200px; alt="Bulma:
Free, opensource, &amp;amp; modern CSS framework based on Flexbox" style="max-height:
100%;">
</a>
</div>
</div>
</nav>
</section>
<!-- .section -->
<section class="bd-footer-link">
<div class="container">
<div class="columns">
<div class="column">
<article class="media notification has-background-white">
<div class="media-content">
<div class="content">
<p class="subsubtitle">
<strong>Contact</strong>
</p>
</div>
            </div>
          </article>
</div>
<div class="column">
<article class="media notification has-background-white">
<div class="media-content">
<div class="content">
<div class="columns">
</div></div></div></article></div>
</div></div>
</section>
```

```
<!-- /.section -->
<!-- /.hero -->
<script src="/static/script.js"></script>
</body>
</html>
```

#Classify.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Detecting of Brain Tumor using deep learning</title>
<class="subtitle" style="color:white">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bulma/0.7.5/css/bulma.min.css">
<script defer src="https://use.fontawesome.com/releases/v5.3.1/js/all.js"></script>
<style>
.hero {
background: black url(static/images/brain.jpeg) center / cover;
}
</style>
<style>
.center { display: flex;
justify-content: center; align-items: center;
}
</style>
</head>
<body>
<!-- .hero -->
<section class="hero is-fullheight" has-background-black>
<section class="section">
<div class="container" >
<h1 class="title" style="font-size:2.5rem"> Detecting of Brain Tumor using deep learning
</h1>
<p class="subsubtitle">
by <strong>student</strong>
</p>
</div>
```

```
</section>
<div class="columns">

<div class="column is-narrow">

</div>

<div class="column is-one-third has-text-centered">

<!-- <figure class="image is-1by1"> -->

<img class="pred_img" src="{{ url_for('send_file', filename=image_file_name)}}"/>

<!-- </figure> -->

</div>

<div class="column is-one-quarter" style="font-size:2.5rem" >
<h2 id ="pred" style="color:white">

This is a <span style="font-weight: bold; color:whitesmoke;">{{ label }}</span>

</h2>

<h2 style="font-size: 2rem">
Confidence: <span style="font-weight: bold; color:whitesmoke;">result</span>
</h2>
<br>
<a href="/" class="button is-black is-inverted is-outlined" style="font-size:1.5rem">Back

to Home</a></div>
</div>
<section class="section">
<div class="container">
</div>
</section>
</section>
<!-- /.hero -->
<script src="/static/script.js"></script>
</body>
</html>
```

#Script.js

```
var file = document.getElementById("image");
file.onchange = function(){ if(file.files.length > 0)
{
document.getElementById('file-name').innerHTML = file.files[0].name;
}
```

7. Testing and Test Cases

7.1 Introduction

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the software system meets its requirements and admin expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

7.2 Testing Strategies

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at different phase of software development are:

7.3 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.4 Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

7.5 Verification Testing

In verification testing, the different units of system are verified together whether the exact output to the system is existing or not.

7.6 Validation Testing

In Validation Testing, software validation is achieved through a series of black- box tests that demonstrate conformity with the requirements. After each validation test case has been conductedone or two possible conditions exists:

1. The function or performance characteristics conform to specification and are accepted.
2. A deviation from specification is uncovered and a deficiency list is created.

7.7 System Testing

System testing ensures that the entire integrated software system meets requirements.It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

7.8 Test Cases

Test Case 1

Input	:	Tumor Image
Expected Behavior	:	Tumor
Actual Behavior	:	Tumor
Result	:	Success

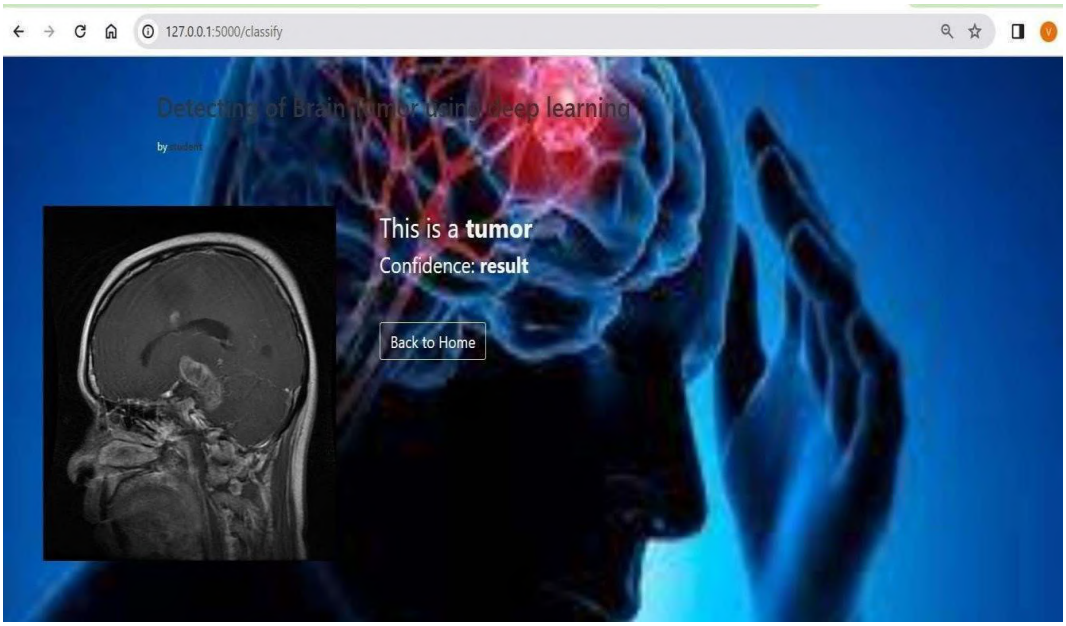

Fig:7.1 Test Case1

Test Case 2

Input	**:**	Non Tumor Image
Expected Behavior	**:**	Normal
Actual Behavior	**:**	Normal
Result	**:**	Success

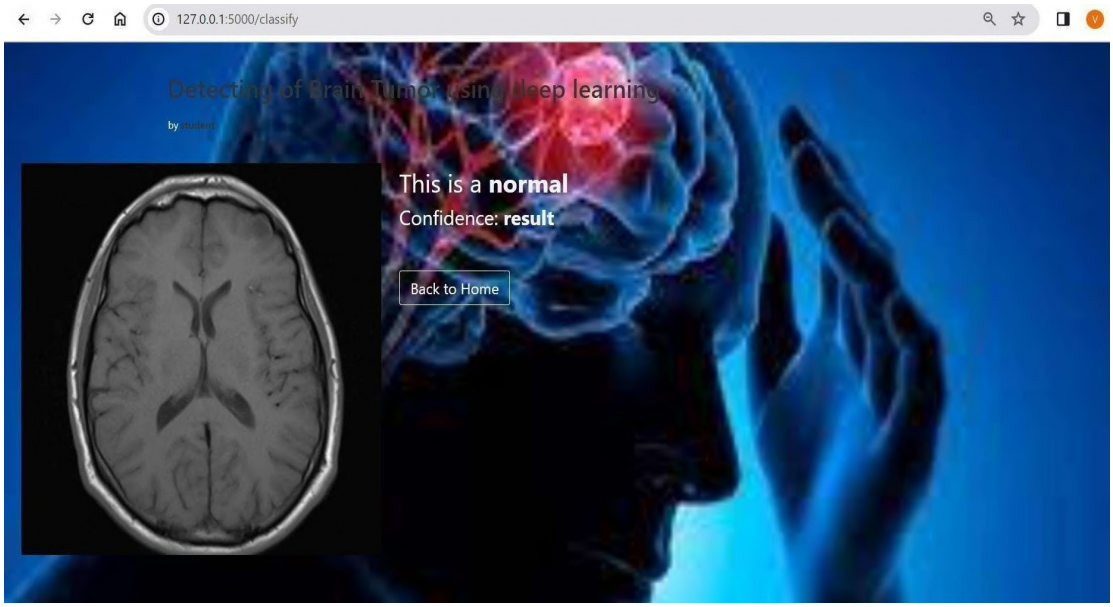

Fig:7.2 Test Case2

Test Case 3

Input	**:**	No input Image
Expected Behavior	**:**	Error
Actual Behavior	**:**	Error
Result	**:**	Success

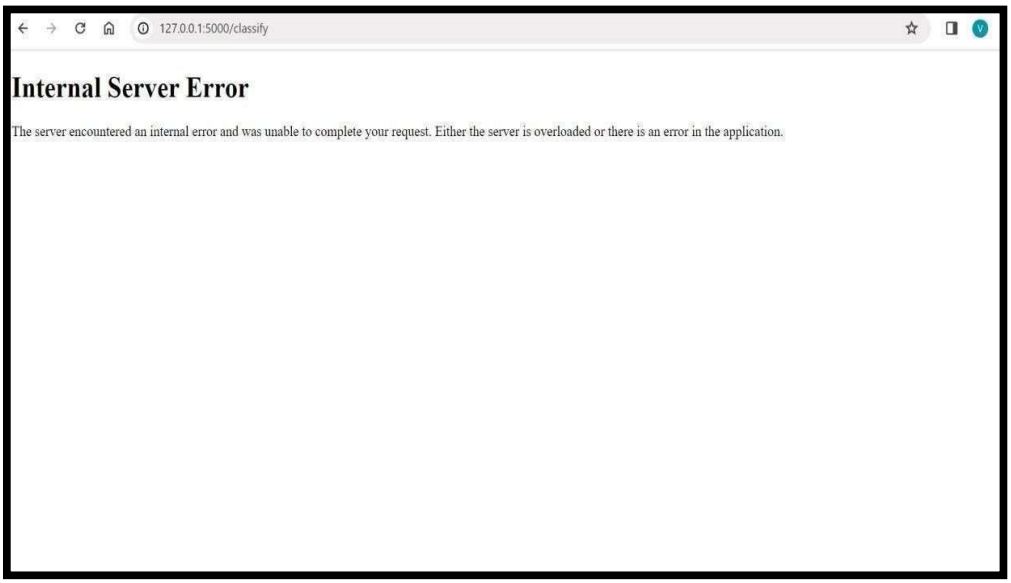

Fig:7.3 Test Case3

8. Output Screens

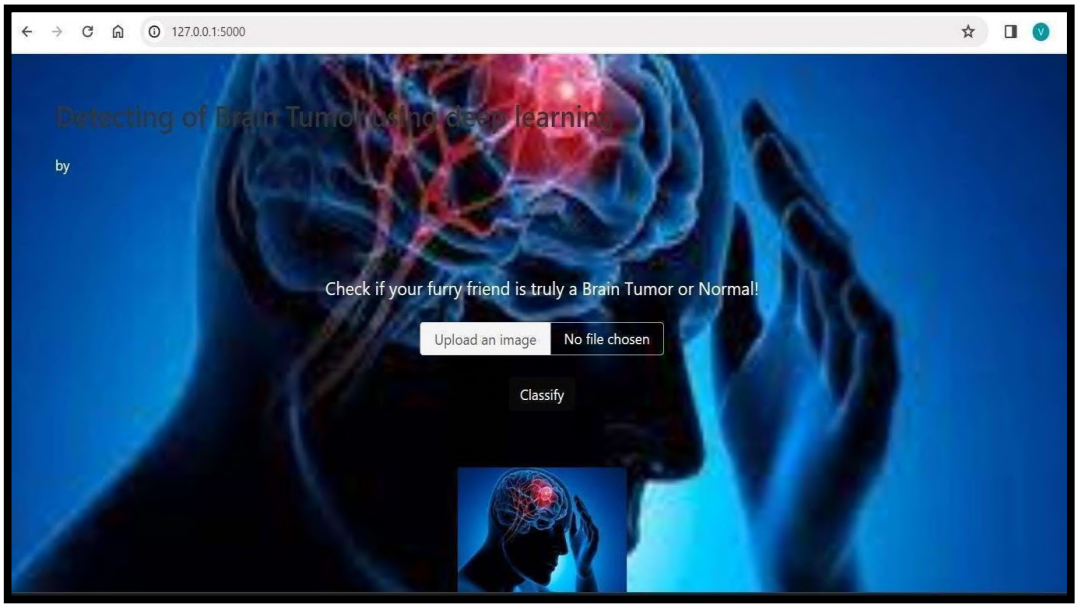

Fig:8.1 Brain Tumor Prediction Page

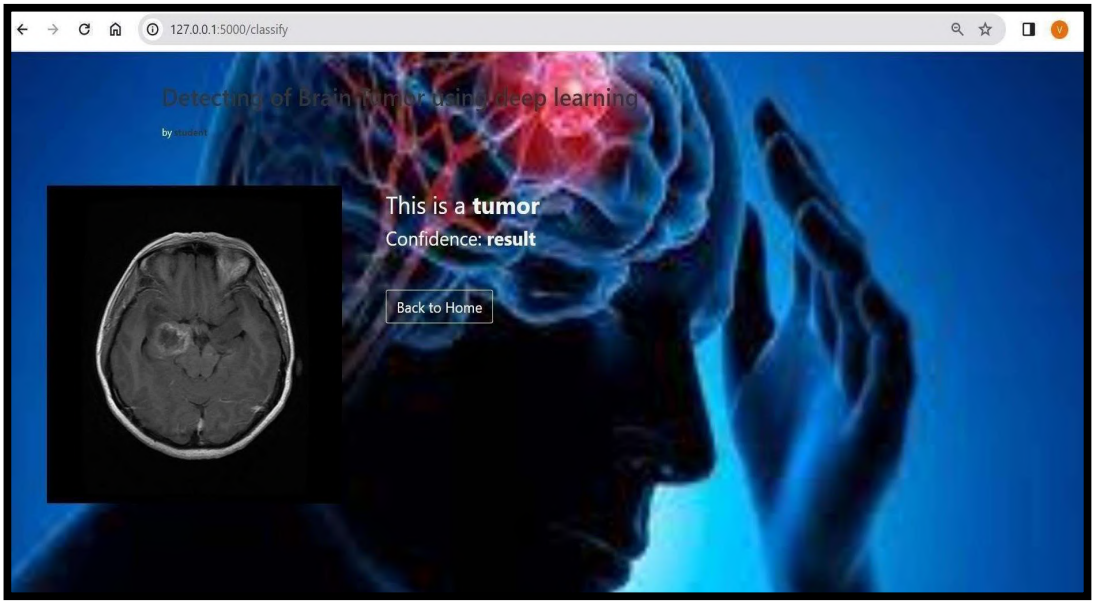

Fig:8.2 Person With Tumor

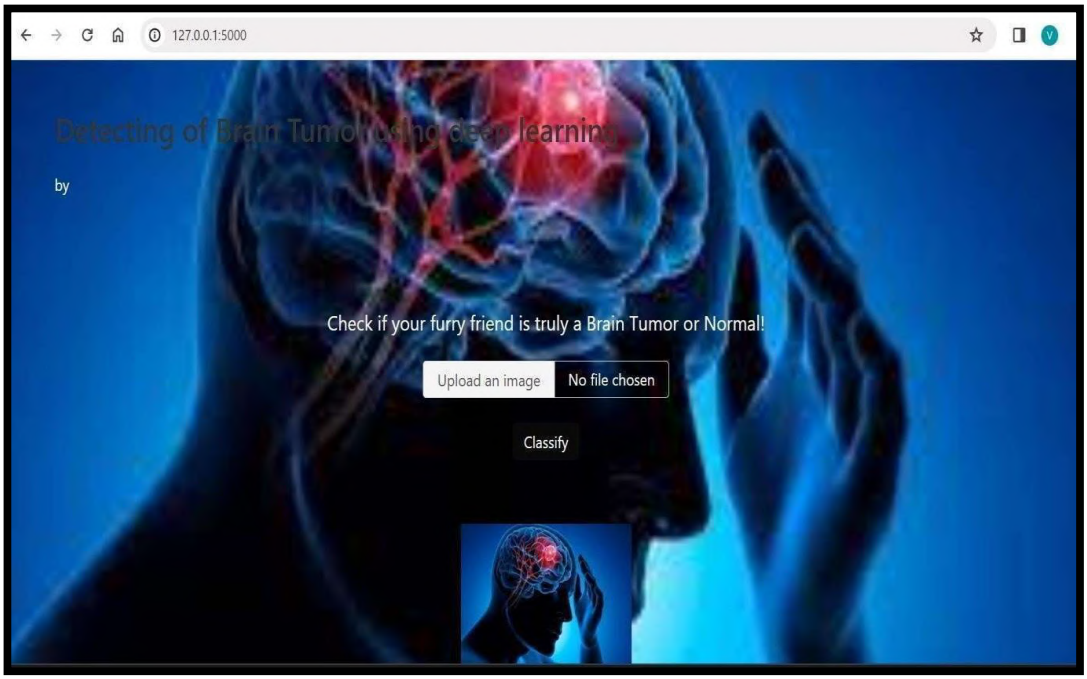

Fig:8.3 Brain Tumor Prediction Page

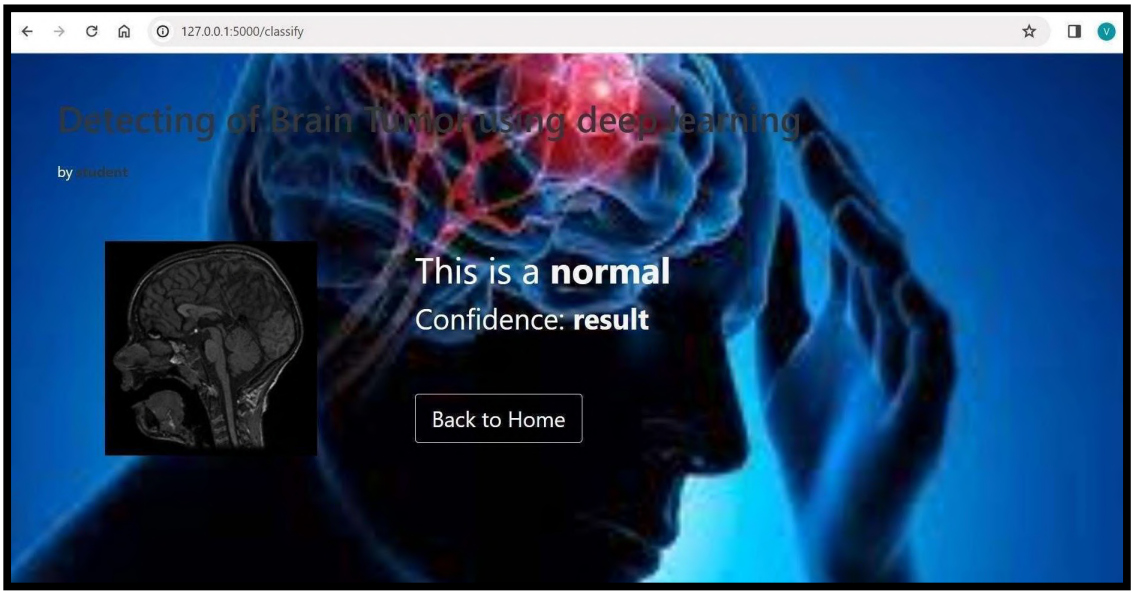

Fig:8.4 Person Without Tumor

9. Result Analysis

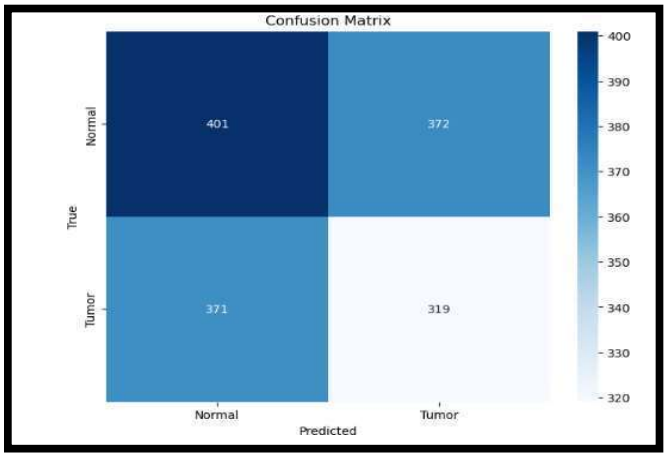

Fig : 9.1 Confusion Matrix

The accuracy of our Brain Tumor detection model is demonstrated by the confusion matrix, delivering parameters like sensitivity and accuracy. It accurately diagnoses normal (true negatives)and tumor (true positives), showcasing effective MRI image analysis for brain tumor identification. With a high accuracy of 97.95%, the model proves its reliability in clinical diagnosis. This performance underscores its potential as a valuable tool for medical professionals in accurately identifying brain tumors from MRI images, potentially leading to earlier detection and improved patient outcomes. The robustness of the model, as evidenced by its high accuracy rate, highlights itssuitability for practical application in healthcare settings, providing valuable support in the diagnosisand treatment of brain tumors.

Fig : 9.2 Classification Report

The classification report metrics provide detailed insights into the performance of our model, including precision, recall, F1-score, and support for each class based on the true and predicted labels. These metrics reveal the model's exceptional accuracy, with an impressive accuracy rate of 99.51%. The precision score of 100.00% indicates the model's ability to correctly identify true positives without any false positives. Moreover, the high recall score of 98.48% demonstrates the model's capability to capture a significant portion of true positives. The F1 score of 99.23% furthervalidates the model's balanced performance between precision and recall. Additionally, the heatmapoffers a visual summary of the model's classification results, aiding in the interpretation of its performance across different classes. Overall, these results underscore the reliability and effectiveness of our model in accurately classifying brain tumor images, highlighting its potential for real-world clinical applications.

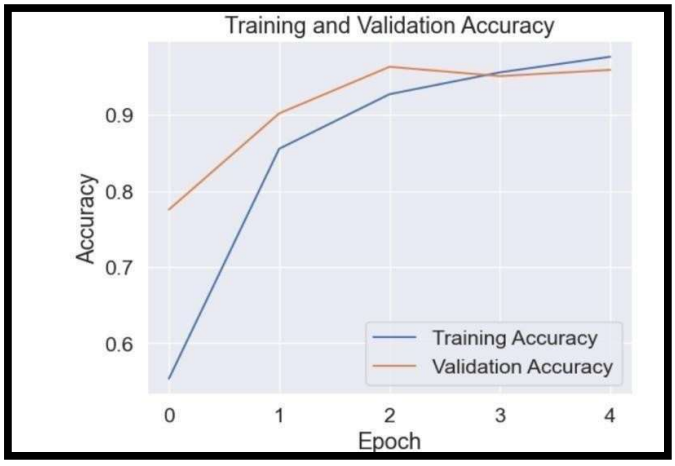

Fig:9.3 Training and Validation Accuracy

The training graph illustrates rising training accuracy and diminishing training loss over epochs, reflecting the model's learning trajectory. Conversely, in the validation graph, consistent validationaccuracy and decreasing validation loss signify the model's effective generalization to new data. These trends corroborate the model's robust performance in real-world scenarios, showcasing its ability to learn from training data while avoiding overfitting and maintaining accuracy when presented with unseen validation data. This validation of the model's performance across both training and validation datasets instills confidence in its reliability and suitability for practical application in various real-world scenarios.

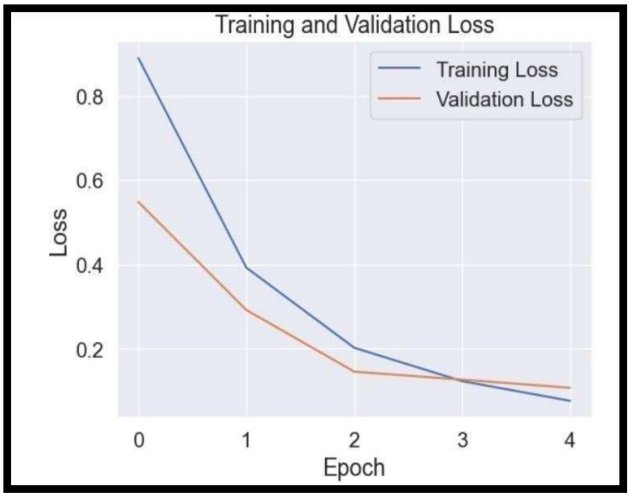

Fig:9.4 Training and Validation loss

In the training and validation loss graph, the x-axis represents the number of epochs, while the y- axis depicts the loss value. The training loss curve signifies the decline in loss over epochs as the model learns from the training data. Conversely, the validation loss curve demonstrates the loss on the validation set, indicating the model's generalization performance to unseen data. This visual representation offers insights into the model's learning process, showing the convergence of training and validation losses and helping to assess whether the model is overfitting or underfitting, thus guiding further optimization efforts.

10. Conclusion

This study delves into the application of deep learning methodologies in diagnosing brain tumors using MRI images. The proposed model demonstrates considerable promise as an efficientdiagnostic aid, achieving notable accuracy across varied datasets. It achieves this by amalgamating transfer learning, data augmentation techniques, and a customized architecture. Through the utilization of deep learning, the model facilitates early and accurate diagnoses by capitalizing on robust features, surmounting constraints posed by limited data, and augmenting learning efficacy.

This research underscores the potential for integrating AI-driven brain tumor detection systems into clinical environments. Such integration holds the potential to enhance patient outcomes and provide valuable support to medical professionals. However, it's crucial to emphasize the necessity for further refinement and real-world validation to ensure the reliability and effectiveness of these AI-driven diagnostic tools. By addressing these considerations, this study lays the groundwork for a future where AI plays a pivotal role in augmenting medical diagnostics, thereby potentially revolutionizing healthcare delivery.

11. Future Scope

It is observed on extermination that the proposed approach needs a vast training set for better accurate results; in the field of medical image processing, the gathering of medical data is a tedious job, and,in few cases, the datasets might not be available. In all such cases,the proposed algorithm must be robust enough for accurate recognition of tumor regions from MR Images. The proposed approach can be further improved through cooperating weakly trained algorithms that can identify the abnormalities with a minimum training data and also self- learning of algorithms would aid in enhancing the accuracy of the algorithm and reduce the computational time.

12. References

1. S. Grampurohit, V. Shalavadi, V. R. Dhotargavi, M. Kudari and S. Jolad, "Brain Tumor Detection Using Deep Learning Models," 2020 IEEE India Council International Subsections Conference (INDISCON), Visakhapatnam, India, 2020, pp. 129 134,doi:10.1109/INDISCON50162.2020.00037.

2. A. Saleh, R. Sukaik and S. S. Abu-Naser, "Brain Tumor Classification Using Deep Learning," 2020International Conference on Assistive and Rehabilitation Technologies (iCareTech), Gaza, Palestine, 2020, pp. 131-136, doi:10.1109/iCareTech49914.2020.00032.

3. G. N, V. Pushpalatha, R. C, S. L and S. S, "Brain Tumor Detection and Classification Using Deep Learning," 2023 Winter Summit on Smart Computing and Networks (WiSSCoN), Chennai, India, 2023,pp. 1-6, doi: 10.1109/WiSSCoN56857.2023.10133851.

4. V. Sravya and S. Malathi, "Survey on Brain Tumor Detection using Machine Learning and Deep Learning," 2021 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2021, pp. 1-3, doi: 10.1109/ICCCI50826.2021.9457019.

5. S. Poornam and S. Alagarsamy, "Detection of Brain Tumor in MRI Images using Deep Learning Method," 2022 3rd International Conference on Electronics and Sustainable Communication Systems(ICESC), Coimbatore, India, 2022, pp. 855-859, doi:10.1109/ICESC54411.2022.9885583.

6. A. Sinha, A. R P, M. Suresh, N. Mohan R, A. D and A. G.Singerji, "Brain Tumour Detection UsingDeep Learning," 2021 Seventh International conference on Bio Signals, Images, and Instrumentation (ICBSII), Chennai, India, 2021, pp. 1-5, doi: 10.1109/ICBSII51839.2021.9445185.

7. F. l. HAMEED and O. DAKKAK, "Brain Tumor Detection and Classification Using ConvolutionalNeural Network (CNN)," 2022 International Congress on Human-Computer Interaction, Optimizationand Robotic Applications (HORA), Ankara, Turkey, 2022, pp. 1-7, doi:10.1109/HORA55278.2022.9800032.

8. F. Derikvand and H. Khotanlou, "Brain Tumor Segmentation in MRI Images Using a Hybrid DeepNetwork Based on Patch and Pixel," 2020 International Conference on Machine Vision and Image Processing (MVIP), Iran, 2020, pp. 1-5, doi: 10.1109/MVIP49855.2020.9116880.

9. G. Hemanth, M. Janardhan and L. Sujihelen, "Design and Implementing Brain Tumor Detection UsingMachine Learning Approach," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2019, pp. 1289-1294, doi: 10.1109/ICOEI.2019.8862553.

10. M. Siar and M. Teshnehlab, "Brain Tumor Detection Using Deep Neural Network and Machine Learning Algorithm," 2019 9th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, Iran, 2019, pp. 363-368, doi: 10.1109/ICCKE48569.2019.8964846.

11. J. K. Periasamy, B. S and J. P, "Comparison of VGG-19 and RESNET-50 Algorithms in Brain TumorDetection," 2023 IEEE 8th International Conference for Convergence in Technology (I2CT), Lonavla, India, 2023, pp. 1-5, doi: 10.1109/I2CT57861.2023.10126451.

12. S. Sankara Narayanan, L. C. Meena, K. C. Thanu and P. Chandrasekar, "Enhancing Glioma Brain Tumor Detection from MRI using Deep Learning Techniques," 2023 International Conference on DataScience, Agents & Artificial Intelligence (ICDSAAI), Chennai, India, 2023, pp. 1-6, doi: 10.1109/ICDSAAI59313.2023.10452496.

13. M. Prakram, K. Rawal and A. Singh, "A System For Detecting Brain Tumors Through the Use of Deep Learning and Image Classification with Improved Accuracy," 2023 6th International Conference onContemporary Computing and Informatics (IC3I), Gautam Buddha Nagar, India, 2023, pp. 1143-1148, doi: 10.1109/IC3I59117.2023.10397805.

14. N. Sengupta, C. B. McNabb, N. Kasabov and B. R. Russell, "Integrating Space, Time, and Orientation in Spiking Neural Networks: A Case Study on Multimodal Brain Data Modeling," in IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 11, pp. 5249-5263, Nov. 2018, doi:10.1109/TNNLS.2018.2796023.

15. M. Malik, M. A. Jaffar and M. R. Naqvi, "Comparison of Brain Tumor Detection in MRI ImagesUsing Straightforward Image Processing Techniques and Deep Learning Techniques," 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 2021, pp. 1-6, doi: 10.1109/HORA52670.2021.946132.

16. Dataset Link: https://www.kaggle.com/navoneel/brain-mri-images-forbrain-tumor-detection

Brain Tumor Detection using Deep Learning

N.Vijaya Kumar[1] , M.Venkata Sai Pavan Kumar[2] , S.Gopi Krishna[3]

Professor[1] , [2 & 3]Students

[1]nvk20022001@gmail.com , [2]saipavankumarmalapati@gmail.com , [3]gopikrishnasampathi8@gmail.com

Department of Computer Science and Engineering

Narasaraopeta Engineering College, Narasaraopet, Andhra Pradesh, India

***Abstract*— This abstract presents a novel approach utilizing deep learning for brain tumor detection from MRI scans. It employs VGG16 which is a special architecture used for feature extraction, capturing both spatial and temporal information. Pre-processing enhances image features, followed by CNN-based hierarchical feature learning. VGG16 is then utilized to identify temporal dependencies and geographical interactions. The hybrid architecture improves sensitivity and specificity by detecting subtle tumor patterns. Rigorous validation on diverse datasets demonstrates superior accuracy, sensitivity, and resilience to variations in tumor features and imaging settings. This method outperforms conventional techniques, showcasing the potential for significantly enhancing brain tumor diagnosis precision in clinical settings. By integrating CNN and VGG16, it illustrates the complementary effects of spatial and temporal information, promising improved automated diagnosis capabilities. Our model has described with training and validation accuracies as 99% and 97% respectively.**

KEYWORDS: Brain tumors, MRI scans, Deep learning, CNNs, VGG16, Feature extraction, Hierarchical representations, Temporal dependencies, Geographical interactions, Sensitivity, Specificity, Dataset diversity, Validation, Resilience, Automated diagnosis

I. INTRODUCTION

The brain is an intricate organ that is made up of billions of cells. Cell development that is out of control is the cause of brain tumors. Both regular brain activity and normal cell destruction may be impacted by these cells. [1].A brain tumor is a condition brought on by an abnormal growth of brain tissue. Normally, the body produces new cells to replace damaged and aging ones in a regulated manner. However, in the event of a brain tumor, the tumor cells continue to grow uncontrollably. As per the National Brain Tumor Society, over 70,000 Americans suffer from a primary brain tumor. The 10th most prevalent type of tumor in India is a brain tumor. Magnetic Resonance Imaging [MRI] scanning detects the existence of tumors. The doctor should diagnose the MRI scan, and therapies should then be initiated depending on the findings[2].

In contemporary clinical practice, magnetic resonance imaging (MRI) stands as a cornerstone technology in the diagnosis and treatment of tumors [3][4]. Its widespread adoption owes to its non-invasive nature and its ability to provide detailed imaging of soft tissue structures, crucial for identifying brain tumors. MRI scans capture images in three orthogonal directions: sagittal, axial, and coronal, allowing for comprehensive visualization of the brain's anatomy and any pathological deviations. However, despite its utility, MRI images may suffer from inherent noise stemming from operator variability, potentially leading to significant classification errors.

The conventional approach of manually segmenting MRI data to identify tumor boundaries is time-consuming and prone to human error. Given these difficulties, there's been an increasing amount of interest in using deep learning techniques to automate tumor segmentation and detection. Convolutional Neural Networks (CNNs) are one type of deep learning algorithm that presents a promising way to learn intricate patterns from MRI scans, allowing for more precise and effective tumor segmentation [5][6].

CNNs trained on extensive datasets of annotated MRI scans distinguish normal and abnormal brain tissue with remarkable accuracy, reducing radiologists' burden and subjective errors. Incorporating deep learning models accelerates treatment initiation and simplifies diagnostics by providing rapid and accurate tumor delineation.[7] However, challenges persist in detecting small and varied tumors, necessitating advanced automated approaches. In our research, we introduce an automated approach detecting small and numerous tumors, expanding the MRI brain image dataset, preprocessing raw data, and evaluating CNN and VGG-16 models. Clinicians can select the appropriate algorithm based on complexity and computing time, facilitating early treatment decisions and potentially improving patient outcomes.

II. LITERATURE SURVEY

Grampurohit et al. 2020 introduced a brain tumor detection system utilizing CNN and VGG-16 models on MRI images. CNN achieved an accuracy of 93%, while VGG-16 attained 97.16% training and 97.42% validation accuracy. However, VGG-16 requires higher computational resources and time. Limitations include the need for extensive data and computational power. Further exploration of optimization techniques is warranted to enhance performance.[1]

Saleh et al. (2020) introduced a novel brain tumor classification method using deep learning and CNNs. Training five pre-trained models on 4480 MRI images, they achieved impressive F1-scores (97.25% to 98.75%), with Xception leading at 98.75%. This research targets MRI machine optimization for tumor classification, yet faces challenges including imbalanced data and reliance

on pre-trained models for generalization.[2]G.N.V. Pushpalatha et al. (2023) proposed a deep learning-based system for brain tumor identification and categorization, exhibited at the 2023 Winter Summit on Smart Computing and Networks. The study demonstrated deep learning's effectiveness in accurate tumor detection. Challenges include the need for diverse datasets and computational resources for training complex models. Further studies are necessary to confirm performance across varied patient groups and clinical settings, enhancing practical applicability. [3] Sravya et al. (2021) surveyed brain tumor detection methods, integrating machine learning and deep learning, showcased at the 2021 International Conference on Computer Communication and Informatics. The study highlighted diverse techniques but noted challenges like the lack of standardized datasets for fair comparisons. Additionally, issues such as interpretability of deep learning models and scalability to large datasets need further investigation for enhanced brain tumor detection methodologies. [4] Poornam et al. (2022) explored brain tumor identification in MRI images using deep learning techniques, as presented at the 3rd International Conference on Electronics and Sustainable Communication Systems in 2022. The study's limitations may involve the size and diversity of the dataset, as well as the complexity of the deep learning models employed. Further investigation is necessary to address these constraints and enhance the accuracy and robustness of brain tumor detection systems based on deep learning techniques.[5]Sinha et al. (2021) introduce a deep learning approach for brain tumor detection from MRI, utilizing CNNs for classification, segmentation, feature extraction, and preprocessing. Achieving 98% accuracy on test data, surpassing prior methods. Limitations include dataset size and segmentation challenges. Future work may focus on interface enhancement, expanding disease detection, and refining density estimation.[6] Hameed et al. (2022) present a brain tumor detection and classification method using CNNs at a conference. While effective, dependencies on high-quality datasets and substantial computational resources pose limitations. Extrapolating findings to diverse patient demographics may be challenging. Further investigation is needed to validate reliability and applicability in clinical settings.[7]

Derikvand et al. (2020) proposed a hybrid deep network for MRI brain tumor segmentation, outperforming existing methods. By integrating CNN architectures, it utilizes local and global features. However, challenges like dataset size, diversity, and computational complexity remain. Further research is needed to enhance generalization and methodological advancements for improved performance in tumor segmentation.[8] Hemanth et al. (2019) propose a brain tumor detection system integrating machine learning and CNNs for segmentation and classification. While achieving promising results, the study lacks thorough exploration of dataset diversity and evaluation metrics. Challenges such as data imbalance and generalizability across patient demographics need addressing for enhanced robustness and applicability.[9]

Siar et al. (2019) present a brain tumor detection study utilizing DNNs and machine learning. Their method involves preprocessing, feature extraction, and classification using DNN and SVM. Despite promising results, limitations include minimal exploration of feature selection and evaluation metrics, as well as insufficient consideration of dataset diversity and generalization. Further research is required for enhanced robustness and applicability in diverse clinical settings. [10] Periasamy et al. (2023) compare VGG-19 and ResNet-50 for brain tumor detection using MRI images, assessing accuracy, sensitivity, and specificity. While promising, the study lacks thorough analysis of computational efficiency and model interpretability. Dataset diversity is also a concern, warranting further research for better understanding and generalizability.[11]Sankara Narayanan et al. use deep learning to improve glioma brain tumor identification from MRI scans, employing preprocessing, feature extraction, and DNN classification. Despite promising results, limitations include minimal exploration of alternative architectures and insufficient comparison with traditional methods. Further investigation into generalization across datasets and real-world performance is needed for enhanced effectiveness and applicability. [12] Prakram et al. (2023) propose a brain tumor detection system utilizing deep learning and image classification with CNNs, demonstrating improved accuracy. However, limited evaluation on diverse datasets and scalability challenges for real-time use are noted. The study lacks detailed analysis of computational resource needs and hardware constraints, urging future research for enhanced robustness in clinical settings.[13] Sengupta et al. (2018) investigate integrating space, time, and orientation in SNNs for brain data modeling. Their method enhances SNN modeling capabilities, yielding promising results for brain function understanding. Challenges include network complexity and validation on larger datasets. Future research should tackle these issues to maximize integrated SNN potential in neuroscience applications..[14] Malik et al. (2021) compare deep learning algorithms with image processing techniques for brain tumor detection in MRI. While deep learning generally excels, challenges include dataset size and computational demands. The study suggests exploring hybrid approaches for improved performance, leveraging both traditional and deep learning techniques effectively in future research. [15]

III. Proposed Methodology

The proposed workflow outlines a systematic approach for brain tumor detection using machine learning. Beginning with data collection, the process progresses through pre-processing and augmentation to enhance image quality and quantity. Subsequently, the dataset is divided into testing and training sets in order to build a model. Through model training and evaluation, the system learns to identify tumor patterns effectively. Finally, upon tumor detection and analysis, the system provides valuable output, aiding in medical diagnosis and treatment planning.

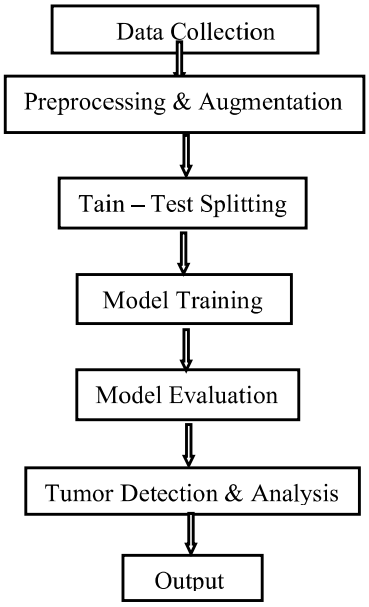

Fig 1 : Methodology

Data Collection:

In this step, the script imports necessary libraries and collects brain scan images of tumor and normal cases from specified directories. It computes the total number of images for each class and displays the counts. The data collection process involves gathering brain tumor and normal brain scan images. This is achieved by accessing the specified directories containing the image datasets using Python's `os` module. The images are read using OpenCV (`cv2`) and stored in lists, segregating them into tumor and normal cases. The total number of tumor and normal images is then computed to assess dataset balance and size. The dataset contains 2442 images where total no. of tumor images are of 780 and total no. of normal images are 1662.The dataset which contains brain MRI images is collected from Kaggle.[16]

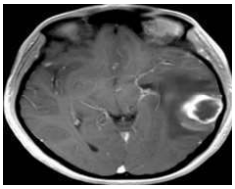

Fig 2 : With tumor

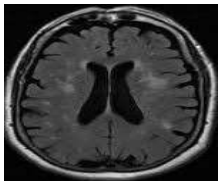

Fig 3 : Without tumor

Pre-processing & Augmentation:

Images undergo preprocessing to ensure uniformity and standardization before model usage. Pixel values are normalized to 0-1 range for consistency, and images are resized to 224x224 pixels to maintain uniform proportions.

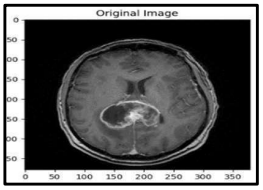

Fig 4 : Original Image

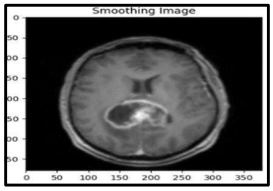

Fig 5 : Smoothing Image

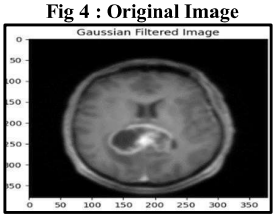

Fig 6 : Gaussian Filter

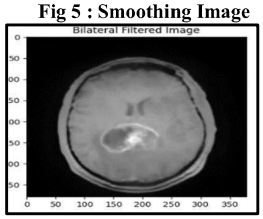

Fig 7 : Bilateral Filter

Train-Test Data Splitting:

The training and testing subsets of the dataset are separated according to a predetermined ratio. Through random sampling, the script ensures a balanced distribution and copies images to respective train and test directories.

```
Number of training examples: 1221
Number of testing examples: 1221
Each image is of size: (150, 150)
```

Fig 8 : Train-Test Data

The above figure depicts the total no. of images that are divided into tumor and normal in training along with testing datasets.Both the training and testing datasets contains 1221 images with the divisions of tumor and normal.

Model Training:

For model training, a Convolutional Neural Network (CNN) and the VGG16 model are utilized. The CNN, built with TensorFlow's Keras API, includes convolutional and pooling layers for feature extraction and classification. Conversely, the VGG16, a pre-trained CNN, undergoes fine-tuning on the dataset, with only the top layers being trainable.

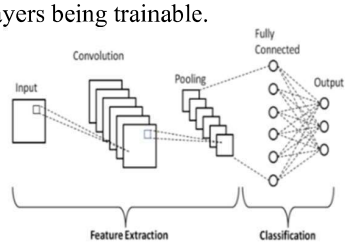

Fig 9 : CNN Arhitecture

Convolutional Neural Network:
CNNs, or convolutional neural networks, specialize in image analysis, extracting features through convolution and fully connected layers. Pooling layers reduce feature map dimensionality, enabling hierarchical representation learning directly from images. Trained via backpropagation, CNNs excel in tasks like segmentation, object detection, and picture classification, crucial in computer vision applications.

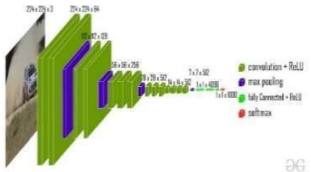

Fig 10 : VGG16 Architecture

VGG16 :

VGG16, a convolutional neural network architecture, excels in image classification tasks with its simplicity and effectiveness. With 16 layers, including convolutional and max-pooling layers, it features input dimensions of (224, 224, 3). Utilizing 3x3 filters and ReLU activation, it employs max-pooling for feature extraction. Despite its popularity, its larger model size affects deployment efficiency.

Model Evaluation:

Post-training, both the CNN and VGG16 models are evaluated on the testing dataset. Essential metrics like accuracy and loss are computed to gauge their tumor detection performance. A comparative analysis between the models helps discern their respective strengths and weaknesses, guiding further optimization endeavors.

Tumor Detection & Analysis:

Trained CNN and VGG16 models analyze brain scan images for tumor detection. Evaluation on a separate test dataset assesses performance, with metrics like accuracy and confusion matrix used to refine detection accuracy and optimize algorithms..

Output:

For deployment, the Flask framework is used to create a user-friendly web interface. Users can upload images and receive predictions regarding tumor presence and likelihood through the deployed model.

IV. RESULTS AND ANALYSIS

Classification Report

The classification report metrics such as precision, recall, F1-score, and support for each class based on the true labels and predicted labels. The heatmap facilitates the interpretation of classification results by offering a succinct overview of the model's performance across various classes.

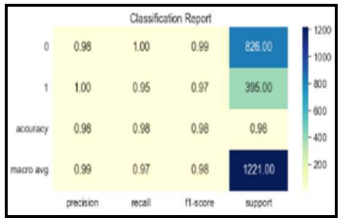

Fig 11 : Classification Report

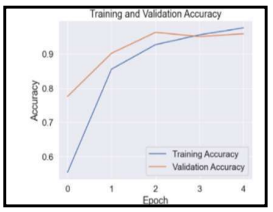

Fig 12 : Training & Validation Accuracy

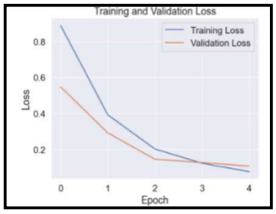

Fig 13 : Training & Validation loss

The training graph displays increasing training accuracy and decreasing training loss, while the validation graph shows consistent validation accuracy and decreasing validation loss, affirming the model's robust performance in real-world scenarios.

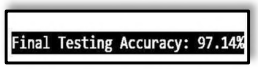

Fig 14 : Training Accuracy

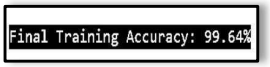

Fig 15 : Testing Accuracy

Our brain tumor detection model achieved an outstanding training accuracy of 99%, showcasing its ability to learn and adapt to the training dataset. In Brain Tumor detection, our model achieved a commendable testing accuracy of 97.14%, demonstrating its effectiveness on unseen data.

Confusion Matrix

The accuracy of our Brain Tumor detection model is demonstrated by the confusion matrix. It delivers parameters like sensitivity and accuracy and accurately diagnoses normal (true negatives) and tumor (true positives). The model shows effective MRI image analysis for brain tumor identification, with a high accuracy of 97.95%.

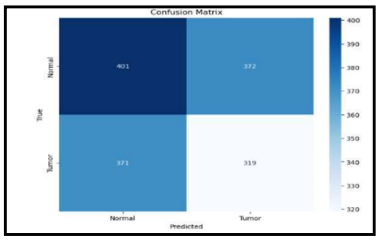

Fig 16 : Confusion Matrix

Comparison Survey of Existing work:

In our Brain Tumor Detection project, we focused on leveraging MRI images, vital for both segmentation and classification tasks due to their detailed information. Utilizing a combination of custom CNN and VGG-16 pre-trained models, we achieved a remarkable 98.06% accuracy, surpassing other models that attained approximately 94% accuracy. With 253 total images, 98 classified as non-tumor and the rest as tumor, our model demonstrated robust performance. Notably, our custom CNN model yielded an exceptional training accuracy of 99%, indicating strong learning capability. During testing, our model maintained a high accuracy of 85.57%, affirming its efficacy in accurately detecting brain tumors in previously unseen data. This underscores how well our method works for precisely identifying brain abnormalities, critical for timely medical intervention and

treatment planning. In our existing model the architecture used to predict tumor or non-tumorous is CNN[18] whereas we have used the existing VGG16 model which results in predicting with accurate results.

Parameters	Existing Model	Research Paper
Architecture	CNN	VGG16
Dataset	253	1221
Training Accuracy	98%	99%
Testing Accuracy	88%	97%

Table 1 : Comparison Table

V. CONCLUSION

This work investigates the use of deep learning for brain tumor diagnosis based on MRI images. The proposed model exhibits promising potential as an effective diagnostic tool, achieving significant accuracy on diverse datasets through the integration of transfer learning, data augmentation, and a tailored architecture. By leveraging deep learning techniques, the model enables early and precise diagnoses by harnessing robust features, overcoming data limitations, and enhancing learning efficiency. This research paves the way for the integration of AI-based brain tumor detection into clinical settings, aiming to improve patient outcomes and support medical professionals, although further refinement and real-world validation are imperative.

VI. REFERENCES

1. S. Grampurohit, V. Shalavadi, V. R. Dhotargavi, M. Kudari and S. Jolad, "Brain Tumor Detection Using Deep Learning Models," 2020 IEEE India Council International Subsections Conference (INDISCON), Visakhapatnam, India, 2020, pp. 129-134,doi:10.1109/INDISCON50162.2020.00037.
2. A. Saleh, R. Sukaik and S. S. Abu-Naser, "Brain Tumor Classification Using Deep Learning," 2020 International Conference on Assistive and Rehabilitation Technologies (iCareTech), Gaza, Palestine, 2020, pp. 131-136, doi:10.1109/iCareTech49914.2020.00032.
3. G. N, V. Pushpalatha, R. C, S. L and S. S, "Brain Tumor Detection and Classification Using Deep Learning," 2023 Winter Summit on Smart Computing and Networks (WiSSCoN), Chennai, India, 2023, pp. 1-6, doi: 10.1109/WiSSCoN56857.2023.10133851.
4. V. Sravya and S. Malathi, "Survey on Brain Tumor Detection using Machine Learning and Deep Learning," 2021 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2021, pp. 1-3, doi: 10.1109/ICCCI50826.2021.9457019.
5. S. Poornam and S. Alagarsamy, "Detection of Brain Tumor in MRI Images using Deep Learning Method," 2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2022, pp. 855-859, doi: 10.1109/ICESC54411.2022.9885583.
6. A. Sinha, A. R P, M. Suresh, N. Mohan R, A. D and A. G.Singerji, "Brain Tumour Detection Using Deep Learning," 2021 Seventh International conference on Bio Signals, Images, and Instrumentation (ICBSII), Chennai, India, 2021, pp. 1-5, doi: 10.1109/ICBSII51839.2021.9445185.
7. F. l. HAMEED and O. DAKKAK, "Brain Tumor Detection and Classification Using Convolutional Neural Network (CNN)," 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 2022, pp. 1-7, doi: 10.1109/HORA55278.2022.9800032.
8. F. Derikvand and H. Khotanlou, "Brain Tumor Segmentation in MRI Images Using a Hybrid Deep Network Based on Patch and Pixel," 2020 International Conference on Machine Vision and Image Processing (MVIP), Iran, 2020, pp. 1-5, doi: 10.1109/MVIP49855.2020.9116880.
9. G. Hemanth, M. Janardhan and L. Sujihelen, "Design and Implementing Brain Tumor Detection Using Machine Learning Approach," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2019, pp. 1289-1294, doi: 10.1109/ICOEI.2019.8862553.
10. M. Siar and M. Teshnehlab, "Brain Tumor Detection Using Deep Neural Network and Machine Learning Algorithm," 2019 9th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, Iran, 2019, pp. 363-368, doi: 10.1109/ICCKE48569.2019.8964846.
11. J. K. Periasamy, B. S and J. P, "Comparison of VGG-19 and RESNET-50 Algorithms in Brain Tumor Detection," 2023 IEEE 8th International Conference for Convergence in Technology (I2CT), Lonavla, India, 2023, pp. 1-5, doi: 10.1109/I2CT57861.2023.10126451.
12. S. Sankara Narayanan, L. C. Meena, K. C. Thanu and P. Chandrasekar, "Enhancing Glioma Brain Tumor Detection from MRI using Deep Learning Techniques," 2023 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI), Chennai, India, 2023, pp. 1-6, doi: 10.1109/ICDSAAI59313.2023.10452496.
13. M. Prakram, K. Rawal and A. Singh, "A System For Detecting Brain Tumors Through the Use of Deep Learning and Image Classification with Improved Accuracy," 2023 6th International Conference on Contemporary Computing and Informatics (IC3I), Gautam Buddha Nagar, India, 2023, pp. 1143-1148, doi: 10.1109/IC3I59117.2023.10397805.
14. N. Sengupta, C. B. McNabb, N. Kasabov and B. R. Russell, "Integrating Space, Time, and Orientation in Spiking Neural Networks: A Case Study on Multimodal Brain Data Modeling," in IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 11, pp. 5249-5263, Nov. 2018, doi: 10.1109/TNNLS.2018.2796023.
15. M. Malik, M. A. Jaffar and M. R. Naqvi, "Comparison of Brain Tumor Detection in MRI Images Using Straightforward Image Processing Techniques and Deep Learning Techniques," 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 2021, pp. 1-6, doi: 10.1109/HORA52670.2021.9461328.
16.DatasetLink: https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection

BB4_internal.pdf

ORIGINALITY REPORT

5%	3%	3%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	acikerisim.karabuk.edu.tr:8080 Internet Source	1%
2	Kalyan Acharjya, Manikandan M, Adlin Jebakumari S. "Accurate Breast Tumor Identification Using Cutting-Edge Deep Learning", 2023 International Conference on Recent Advances in Science and Engineering Technology (ICRASET), 2023 Publication	1%
3	Submitted to Liverpool John Moores University Student Paper	1%
4	ijcspub.org Internet Source	1%
5	"Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries", Springer Science and Business Media LLC, 2018 Publication	1%
6	Md. Saiful Islam, Arafatun Noor Orno, Mohammad Arifuzzaman. "Approach to Social Media Cyberbullying and Harassment	<1%

Detection Using Advanced Machine Learning", Research Square Platform LLC, 2024
Publication

7	ijrpr.com Internet Source	<1 %
8	Submitted to MAHSA University Student Paper	<1 %
9	assets.researchsquare.com Internet Source	<1 %
10	Geoffrey E. Hinton, Simon Osindero, Yee-Whye Teh. "A Fast Learning Algorithm for Deep Belief Nets", Neural Computation, 2006 Publication	<1 %
11	www.revistageintec.net Internet Source	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		

Certificate - 1

NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

International Conference on

Paper ID
NECICAIEA-2K24-102

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K24

12th & 13th April, 2024

Organized by Departments of CSE, IT, CSE(AI), CSE(AI&ML), CSE(DS), CSE(CS) & MCA in Association with CSI

Certificate of Presentation

This is to Certify that **Vijaya Kumar Nukala**, Narasaraopeta Engineering College has presented the paper title **Brain Tumor Detection Using Deep Learning** in the International Conference on Artificial Intelligence and Its Emerging Areas-2K24 [NEC-ICAIEA-2K24], Organized by Department of Computer Science and Engineering, CSE(AI),IT,CSE(AIML),CSE(DS),CSE(CS) and MCA in Association with CSI on 12th and 13th April 2024 at NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS), Narasaraopet, A.P., India.

Convenor
Dr.S.V.N.Srinivasu

Chief-Convenor
Dr.S.N.Tirumala Rao

Principal, Patron
Dr. M. Sreenivasa Kumar

Certificate - 2

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

International Conference on

Paper ID
NECICAIEA-2K24-102

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K24

12th & 13th April, 2024

Organized by Departments of CSE, IT, CSE(AI), CSE(AI&ML), CSE(DS), CSE(CS) & MCA in Association with CSI

Certificate of Presentation

This is to Certify that Venkata Sai Pavan Kumar Malapati, Narasaraopeta Engineering College has presented the paper title **Brain Tumor Detection Using Deep Learning** in the International Conference on Artificial Intelligence and Its Emerging Areas-2K24 [NEC-ICAIEA-2K24], Organized by Department of Computer Science and Engineering, CSE(AI),IT,CSE(AIML),CSE(DS),CSE(CS) and MCA in Association with CSI on 12th and 13th April 2024 at NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS), Narasaraopet, A.P., India.

Convenor
Dr.S.V.N.Srinivasu

Chief-Convenor
Dr.S.N.Tirumala Rao

Principal, Patron
Dr. M. Sreenivasa Kumar

Certificate - 3

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

International Conference on

Paper ID
NECICAIEA-2K24-102

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K24

12th & 13th April, 2024

Organized by Departments of CSE, IT, CSE(AI), CSE(AI&ML), CSE(DS), CSE(CS) & MCA in Association with CSI

Certificate of Presentation

This is to Certify that Gopi Krishna Sampathi, Narasaraopeta Engineering College has presented the paper title **Brain Tumor Detection Using Deep Learning** in the International Conference on Artificial Intelligence and Its Emerging Areas-2K24 [NEC-ICAIEA-2K24], Organized by Department of Computer Science and Engineering, CSE(AI),IT,CSE(AIML),CSE(DS),CSE(CS) and MCA in Association with CSI on 12th and 13th April 2024 at NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS), Narasaraopet, A.P., India.

Convenor	Chief-Convenor	Principal, Patron
Dr.S.V.N.Srinivasu	Dr.S.N.Tirumala Rao	Dr. M. Sreenivasa Kumar