**A Project report on**

**“ REVOLUTIONIZING BRAIN TUMOUR DIAGNOSIS : DEEP LEARNING- POWERED PRECISION AND**

**PERSONALIZED HEALTHCARE ”**

Submitted in partial fulfillment of the requirement for the award of the degree of

**BACHELOR OF TECHNOLOGY**

**in**

**ARTIFICIAL INTELLIGENCE & DATA SCIENCE**

**By**

**SHAIK IDAITHULLA 20701A3014**

**G.L.N CHAKRADHAR 21705A3002**

**SHAIK Md GHOUSE 20701A3027**

**CHAVALA SANTHOSH 20701A3040**

*Under the esteemed guidance of*

**Mr. A RAMESH BABU**

Assistant Professor in AI & DS, AITS.



Submitted to

**Department of Artificial Intelligence & Data Science Annamacharya Institute of Technology and Sciences**

(An Autonomous Institution)

(Approved by AICTE, New-Delhi and affiliated to J.N.T.U, Anantapur)

( Accredited by NBA &NAAC)

New Boyanapalli, Rajampet, Annamaiah (Dt), A.P-516 126

**2022-2023**

**Department of Artificial Intelligence & Data Science**

**Annamacharya Institute of Technology and Sciences**

(An Autonomous Institution)

(Approved by AICTE, New-Delhi and affiliated to J.N.T.U, Anantapur)

( Accredited by NBA & NAAC )

New Boyanapalli, Rajampet ,Annamaiah (Dt), A.P-516 126



**CERTIFICATE**

This is to certify that the project report entitled **“REVOLUTIONIZING BRAIN**

**TUMOUR DIAGNOSIS : DEEP LEARNING- POWERED PRECISION AND PERSONALIZED HEALTHCARE**” is submitted by

**SHAIK IDAITHULLA 20701A3014**

**G.L.N CHAKRADHAR 21705A3002 SHAIK Md GHOUSE 20701A3027**

**CHAVALA SANTHOSH 20701A3040**

in partial fulfillment of the requirements for the award of Degree of **Bachelor of Technology** in “**Artificial Intelligence and Data Science**” for the academic year 2023-24.

|  |  |
| --- | --- |
| **Signature of Guide:** | **Signature of HOD:** |
| **Mr.B Dilip Chakravarthy,M.Tech.,(Ph.D)** | **Dr. M. Subba Rao Ph.D.,** |
| Assistant Professor in CSE, | Professor & Head, Dept. of CSE, |
| AITS,Rajampet. | Dean of Student Affairs, AITS, Rajampet. |

**Department of Artificial Intelligence & Data Science**

**Annamacharya Institute of Technology and Sciences**

(An Autonomous Institution)

(Approved by AICTE, New-Delhi and affiliated to J.N.T.U, Anantapur)

( Accredited by NBA & NAAC )

New Boyanapalli, Rajampet ,Annamaiah (Dt), A.P-516 126



**CERTIFICATE**

This is to certify that the project report entitled **“REVOLUTIONIZING BRAIN**

**TUMOUR DIAGNOSIS : DEEP LEARNING- POWERED PRECISION AND PERSONALIZED HEALTHCARE**” is submitted by

**SHAIK IDAITHULLA 20701A3014**

**G.L.N CHAKRADHAR 21705A3002 SHAIK Md GHOUSE 20701A3027**

**CHAVALA SANTHOSH 20701A3040**

in partial fulfillment of the requirements for the award of Degree of **Bachelor of Technology** in “**Artificial Intelligence and Data Science**” is a record of bonafide work carried out by her during the academic year 2023-24.

Project viva-voce held on :

Internal Examiner External Examiner

**Department of Artificial Intelligence and Data Science**

**Annamacharya Institute of Technology and Sciences**

(An Autonomous Institution)

(Approved by AICTE, New-Delhi and affiliated to J.N.T.U, Anantapur)

(Accredited by NBA & NAAC)

New Boyanapalli, Rajampet , Annamaiah (Dt), A.P-516 126



**ANTI-PLAGIARISM CERTIFICATE**

This is to certify that the project report entitled **“REVOLUTIONIZING BRAIN**

**TUMOUR DIAGNOSIS : DEEP LEARNING- POWERED PRECISION AND PERSONALIZED HEALTHCARE**” is submitted by

**SHAIK IDAITHULLA 20701A3014**

**G.L.N CHAKRADHAR 21705A3002**

**SHAIK Md GHOUSE 20701A3027**

**CHAVALA SANTHOSH 20701A3040**

in partial fulfillment of the requirements for the award of Degree of **Bachelor of Technology** in “**Artificial Intelligence and Data Science**”. Course contains the plagiarism of % which is within the acceptable limits.

**Date: Dean / Coordinator**

**Research & Development Cell Date:**

**DECLARATION**

We hereby declare that the project report entitled “**REVOLUTIONIZING BRAIN**

**TUMOUR DIAGNOSIS : DEEP LEARNING- POWERED PRECISION AND**

**PERSONALIZED HEALTHCARE”** under the guidance of **Mr. A RAMESH BABU**

**Assistant Professor in AI&DS**, Department of Artificial Intelligence and Data Science is submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Artificial Intelligence and Data Science.

This is a record of bonafide work carried out by me/us and the results embodied in this project report have not been reproduced or copied from any source. The results embodied in this project report have been submitted to any other University or institute for the Award of any other Degree or Diploma.

**PROJECT ASSOCIATES**

**SHAIK IDAITHULLA**

**G.L.N CHAKRADHAR**

**SHAIK MOHAMMED GHOUSE**

**CHAVALA SANTHOSH**

### *ACKNOWLEDGMENT*

*We endeavor of a long period can be successful only with the advice of many well- wishers. We take this opportunity to express my deep gratitude and appreciation to all those who encouraged me for the successful completion of the project work.*

*Our heartfelt thanks to our Guide,* ***Mr. A Ramesh Babu****Assistant Professor in Department of Artificial Intelligence and Data Science, Annamacharya Institute of Technology and Sciences, Rajampet, for his valuable guidance and suggestions in analyzing and testing throughout the period, till the end of the project work completion.*

*We wish to express sincere thanks and gratitude to* ***Dr. M. Subba Rao****, Head of the Department of Computer Science and Engineering, for his encouragement and facilities that were offered to us for carrying out this project.*

*We take this opportunity to offer gratefulness to our Principal* ***Dr. S.M.V. Narayana****,*

*for providing all sorts of help during the project work.*

*We are very much thankful to* ***Dr. C. Gangi Reddy****, Honorary Secretary of the Annamacharya Educational Trust, for his help in providing good facilities in our college.*

*We would express our sincere thanks to all faculty members of* ***Artificial Intelligence and Data Science Department, batch-mates, friends*** *and* ***lab-technicians****, who have helped us to complete the project work successfully.*

*Finally, we express our sincere thanks to* ***our parents*** *who has provided their heartfelt support and encouragement in the accomplishment to complete this project successfully.*

**PROJECT ASSOCIATES**

*SHAIK IDAITHULLA*

*G.L.N CHAKRADHAR*

*SHAIK MOHAMMED GHOUSE*

*CHAVALA SANTHOSH*

**CONTENTS**

**CHAPTER PAGE NO**

**ABSTRACT**

1. **INTRODUCTION** 1-4
   1. Motivation 1-2
   2. Definition of Brain Tumor 3
2. **LITERATURE SURVEY** 4-13
3. **SYSTEM ANALYSIS** 15-19

[3.1. Existing System 14](#_Toc74711)

[3.1.1. Disadvantages 14 15](#_Toc74712)

[3.2. Proposed System 15](#_Toc74713)

[3.2.1. Advantages 15](#_Toc74714)

[3.3. Convolutional Neural Networks 15 16](#_Toc74715)

[3.4. Modules in Proposed System 16](#_Toc74716)

[3.4.1. User 16](#_Toc74717)

[3.4.2. System 17](#_Toc74718)

[3.4.3. Algorithms Used 17 18](#_Toc74719)

3.4.3.1. CNN 17

1. **SYSTEM REQUIREMENTS SPECIFICATIONS** 20-23
   1. Software Requirements 20
   2. Hardware Requirements 20
   3. Feasibility Study 20-21
      1. Economic Feasibility 20
      2. Technical Feasibility 21
      3. Behavioural Feasibility 21
      4. Benefits of Doing Feasibility Study 21
   4. Functional and Non-Functional Requirements 22-23
      1. Functional Requirements 22
      2. Non-Functional Requirements 23
2. **SYSTEM DESIGN** 24-38
   1. Architecture Design 24
   2. Introduction to UML Diagrams 25
      1. Goals 25-26
   3. UML Notations 26-27
   4. UML Diagrams 32-39
      1. Use Case Diagram 28-29
      2. Class Diagram 29-30
      3. Sequence Diagram 30-31
      4. Collaboration Diagram 31-32
      5. Deployment Diagram 32-33
      6. Activity Diagram 33-34
      7. Component Diagram 34-35
   5. ER Diagram 36
3. **SYSTEM CODING AND IMPLEMENTATION** 39-58
   1. Introduction to Python Programming Language 39-41
      1. Benefits of Python 41
   2. Python Libraries Used in Python 41-42
   3. Code 42-54
4. **SYSTEM TESTING** 55-60
   1. Software Testing Techniques 55-56
      1. Testing Objectives 55
      2. Test Case Design 55
      3. White Box Testing 55
      4. Black Box Testing 56
   2. Software Testing Strategies 57-64
      1. Unit Testing 57
      2. Integration Testing 57-58
      3. Validation Testing 58
      4. System Testing 59
      5. Security Testing 59
      6. Performance Testing 59-60
   3. Test cases 60
5. **RESULTS** 61-66
6. **CONCLUSION AND FUTURE ENHANCEMENTS** 67

**BILIOGRAPHY PLAGARISM REPORT JOURNAL PUBLICATION**

**LIST OF FIGURES & TABLES**

**Fig. No. Figures Page No.**

1.1. Types of Hate Content 3

* 1. Architecture Diagram 24
  2. Use Case Diagram 29
  3. Class Diagram 30
  4. Sequence Diagram 31
  5. Collaboration Diagram 32
  6. Deployment Diagram 33
  7. Activity Diagram 34
  8. Component Diagram 35

5.10. ER Diagram 36

* 1. Working Of Python Program 39
  2. Implementation of Python Program 40
  3. Home Page 61
  4. About Page 61
  5. Registration Page 62
  6. Login Page 62
  7. User Home Page 63
  8. Load Page 63
  9. View Page 64
  10. Pre processing Page 64
  11. Model Training Page 65
  12. Prediction Page 66

**ABSTRACT**

This study uses a user-friendly Streamlit user interface (UI) along with Convolutional Neural Networks (CNN) to provide a fresh method to brain tumor identification. 'Glioma tumor','meningioma tumor', 'no tumor', and 'pituitary tumor' are the four categories into which the CNN model effectively classifies brain pictures (59.95% accuracy). Users may easily engage with the system using the accompanying UI, which provides an easy platform for uploading brain pictures for analysis. As soon as the photos are uploaded, the CNN model quickly processes them and produces a very accurate categorization result. Moreover, each tumor type's symptoms and description are fully provided by the user interface (UI), which helps users comprehend the consequences of a diagnosis. In summary, our initiative improves the accessibility and precision of brain tumor diagnosis by smoothly fusing cutting-edge CNN technology with a user-centric user interface. It enables users to make well-informed decisions about brain health by providing them with thorough information and simple functionality.

***Keywords:*** *Brain tumor detection, Convolutional Neural Networks (CNNs,*

*Medical imaging, Streamlit user interface (UI), Diagnostic accuracy, Healthcare technology, Automated classification, Patient empowerment Healthcare accessibility and Future enhancements.*

**CHAPTER-1**

**INTRODUCTION**

**1. INTRODUCTON**

### 1.1. MOTIVATION

This work is motivated by the urgent need for more accessible and accurate brain tumor detection techniques. Brain tumors are a major global health problem that have a variety of effects on people and healthcare systems. Conventional diagnostic techniques sometimes entail invasive procedures or drawn-out imaging processes, which delays the start of therapy and raises patient concern. In addition, a delayed or incorrect diagnosis can have serious repercussions that lower quality of life and affect patient outcomes. As a result, creative solutions that expedite detection and enhance diagnostic precision are desperately needed. The frequency of brain tumors poses a variety of issues to society. Lack of widespread access to specialized healthcare facilities with cutting-edge imaging technology and highly qualified medical staff is one of the main problems. Disparities in diagnostic and treatment results result from this, especially in underprivileged groups and rural locations with inadequate healthcare infrastructure. In addition, non-specialist healthcare practitioners may make mistakes or delay in diagnosing patients due to the intricacy of interpreting brain imaging data. Furthermore, the ambiguity surrounding diagnosis and treatment choices can add to the significant emotional and financial strain placed on patients and their families.

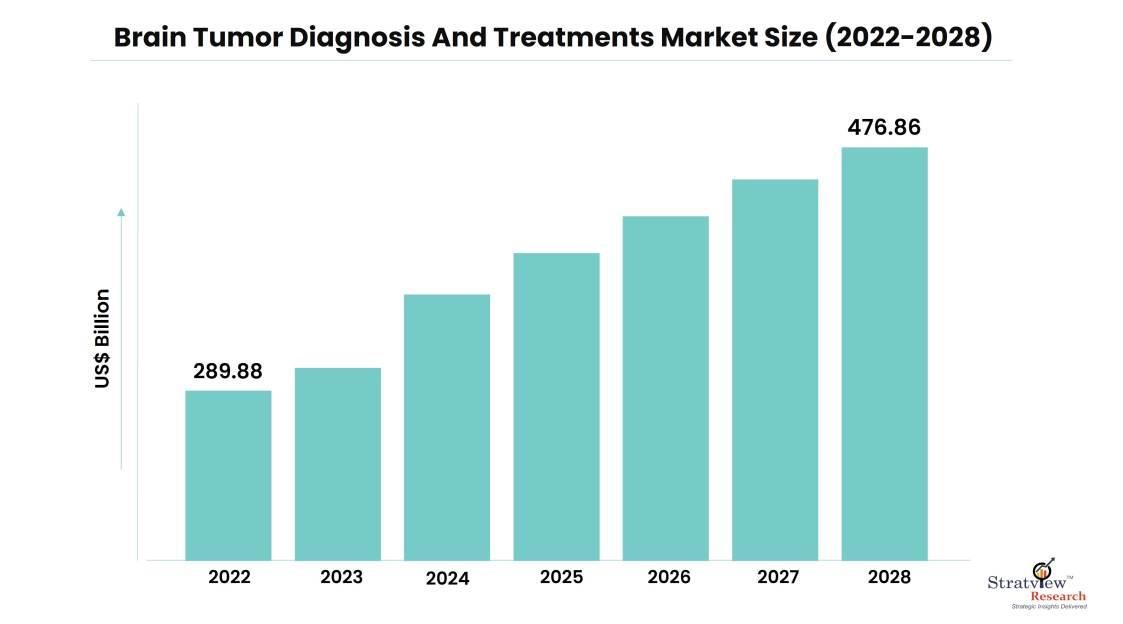
Several social issues might be resolved by creating a brain tumor detection system that is both precise and easy to use. First off, this system may greatly improve the precision and efficacy of brain tumor diagnostics by utilizing state-of-the-art technologies like Convolutional Neural Networks (CNN). Fast and accurate brain picture categorization allows medical professionals to act quickly and make well-informed judgments, which improves patient outcomes and allows therapy to start on time. Furthermore, the use of an intuitive interface guarantees that the system is usable by a broad spectrum of users, including both patients and non-specialist healthcare personnel.

BRAIN TUMOR DETECTION

Additionally, the system enables patients to take an active role in their healthcare by providing comprehensive information on various tumor kinds and their ramifications. This encourages patient participation and education while building their sense of agency and empowerment. Additionally, the technology lowers healthcare expenses related to pointless procedures or postponed treatment commencement by optimizing the diagnosis process. It may also increase overall survival rates and lessen the strain on healthcare systems by enabling early identification and treatments. All things considered, the creation of this brain tumor detection technology is a big step in the right direction toward solving the social issues brought on by brain tumors. It lowers healthcare inequities, encourages patient empowerment, and improves diagnostic accuracy and efficiency by fusing cutting-edge technology with user-centric design concepts. As a result, it has the capacity to significantly influence both society at large and the lives of those impacted by brain tumors.

**1.2 DEFINITION OF PEST MANAGEMENT:**

The process of locating and categorizing anomalies in brain imaging that suggest the existence of tumors is known as "brain tumor detection." In order to analyze and interpret the pictures, it makes use of a variety of imaging modalities, including MRI, CT, and PET scans, as well as sophisticated computer methods, such as machine learning algorithms. The main goal is to precisely locate, measure, and classify any brain tumors so that medical professionals can plan treatments and give the best possible care for their patients. The early detection, prognosis, and monitoring of brain tumors are all greatly aided by this procedure, which ultimately improves patient outcomes and quality of life.



**Figure1.1 Brain Tumor Diagnosis**

**CHAPTER-2**

**LITERATURE SURVEY**

**2. LITERATURE SURVEY**

1. **Siar, Masoumeh, and Mohammad Teshnehlab. "Brain tumor detection using deep neural network and machine learning algorithm." 2019 9th international conference on computer and knowledge engineering (ICCKE). IEEE, 2019.**

There are two categories of brain tumors: benign and malignant. Patients who receive early and efficient illness identification and treatment have longer life expectancies and a higher quality of life. The usage of deep neural networks is among the most significant and useful techniques (DNN). This research presents the usage of a Convolutional Neural Network (CNN) for brain Magnetic Resonance Imaging (MRI) tumor detection. Initially, images were applied to CNN. The Softmax Fully Connected layer achieved a classification accuracy of 98.67% for the pictures. Additionally, the CNN's accuracy is 97.34% with the Radial Basis Function (RBF) classifier and 94.24% with the Decision Tree (DT) classifier. We employ the benchmarks of Sensitivity, Specificity, and Precision in addition to the accuracy criteria to assess network performance. Based on the categorization findings, the CNN's Softmax classifier has the highest accuracy according to the network accuracy results for the picture testing.

1. **Hemanth, G., 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). IEEE, 2019.**

In the field of medicine nowadays, brain tumor detection has emerged as a widespread cause. A brain tumor is defined as an abnormal mass of tissue in which there is no control over the cells' uncontrollable, rapid multiplication. Image segmentation is the method used to identify the aberrant tumor location in the brain. The ability to discern the existence of outlines related to a brain tumor in an MRI requires a high degree of brain tissue segmentation. The health care industry has a lot of secret data locked away. Early illness prediction is effectively possible with the right application of precise data mining categorization algorithms.

1. ***Sajid, Sidra, Saddam Hussain, and Amna Sarwar. "Brain tumor detection and segmentation in MR images using deep learning." Arabian Journal for Science and Engineering 44 (2019): 9249-9261.***

The most insidious and potentially fatal brain tumors, gliomas grow incredibly quickly. Because of their uneven form and dispersed borders with the surrounding region, gliomas are difficult to segment using computer-aided diagnostics. The most popular technique for visualizing interesting brain regions is magnetic resonance imaging, or MRI. This paper presents a deep learning-based approach for brain tumor segmentation using several MRI modalities. In order to predict the output label, the suggested hybrid convolutional neural network architecture employs a patch-based methodology and considers both local and contextual information. The suggested network uses a two-phase training process to address the data imbalance issue and a dropout regularizer in conjunction with batch normalization to address the over-fitting problem.

1. **Saba, Tanzila, et al. "Brain tumor detection using fusion of hand crafted and deep learning features." Cognitive Systems Research 59 (2020): 221-230.**

Brain tumors are currently the most dangerous illness in the world. A tumor damages healthy brain tissue or raises intracranial pressure, which both have an impact on the brain. Therefore, tumor cells may die as a result of their fast development. Early detection of brain tumors is therefore more important since it can protect patients from harmful consequences. The Grab Cut method is utilized in the proposed work to accurately segment actual lesion symptoms, and the Transfer learning model visual geometry group (VGG-19) is fine-tuned to acquire the features. The hand-crafted (shape and texture) features are then concatenated using a serial based method. Entropy is utilized to optimize these characteristics for quick and accurate classification, and classifiers are given a fused vector.

1. **Amin, Javaria, et al. "Brain tumor detection using statistical and machine learning method." Computer methods and programs in biomedicine 177 (2019): 69-79.**

* 1. brain tumor develops as a result of abnormal cell growth. It is among the leading causes of adult mortality worldwide. Early brain tumor identification has the potential to save millions of lives. A patient's chance of survival may rise with earlier Magnetic Resonance Imaging (MRI) brain tumor discovery. Tumor visibility is improved in MRIs, which facilitates the course of further therapy. The goal of this endeavor is to find tumors early on. In this paper, the input slices are enhanced and the noise is reduced using a Weiner filter with several wavelet bands. Using Potential Field (PF) clustering, subsets of tumor pixels are identified. Furthermore, in Fluid Attenuated Inversion Recovery (Flair) and T2 MRI, the tumor zone is isolated using a global threshold and several mathematical morphological techniques. Features from the Gabor Wavelet Transform (GWT) and Local Binary Pattern (LBP) are combined for reliable classification.

1. **Khan, Abdul Hannan, et al. "Intelligent model for brain tumor identification using deep learning." Applied Computational Intelligence and Soft Computing 2022 (2022):**

**1-10.**

* 1. significant contributing factor to psychological issues including anxiety and despair is brain tumors. Tumor healing is more successful when a brain tumor is detected early and treated appropriately. Medical image processing is a critical component in helping people diagnose various illnesses. A large portion of brain tumor categorization is based on the medical professional's experience and training. To assist doctors, a sophisticated system for identifying and categorizing brain tumors is crucial. The study's unique aspect is its use of a hierarchical deep learning technique to classify brain tumors into three categories: glioma, meningioma, and pituitary. For a prompt and effective cure, the diagnosis, tumor classification, and medical image processing with a convolutional neural network (CNN) are crucial.

1. **Almadhoun, Hamza Rafiq, and Samy S. Abu-Naser. "Detection of brain tumor using deep learning." (2022).**

The field of computer science known as artificial intelligence (AI) focuses on building intelligent computers or software that functions and behaves like humans. Speech, recognition, learning, planning, and problem solving are just a few of the computer tasks that fall under this umbrella. Deep learning is a subset of machine learning algorithms that belong to a large family of machine learning techniques that are predicated on learning data representations. Utilizing Magnetic Resonance imagery (MRI) imagery for quick and simple brain tumor diagnosis, deep learning is utilized to create models for brain tumor detection and classification. This thesis will examine several methods and processes for creating a model using deep learning techniques.

1. **Nazir, Maria, Sadia Shakil, and Khurram Khurshid. "Role of deep learning in brain tumor detection and classification (2015 to 2020): A review." Computerized medical imaging and graphics 91 (2021): 101940.**

In every sense imaginable, computer vision and machine learning have transformed the world in the past ten years. Because of its prowess in managing vast volumes of data, deep learning is a subfield of machine learning that has demonstrated impressive outcomes across several domains, particularly in the biomedical area. Its potential and abilities have also been put to use and evaluated in the successful prognosis of brain tumor detection utilizing MRI scans, and the results have been impressive. This study work's primary goal is to provide a thorough critical examination of the studies and research that have previously been conducted to identify and categorize brain tumors using MRI scans. Researchers that specialize in deep learning and have an interest in the field would particularly benefit from this work. First, a quick overview of previous studies employing Deep Learning for brain tumor identification and classification is done. Subsequently, a Table-based critical study of the Deep Learning methods suggested in these research publications (2015–2020) is being conducted. The conclusion, at last, emphasizes the benefits and drawbacks of deep neural networks.

.

1. **Amin, Javaria, et al. "Brain tumor detection and classification using machine learning: a comprehensive survey." Complex & intelligent systems 8.4 (2022): 31613183.**

Cells develop quickly and uncontrollably, which can lead to brain tumors. It might be fatal if treatment is not received in the early stages. Accurate segmentation and classification remain a difficult issue in this sector, despite several noteworthy attempts and promising results. The differences in tumor size, form, and location present a significant obstacle to the identification of brain tumors. This survey aims to provide researchers with a thorough literature review on magnetic resonance imaging (MRI)-based brain tumor detection. This assessment addressed deep learning, transfer learning, and quantum machine learning for the analysis of brain tumors, as well as the anatomy of brain tumors, publically accessible datasets, augmentation approaches, segmentation, feature extraction, and classification.

1. **Grampurohit, Sneha, et al. "Brain tumor detection using deep learning models."**

**2020 IEEE India Council International Subsections Conference (INDISCON). IEEE, 2020.**

An illness known as a brain tumor is brought on by the proliferation of aberrant brain cells. Brain tumors fall into two primary categories: benign (non-cancerous) brain tumors and malignant (cancerous) brain tumors. Because brain tumors are rare and come in a variety of forms, it is challenging to estimate the survival rate of a patient who is tumor-prone. According to UK cancer study, 15 out of every 100 patients with brain cancer will be able to live for ten years or more following their diagnosis. The type of brain tumor, the abnormality of the cells, the location of the tumor in the brain, and other factors all affect how the tumor is treated. In order to identify the tumor location in the scanned brain pictures, the study effort is done using deep learning models such as convolutional neural network (CNN) model and VGG-16 architecture (made from scratch). We have analyzed brain MRI pictures from 253 patients, 155 of whom have tumors and 98 of which do not. The results of the CNN model and the VGG-16 architecture are compared in this research.

**11.Khan, Md Saikat Islam, et al. "Accurate brain tumor detection using deep convolutional neural network." Computational and Structural Biotechnology Journal 20 (2022): 4733-4745.**

Understanding the mechanism of a brain tumor better requires first detecting and classifying it. An innovative medical imaging method called Magnetic Reasoning Imaging (MRI) aids radiologists in locating the tumor site. Manually testing the MRI pictures is a laborious process that needs experience. The development of deep learning, machine learning, and computer-assisted diagnosis (CAD) has made it possible for radiologists to more accurately diagnose brain tumors these days. In order to solve this difficulty, the conventional machine learning techniques call for a manually created feature for categorization. On the other hand, deep learning techniques may be created to produce accurate classification results without the need for manual feature extraction. In order to distinguish between binary (normal and pathological) and multiclass (meningioma, glioma, and pituitary) brain tumors, this research presents two deep learning models. We make use of two datasets that are freely accessible and contain, respectively, 3064 and 152 MRI pictures. As the first dataset contains a significant number of MRI images for training, we initially use a 23-layer convolution neural network (CNN) to develop our models.

1. **Choudhury, Chirodip Lodh, et al. "Brain tumor detection and classification using convolutional neural network and deep neural network." 2020 international conference on computer science, engineering and applications (ICCSEA). IEEE, 2020.**

It is crucial to identify brain tumors accurately and promptly in order to treat the illness successfully. Early identification has the potential to save lives and aid in the development of better drugs. The development of biomedical informatics and computer-aided diagnosis has several benefits for neuro-oncologists. Recently, machine learning algorithms have been used to evaluate medical images and data instead of manually diagnosing tumors, which is a laborious procedure that is prone to human mistake. When compared to manual, conventional diagnosing procedures, computer-aided methods yield superior outcomes. Typically, a convolutional neural network (CNN) is used to extract features, and a fully connected network is used for classification.

1. **Maqsood, Sarmad, Robertas Damaševičius, and Rytis Maskeliūnas. "Multi-modal brain tumor detection using deep neural network and multiclass SVM." Medicina 58.8 (2022): 1090.**

In today’s health system, clinical diagnosis has become increasingly important. Brain cancer, the most dangerous illness and the world's biggest cause of death, is a major area of study for medical imaging researchers. An accurate and timely diagnosis based on magnetic resonance imaging can enhance the assessment and prognosis of brain malignancies. Medical imaging needs to be identified, separated, and categorized in order for computer-aided diagnostic techniques to help radiologists correctly identify brain tumors. Radiologists find that diagnosing brain tumors manually is a tedious and error-prone process, therefore it's critical to have an automated system in place. Consequently, the accurate technique for detecting and classifying brain tumors is introduced.

1. **Methil, Aryan Sagar. "Brain tumor detection using deep learning and image processing." 2021 international conference on artificial intelligence and smart systems (ICAIS). IEEE, 2021.**

One of the most challenging issues in medical image processing is the detection of brain tumors. Because brain tumors can have a variety of forms and textures, the identification task is challenging to complete due to the visual diversity. Different types of cells are the source of brain tumors, and the cells can provide information on the tumor's origin, severity, and rarity. Tumors may appear anywhere, and their location might reveal information about the kind of cells that are creating them, which can help with additional diagnosis. Issues with lighting, for example, that arise in nearly all digital photographs might exacerbate the issue of detecting brain tumors. This study presents a unique approach to identify brain cancers from a variety of brain pictures by first using several image preprocessing techniques, such as opening and histogram equalization, and then using a convolutional neural network. The influence of other picture preprocessing methods on our dataset, in addition to those that are completed for training, are also covered in the study. The dataset used in the experimental investigation included tumors of various sizes, forms, textures, and locations.

1. **Mohsen, Heba, et al. "Classification using deep learning neural networks for brain**

**tumors." Future Computing and Informatics Journal 3.1 (2018): 68-71.**

Over the past several years, there has been a lot of interest in the emerging topic of deep learning in machine learning. It was extensively used in numerous applications and showed to be an effective machine learning technique for solving a lot of challenging issues.

In this study, we classified a dataset of 66 brain MRIs into four classes—normal, glioblastoma, sarcoma, and metastatic bronchogenic carcinoma tumors—using a Deep Neural Network classifier, one of the DL architectures. Principal components analysis (PCA) and the discrete wavelet transform (DWT), a potent method for extracting features, were integrated with the classifier, and the evaluation of the performance was rather excellent across all performance criteria.

With billions of cells, the brain is one of the body's most intricate organs. Uncontrolled cell division that results in an aberrant collection of cells surrounding or inside the brain is the cause of a brain tumor. That cell type has the ability to damage healthy cells and interfere with brain activity that occurs normally. Brain tumors are categorized as malignant or highgrade (grade III and IV) and benign or low-grade (grade I and II). Benign tumors begin in the brain, develop slowly, and cannot spread to other parts of the body. Because they are nonprogressive (non-cancerous), they are regarded as less aggressive. Malignant tumors, on the other hand, are cancerous and develop quickly with no clear borders.

**CHAPTER-3**

**SYSTEM ANALYSIS**

**3. SYSTEM ANALYSIS**

An in-depth examination of project data using a variety of phases, methods, functions, and entities constitutes the analysis of computer data, project data, algorithm data, and other inner and outer data relevant to the proposed study. System analysis is a collection of scientific methods for figuring out the specifications for project task design. For the design of the suggested system, system analysis examined a variety of functional and non-functional requirements. In order to create a logical model of the system, the current system analysis has examined numerous publications pertinent to the project's work and planned the design using a variety of tools, including class diagrams, sequence diagrams, data flow diagrams, and data dictionaries.

# 3.1 EXISTING SYSTEM:

The current method for detecting brain tumors mostly depends on radiologists manually interpreting medical imaging, which can be laborious and prone to human error. Conventional techniques like MRIs and CT scans offer useful diagnostic data, but proper interpretation frequently necessitates specialist knowledge. Tumor identification has been aided by automated systems that use machine learning techniques, although these systems might not be accurate or efficient enough for practical clinical use.

## 3.1.1 DISADVANTAGES:

The current methods for detecting brain tumors have a number of drawbacks, including their reliance on human interpretation of medical imaging, which can result in mistakes, discrepancies, and delays in diagnosis. Human interpretation is arbitrary and prone to variation depending on the radiologist's training and experience. This might lead to an incorrect diagnosis or a postponed start of therapy. Furthermore, using traditional methods might take a while, especially if more than one photograph has to be examined or a specialist's advice is needed. Furthermore, non-specialist healthcare practitioners may find it difficult to interpret imaging data due to their complexity, which might cause additional delays or incorrect diagnoses. All things considered, the shortcomings of the current system underscore the necessity for more sophisticated and automated methods that enhance diagnostic precision, effectiveness, and accessibility.

# 3.2 PROPOSED SYSTEM:

The suggested brain tumor detection system uses Convolutional Neural Networks (CNNs) in conjunction with an intuitive Streamlit user interface (UI) to address the shortcomings of current techniques. By automatically and extremely accurately classifying brain scans into several tumor classifications, this cutting-edge device gives medical professionals quick access to precise diagnostic data. The suggested method improves accessibility for a larger variety of users, decreases the need for human interpretation, and speeds the detection process by fusing cutting-edge technology with an easy-to-use user interface.

## 3.2.1 ADVANTAGES:

When compared to current techniques for brain tumor identification, the suggested approach has a number of advantages. Initially, the utilization of Convolutional Neural Networks (CNNs) considerably enhances the precision and effectiveness of tumor categorization. Because CNNs are good at deciphering complex patterns and characteristics in medical pictures, the system can accurately identify tiny anomalies that may be cancers. This improves diagnostic precision and lowers the possibility of a false positive, which results in more efficient treatment planning and improved patient outcomes. Second, the addition of an intuitive Streamlit user interface (UI) improves usability and accessibility for both patients and healthcare professionals. Users may engage with the system with ease thanks to the user-friendly interface design. They can input brain photos and quickly receive categorization results. This expedites the diagnostic procedure, cutting down on the time and resources needed to find tumors and enabling the start of therapy on schedule.

# 3.3 CONVOLUTIONAL NEURAL NETWORKS

A family of deep learning models called Convolutional Neural Networks (CNNs) is especially made for processing and evaluating visual input, such pictures and movies. CNNs are widely used and have become the dominant solution for a wide range of computer vision applications because of their automated learning of hierarchical representations of visual characteristics from raw input data. The convolutional layer, which applies a collection of learnable filters, or kernels, to the input picture, is one of the essential parts of CNNs. These filters extract local characteristics like edges, textures, and patterns by convolving over the picture.

Additionally, CNNs usually include pooling layers to down-sample the feature maps that are produced from the convolutional layers, such as average or max pooling. By retaining the most important information while lowering the spatial dimensionality of the feature maps, pooling layers improve the network's computational efficiency and resilience to changes in the input data. Furthermore, fully connected layers are frequently seen at the conclusion of CNN designs. These layers use the high-level feature representations that the convolutional layers have taught them to carry out tasks like regression or classification. In classification tasks, these fully connected layers are usually followed by softmax activation functions to provide probability distributions over output classes. Stochastic gradient descent (SGD) and its variants are optimization algorithms used in CNN training that iteratively modify the network's parameters (like filter weights and biases) in order to minimize a predefined loss function, like categorical cross-entropy loss in classification tasks. Through the process of backpropagation, in which gradients are produced and transmitted backward through the network to change the parameters, CNNs are trained to automatically extract discriminative features from the input data and generalize to unseen cases.

# 3.4 MODULES USED IN PROPOSED SYSTEM

## 3.4.1 USER:

* **View Home page:** Here user view the home page of the Classifications web application.
* **View Upload page:** In the about page, users can learn more about the prediction.
* **Input Model:** The user must provide input values for the certain fields in order to get results.
* **View Results:** User view’s the generated results from the model.
* **View score:** Here user have ability to view the score in %

## 3.4.2 SYSTEM

* **Working on dataset:** System checks for data whether it is available or not and load the image files.
* **Pre-processing:** Data need to be pre-processed according the models it helps to increase the accuracy of the model and better information about the data.
* **Training the data:** After pre-processing the data will split into two parts as train and test data before training with the given algorithms.
* **Model Building:** To create a model that predicts the personality with better accuracy, this module will help user.
* **Generated Score:** Here user view the score in %
* **Generate Results:** We train the machine learning algorithm and predict the hate speech.

## 3.4.3 ALGORITHMS USED

**3.4.3.1 CNN**

A particular class of machine learning model known as a convolutional neural network (CNN) is a deep learning technique that is particularly well-suited for the analysis of visual input. CNNs, also known as convnets, extract features and recognize patterns in pictures using concepts from linear algebra, namely convolution processes. CNNs may be configured to handle audio and other signal data, even if processing pictures is their primary function. The connection patterns seen in the human brain, particularly in the visual cortex, which is crucial for the perception and processing of visual inputs, served as the model for CNN design. These models can comprehend whole pictures because the artificial neurons in a CNN are constructed to efficiently interpret visual information. CNNs are widely used for computer vision tasks like picture identification and object detection because they are so good at detecting things. Common use cases include medical image analysis, self-driving automobiles, and facial recognition.

**Convolutional Operation:** Learnable filters are applied to input pictures during the convolutional operation, which is the foundation of CNNs. Every filter, which is sometimes referred to as a kernel, convolves across the input picture, calculating dot products between local parts of the image and the filter weights. Local characteristics including edges, textures, and patterns are extracted using this approach. CNNs may capture hierarchical information in the input data by learning increasingly abstract and complicated representations of visual characteristics through the stacking of many convolutional layers.

**Pooling Operation:** In CNN topologies, pooling layers are often added after convolutional layers. The most notable characteristics are retained in feature maps while their spatial dimensionality is decreased using pooling procedures like max pooling and average pooling. Each feature map region is combined into a single value during the pooling process, usually by picking the maximum or average value within the region. Pooling lowers the computational complexity of succeeding layers and helps to make the representations more invariant to tiny spatial translations.

**Activation Operation:** By adding non-linearity to the CNN design, activation functions enable the network to simulate intricate feature interactions. Convolutional layers frequently employ the Rectified Linear Unit (ReLU) activation function because of its efficiency and ease of usage. ReLU introduces non-linearity into the network by setting negative values to zero and leaving positive values unaltered. This allows the network to learn more complicated functions. Other activation functions, including tanh and sigmoid, could be employed at the output layer of classification tasks or in certain conditions.

**Fully Connected Operation:** To carry out classification or regression tasks, fully connected layers combine the high-level feature representations that were taught to them by the convolutional layers. These layers have a dense connection pattern because every neuron in the layer above is linked to every other neuron. In a network design, fully linked layers usually come after convolutional and pooling layers. In order to enable the network to generate predictions based on the extracted features, they are frequently utilized in the last layers of the network to translate the learnt features to the output classes or regression values.

**Softmax Operation:** To generate probability distributions over the output classes in classification tasks, the output of fully connected layers is frequently subjected to the softmax operation. In order to guarantee that the projected probabilities add up to one, Softmax normalizes the output scores into a probability distribution. As a result, given the input data, the network can now output the probability of each class. Softmax is very helpful when the network has to give a probability to every potential class label in a multi-class classification problem.

**CHAPTER-4**

**SYSTEM REQUIREMENTS SPECIFICATION**

**SYSTEM REQUIREMENTS SPECIFICATION**

Software requirements specifications (SRS), also known as software system requirements specifications, offer a comprehensive description of the duties that a system must do. The use cases in this section describe how the software interacts with its users. The SRS also contains non-functional specifications in addition to the usage case. Non-functional specifications are criteria that limit design or execution (such as requirements for performance engineering, quality standards or design constraints).

### 4.1. SOFTWARE REQUIREMENTS

|  |  |  |  |
| --- | --- | --- | --- |
| • | Operating System | : | Windows 11 |
| • | Serverside Script | : | Python |
| • | IDE | : | Visual Studio Code |
| • | Framework | : | Streamlit |
| • | Dataset | : | Images Dataset |

### 4.2. HARDWARE REQUIREMENTS

|  |  |  |  |
| --- | --- | --- | --- |
| • | Processor | : | I3/Intel Processor |
| • | RAM | : | 4GB (min) |
| • | Hard Disk | : | 160GB |
| • | Key Board | : | Standard Windows Keyboard |
| • | Monitor | : | SVGA |

### 4.3. FEASIBILITY STUDY

Finding the optimum solution to meet performance requirements is the goal of a feasibility study. They include a description of identification, an assessment of potential system candidates, and the choice of the best candidate.

* Economic Feasibility
* Technical Feasibility
* Behavioral Feasibility

**4.3.1. Economic Feasibility:**

The most popular way for determining whether a potential system is effective is economic analysis. The process, more popularly known as cost/benefit analysis, entails calculating savings and benefits to see if they outweigh expenses. If they do, the decision to design and execute the system is then made. If the system is to have an enhancement that can be approved, more justification or changes must be made.

**4.3.2. Technical Feasibility:**

The existing computer system's capabilities to accommodate the planned expansion are the focus of the technical analysis (hardware, software, etc.). To allow technical advancement, there must be financial concerns. The project is deemed unfeasible if funding is a severe restriction.

**4.3.3. Behavioral Feasibility**:

The strength of the user staff's expected opposition to the creation of a computerised system should be estimated. The introduction of a potential system necessitates extra effort to inform, persuade, and train the current methods of thinking about business. It is well known that computer installations have something to do with understanding.

**4.3.4. Benefits of Doing a Feasibility Study:**

The following list summarises some of the benefits of doing a feasibility study.

* The analysis portion of this study, which is being created as the first stage of the software development life cycle, assists in thoroughly examining the system requirements.
* Aids in determining the risk variables associated in creating and implementing the system.
* Planning for risk analysis is aided by the feasibility study.
* Cost-benefit analyses made possible by feasibility studies enable effective operation of the system and organisation.
* Planning for training developers to put the system into place is aided by feasibility studies.

### 4.4. FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

Analysis of requirements is a vital step in determining whether a system or software project will be successful. Functional requirements and non-functional requirements are the two main categories of requirements.

**4.4.1. Functional Requirements:**

These are the necessities that the system must provide in order to meet the end user's individual requests for basic amenities. The contract must unavoidably stipulate that each of these functionalities be built into the system. They are portrayed or described as input to be provided to the system, an operation to be carried out, and an anticipated output. Unlike non-functional needs, they are essentially the user-stated requirements that can be seen immediately in the finished product.

An illustration of a functional requirement is:

1. Whenever a user logs into the system, they must authenticate themselves.
2. In the event of a cyberattack, shut down the system.
3. When a user registers for the first time on a software system, a verification email is automatically sent to them.

**4.4.2. Non-functional requirements**

In essence, they are the quality requirements that the system must meet in accordance with the project contract. Depending on the project, different aspects may be given varying degrees of priority or implementation. These are also known as non-behavioral requirements.

They primarily address things like: Portability, Security, Maintainability, Reliability, Scalability, Performance, Reusability, Flexibility.

Non-functional needs examples include:

1. With respect to such an activity, emails should be sent no more than 12 hours afterwards.
2. Each request should be processed in less than ten seconds.

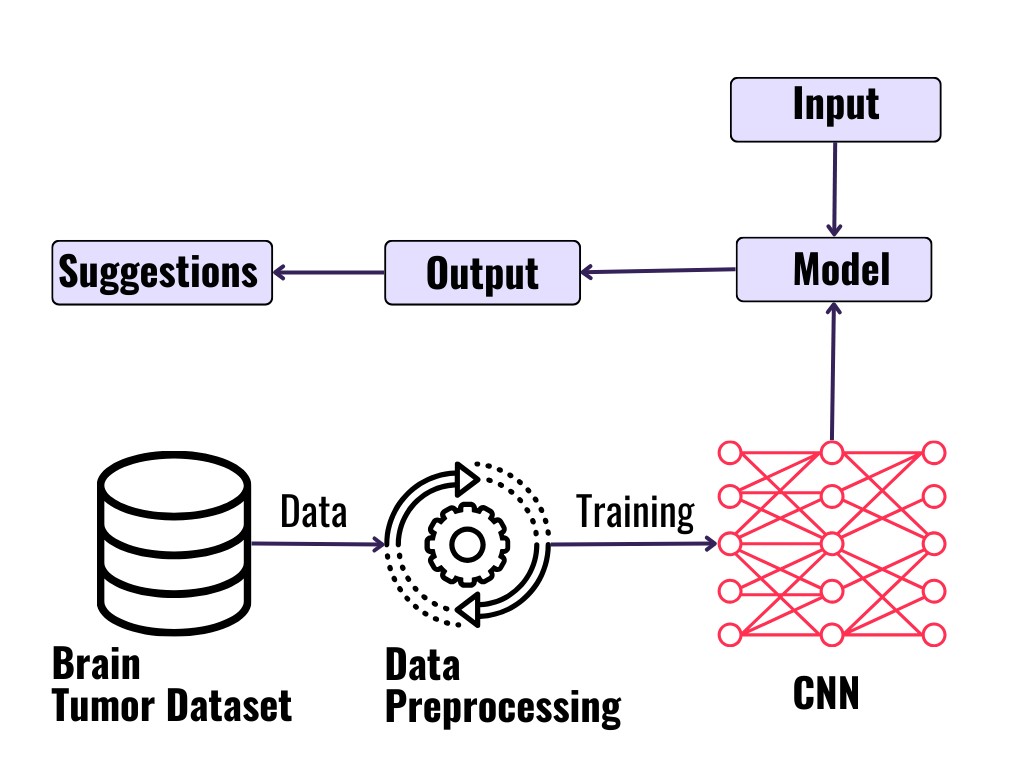
1. If there are more than 10,000 simultaneous users, the website should load in 3 seconds.

**CHAPTER-5**

**SYSTEM DESIGN**

**5. SYSTEM DESIGN**

### 5.1. ARCHITECTURE DESIGN



#### Figure 5.1: Architecture diagram

The process flow of a convolutional neural network (CNN)-based brain tumor detection system is depicted in the diagram. The first step is to use a Brain Tumor Dataset as the primary source of information for training the CNN model. Preprocessing is done on the data to improve its training appropriateness and guarantee best-in-class model performance. The preprocessed data is then used to train CNN, a deep learning architecture created especially for image identification applications. The CNN gains the ability to identify pertinent features from the input pictures and categorize them into the proper groups during training. After being trained, the CNN model can evaluate fresh input data, like pictures of the brain, and produce predictions about the existence or absence of tumors.

### 5.2. INTRODUCTION TO UML DIAGRAMS

As the strategic importance of software grows, the industry searches for ways to

automate software development, enhance quality, cut costs, and accelerate time-to-market. Component technology, visual programming, patterns, and frameworks are a few examples of these techniques. When a company grows, it searches for ways to control the scope and size of its systems. reduce their complexity. The issues with load balancing, fault tolerance, concurrency, replication, and physical distribution are all issues they are aware of. The Internet has also made many structural problems worse while simplifying some tasks. The Unified Modeling Language (UML) was created to satisfy these requirements. Simply described, systems design is the process of creating a system's architecture, components, modules, interfaces, and data to meet certain goals. This can be accomplished fast using UML diagrams.

Throughout the project, eight fundamental UML diagrams were explained.

* Use Case Diagram
* Class Diagram
* Activity Diagram
* Sequence Diagram
* Collaboration Diagram
* State chart Diagram
* Component Diagram
* Deployment Diagram

**5.2.1 GOALS**

1. Make available to users a ready-to-use, expressive visual modeling language that enables them to create and share meaningful models.

1. Provide mechanisms for extendibility and specialisation in order to broaden the scope of the core concepts.

1. Refrain from using specific programming languages or development processes.

1. Lay the groundwork for a formal understanding of the modeling language.

1. The following are the primary goals of the UML design:
2. Encourage the growth of the market for OO tools.

1. Help with the implementation of higher-level development concepts like collaborations, frameworks, patterns, and components.

1. Implement best practices**.**

### 5.3. UML NOTATIONS

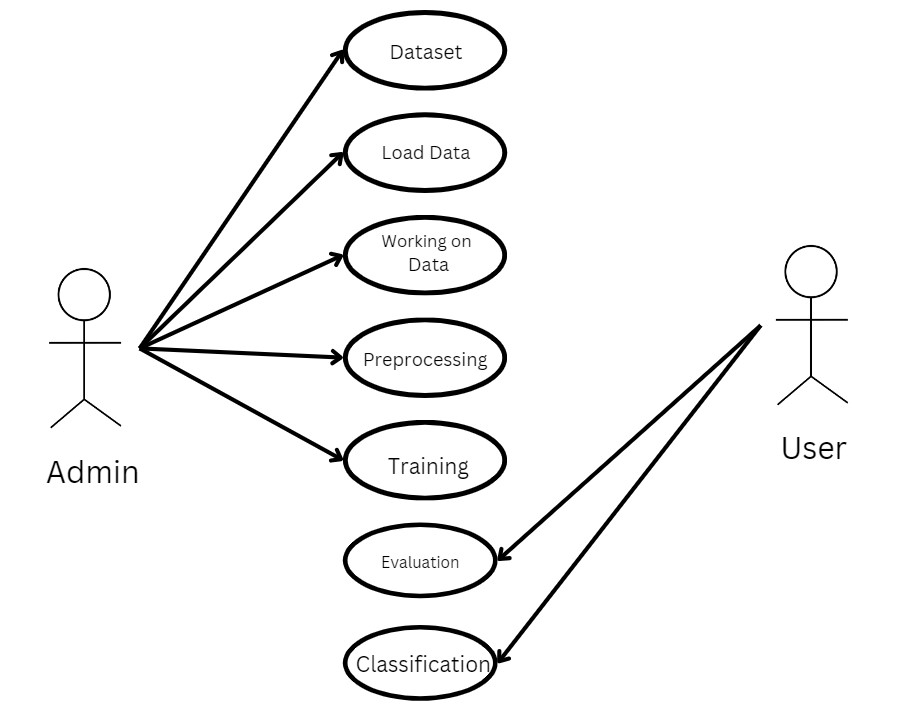
|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **SYMBOL**  **NAME** | **NOTATION** | **DESCRIPTION** |
| 1. | Initial Activity |  | This diagram depicts the flows initial point or activity. |
| 2. | Final Activity |  | A bull’s eye icon  marks the conclusion of the activity graphic. |
| 3. | Activity |  | Represented by a rectangle with a rounded edge. |
| 4. | Decision |  | One that requires decision-making. |
| 5. | Use Case |  | Explain how a user and a system communicate. |
| 6. | Actor |  | A function a user has in relation to the system. |

|  |  |  |  |
| --- | --- | --- | --- |
| 7. | Object |  | A Real -Time entity. |
| 8. | Message |  | To communicate between the lives of object. |
| 9. | State |  | It depicts events that occur during an objects lifetime. |
| 10. | Initial State |  | Represents the objects initial state. |
| 11. | Final State |  | Represents the objects final state. |
| 12. | Transition |  | Label the transition with the event that triggered it and the action that result from it . |
| 13. | Class |  | A group of items with similar structures and behaviours. |
| 14. | Association |  | Relationship between classes. |
| 15. | Generalization |  | Relationship  between more general class and a more specific class. |

### 5.4. UML DIAGRAMS

**5.4.1. USE CASE DIAGRAM**

A use case diagram is a form of behavioural diagram created from use-case research and is an example of software engineering's use of the Unified Modeling Language (UML). Its goal is to demonstrate the actors, goals (represented as use cases), and any dependencies among those use cases in a system. The main goal of a use case diagram is to show which system functions are executed for each actor. It is clear what the system's actor roles are. Throughout the requirements elicitation and analysis phase, use cases are used to illustrate the capabilities of the system. To describe how the technology works when not in use, use scenarios are utilised. Use cases are inside the system, whereas actors are outside. A device border separates a group of use cases in the case diagram, which is a diagram of actors. The application A diagram is necessary to comprehend the element's behaviour.

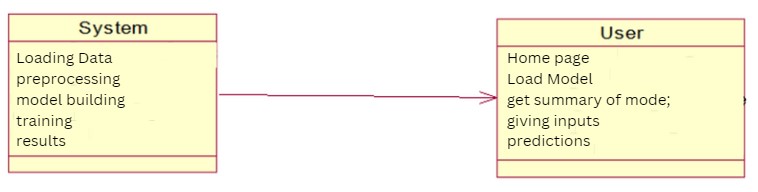
1. Sequences highlight the relationship to outside circumstances.
2. This covers both the performer's job and the system.
3. Actors can portray people or a building. 

#### Figure 5.2: Use Case Diagram

**5.4.2. CLASS DIAGRAM**

A class diagram in the Unified Modeling Language (UML) is a type of static structural diagram used in software engineering to show the classes, properties, and relationships between the classes that make up a system.

It is utilised in analysis to show the system's specifics. Architecture examines the class diagram to determine which classes have an excessive number of functions and, if any do, whether they should be divided. The connections between the classes are made. The Class Diagram is a tool used by developers to create classes. A class diagram is a group of related objects that are all connected and have the same features, operations, relationships, and connections and regulations referred to as semantics. A class is a huge group of items in a production.



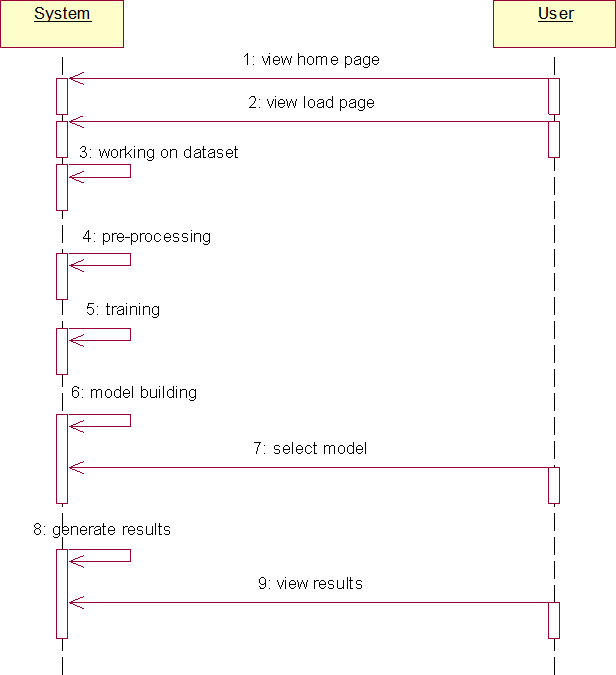
#### Figure 5.3: Class Diagram

In the Unified Modeling language, a class diagram is a type of static structural diagram that displays the functions, interactions, and relationships between objects to depict a system's structure. The cornerstone of object-oriented modeling is the class diagram. Image, build dataset, pre-processing, segmentation, and classification are the classes represented in together with the corresponding properties, processes, and relationships between those classes.

**5.4.3. SEQUENCE DIAGRAM**

In the Unified Modelling Language (UML), a sequence diagram is a type of interaction diagram that illustrates the order and relationship between activities. A message sequence chart is the name given to it. Sequence diagrams include timing diagrams, event-trace diagrams, and representations of event contexts. One can also refer to a sequence diagram as an event diagram or an event scenario. Sequence diagrams show how a system's components interact with one another. The requirements for both new and current systems are frequently described and understood by entrepreneurs and software engineers using these diagrams.

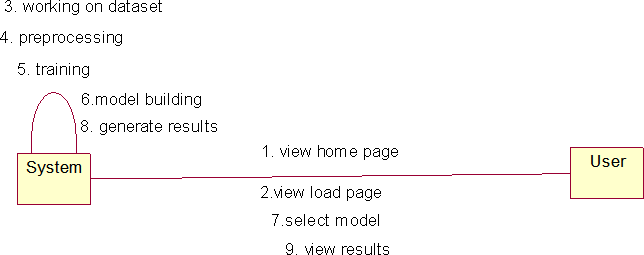
An interaction diagram that emphasises the timing of message delivery. Depending on their lifespan and the messages they transmit or arrange over time, objects taking part in an interaction are represented in a sequence diagram.



#### Figure 5.4: Sequence Diagram

**5.4.4. COLLABORATION DIAGRAM**

The method call sequence in a collaboration diagram is indicated by some numbering technique, as shown below. The number indicates the order in which the methods are called. The collaboration diagram is described using the same order management system. The method calls resemble those of a sequence diagram. The difference is that the sequence diagram does not describe the object organization, whereas the collaboration diagram does.



**Figure 5.5: Collaboration Diagram**

**5.4.5. DEPLOYMENT DIAGRAM**

The hardware and software components that make up a deployment are described using deployment diagrams. Diagrams of components and deployments have a lot in common. Diagrams of the components' deployment in hardware are shown in deployment diagrams, which are used to describe the components.

The software artefacts of a system are the primary emphasis of UML. However, these two particular diagrams are meant to highlight the hardware and software parts. In contrast to deployment diagrams, which are designed to concentrate on a system's hardware topology, most UML diagrams are used to manage logical components. The system engineers utilize diagrams for deployment. You can characterize the function of deployment diagrams as:

1. Think about how a system's hardware is organized.
2. Explain the hardware elements that are deployed in order to run software components.
3. Tell us about the runtime processing nodes.



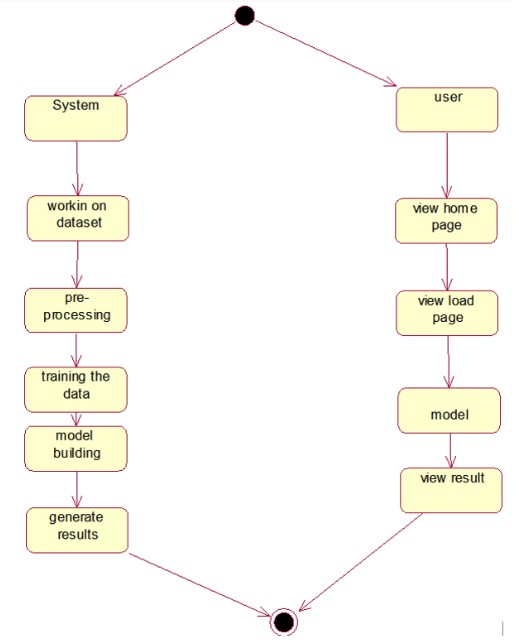
#### Figure 5.6: Deployment Diagram

**5.4.6. ACTIVITY DIAGRAM**

Activity diagrams offer choice, iteration, and concurrency in their depiction of the work flows of evolving tasks and actions. The operational and business processes of system components can be represented in detail using activity flowcharts in the Unified Modified Language.

An activity diagram illustrates the entire control flow. A flowchart with specific states is similar to an activity diagram. With the activity diagram, you can keep track of the sequence of actions occurring in your system. Activities look like states; however, they are a little more rounded. They are stateless because they take place and then go unabatedly to the following state. The "diamond" conditional branch determines which activity to switch to based on a characteristic and is also stateless. Activity Diagram includes

1. Action states.
2. Transition.
3. Objects.
4. Contains Fork, Join and branching relations along with flow Chart symbols.

 .

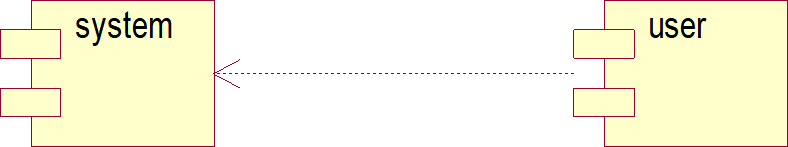
**Figure 5.7: Activity Diagram**

**5.4.7. COMPONENT DIAGRAM**

A specific type of diagram in UML is called a component diagram. The goal is also distinct from the previous diagrams mentioned. Although it defines the components utilized to provide certain functionalities, it does not describe the system's functionality as a whole.

Component diagrams are used to represent the physical parts of a system from that perspective. These parts include files, libraries, and other things. A static implementation view of a system is another way to explain component diagrams. The arrangement of the components at a specific time is represented by static implementation. The entire system cannot be represented by a single component diagram; instead, a collection of diagrams is employed. The component diagram's goal can be summed up as follows:

1. Identify the parts of a system visually.
2. Use both forward and reverse engineering to create executables. 3. Explain how the components are arranged and their connections.

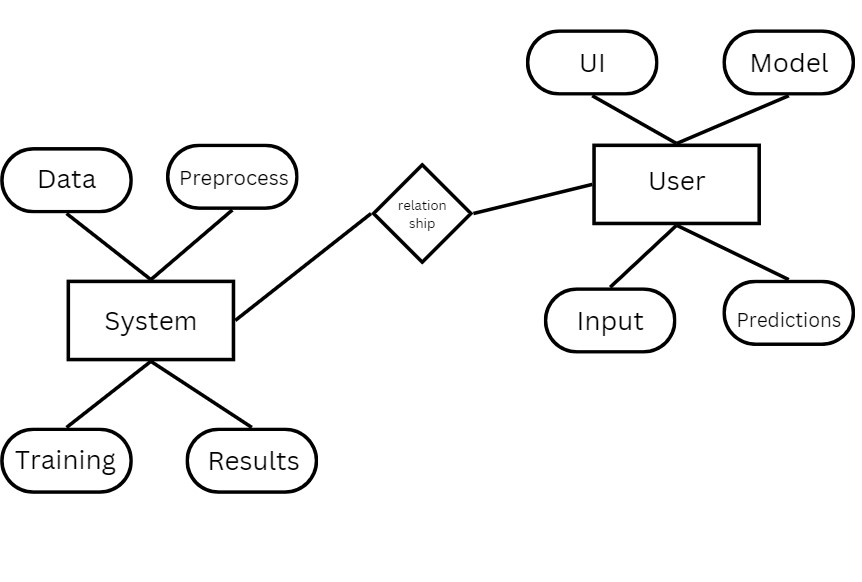


#### Figure 5.8: Component Diagram

#### 5.5. ER DIAGRAM:

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let’s have a look at a simple ER diagram to understand this concept.



**Figure 5.9: ER Diagram**

**CHAPTER-6**

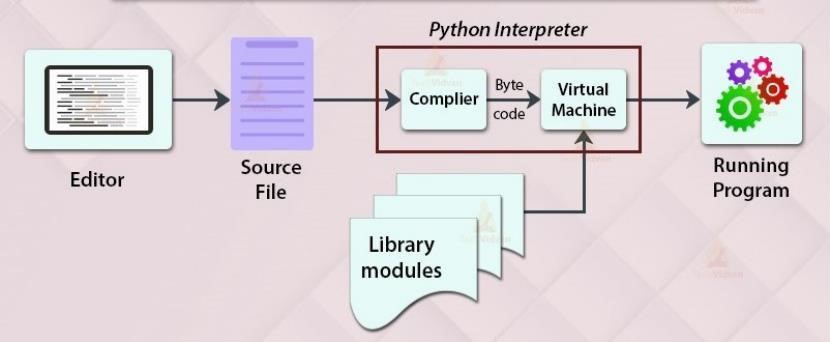
**SYSTEM CODING AND IMPLEMENTATION**

1. **SYSTEM CODING AND IMPLEMENTATION**

**6.1 Introduction to python programming language:**

Popular computer language Python is renowned for its simple syntax, scalability, and application in artificial intelligence (AI) and machine learning. Python is a key component of the systems used by some of the biggest companies in the world, including Google, NASA, and Facebook. Python is an object-oriented programming language. By definition, it is a high- level programming language, enabling the construction of both simple and complex processes. **Features:**

* Simple
* Easy
* Portable
* Object oriented
* High Level
* Open Source and Free
* Support for GUI
* Interpreted
* Dynamic
* Readable
* Extendable
* Scalable

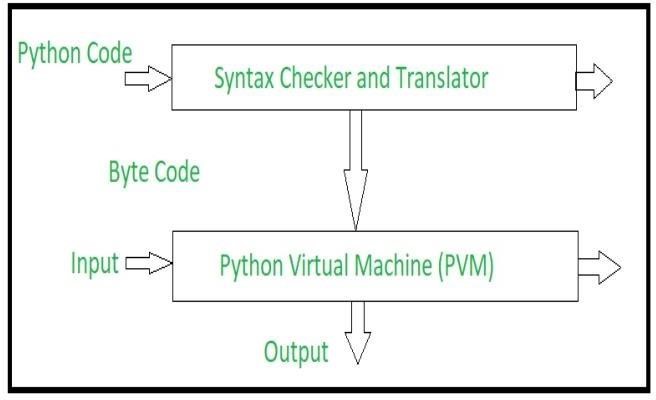


**Figure 6.1 working of python program**

Similar to Java, Python is an object-oriented programming language. The term

"interpreted language" refers to Python. Instead of a single lengthy list of instructions, which was the norm for functional programming languages, Python uses interchangeable code modules. "cpython" is the name of the preferred Python implementation. It serves as Python's default and most used implementation.

Python does not translate its code into hardware-understandable machine code. In actuality, it transforms it into byte code. Compilation does take place in Python, but not into a machine language. It is encoded in byte code (.pyc or.pyo), which the CPU is unable to understand. To execute the byte codes, we therefore require an interpreter known as the Python virtual machine.



**Figure 6.2 Implementation of python program**

**Python source code goes through the following to generate an executable code : Step 1:** The python compiler reads a python source code or instruction. Then it verifies that the instruction is well-formatted, i.e. it checks the syntax of each line. If it encounters an error, it immediately halts the translation and shows an error message.

**Step 2:** If there is no error, i.e. if the python instruction or source code is well-formatted then the compiler translates it into its equivalent form in an intermediate language called “Byte code”.

**Step 3:** Byte code is then sent to the Python Virtual Machine(PVM) which is the python interpreter. PVM converts the python byte code into machine-executable code. If an error occurs during this interpretation then the conversion is halted with an error message.

**6.1.1 BENEFITS OF PYTHON**

1. Python may be used to create prototypes, and because it is so simple to use and read, it can be done rapidly.
2. The majority of platforms for automation, data mining, and big data rely on Python. 3) Compared to large languages like C# and Java, Python offers a more productive coding environment. By using Python, seasoned programmers tend to stay more organised and productive.
3. Even if you're not an experienced programmer, Python is simple to read. Everyone can start using the language; all it needs is some perseverance and lots of practise. Additionally, this makes it a perfect choice for use by large development teams and teams with multiple programmers.
4. Django is a full and open-source web application framework that is powered by Python. The process of developing software can be made simpler by using frameworks like Ruby on Rails. 6) Because it was built by the community and is open source, it has a huge fan base. Millions of like-minded programmers use the language every day and keep its foundational features up to date. As time goes on, Python's most recent version continues to get updates and improvements. This is a fantastic method of connecting with other developers.

**6.2 Libraries used in Python**

**Pandas** :is a well-known Python open-source data science, data analysis, and machine learning toolbox. NumPy, another Python module that supports multidimensional arrays, was used to build it.

**NumPy:** A Python module called Numpy is used to work with arrays. Matrixes, Additionally provided are linear algebraic functions and Fourier transformations.

**Matplotlib:** The Python numerical mathematics extension NumPy is compatible with Matplotlib, a graphing software written in Python. For integrating charts into graphical user interface toolkits like Tkinter, wxPython, and Qt, it offers an object-oriented API.

**Sklearn:** Scikit-learn, formerly known as scikits.learn, is a free-to-use Python machine learning library. Featured classification, regression, and clustering methods include support- vector machines, random forests, gradient boosting, k-means, and DBSCAN.

**TensorFlow :** Is a Python library that Google created and released for quick numerical computation. It serves as the basis for building Deep Learning models, either directly or indirectly through wrapper libraries built on top of TensorFlow to facilitate the process.

**Pycharm**

PyCharm is a popular Python IDE. There are several reasons for this, one of which is that it was created by JetBrains, the company that also created the well-known IntelliJ IDEA IDE, one of the "big 3" Java IDEs, and WebStorm, the "smartest JavaScript IDE."Another compelling reason is the availability of Django support for web development.

Pycharm created this IDE primarily for Python programming and to run on various operating systems such as Windows, Linux, and macOS. The IDE includes version control, a debugger, testing tools, and code analysis tools. It also assists programmers in creating Python plugins by leveraging the numerous APIs available.The IDE allows us to work directly with a variety of databases without having to integrate them with other programmes. Despite being designed specifically for Python, this IDE also supports HTML, CSS, and other markup languages.Javascript documents. It also has a stunning user interface that can be customised using plugins to meet specific needs.

**6.3. Code**

**#app.py**

ADD YOUR MODEL CODE

**Ui.py**

ADD YOUR UI CODE

**CHAPTER-7**

**SYSTEM TESTING**

1. **SYSTEM TESTING**

**7.1. SOFTWARE TESTING TECHNIQUES**

Software testing is a method for evaluating the quality of software products and identifying defects so that they can be rectified. Software testing makes an effort to accomplish its goals, but there are significant constraints. On the other side, for testing to be effective, dedication to the set objectives is required.

### 7.1.1. Testing Objectives

1. The user stories, designs, specifications, and code that make up the work products
2. To ensure that all conditions are satisfied.
3. Ensuring that the test object is complete and meets the expectations of users and stakeholders

**7.1.2. Test Case Design**

Every engineering product can be tested in one of these.

### 7.1.3. TESTING OF A WHITE BOX

Black box testing and white box testing are two types of software testing methodologies. White Box testing, also known as structural testing, clear box testing, open box testing, and transparent box testing, is covered in this article. It focuses on evaluating the infrastructure and software's fundamental code against current inputs and anticipated and desired outcomes. It emphasises internal structure analysis and is focused on a program's internal activities. To construct test cases for this type of testing, programming knowledge is needed. Focusing on the inputs and outputs of the software while also ensuring its security is the core aim of white box testing. The phrases "clear box," "white box," and "transparent box" all allude to being able to see through the exterior covering of the software. White testing a box is used by designers. This stage involves testing every line of the program's code. Prior to handing off the programme or software to the testing team, the developers run white-box testing on it to ensure that it conforms with the requirements and to identify any mistakes.

Before releasing the project to the testing team, the developer fixes the issues and does one round of white box testing. In this case, fixing problems includes removing the problem and activating the specific functionality of the application. For the following reasons, the test engineers won't be helping to fix the problems: o Resolving the problem might impair other features. As a result, developers should keep making advancements while the test engineer should constantly look for faults. If the test engineers spend most of their time fixing problems, they might not be able to find any new flaws in the program.

The following tests are part of the white box testing:

* Path testing
* Loop testing
* Condition evaluation
* Testing from the viewpoint of memory
* Test results for the program

### 7.1.4. BLACK BOX TESTING

Testing software applications' functionalities without having access to the internal code structure, implementation details, or internal paths is known as "black box testing" in the software industry. The term "black box testing" refers to a sort of software testing that is solely concerned with the input and output of software programm as well as the requirements and specifications for software. You are free to use any software package you choose as a Black- Box. A few examples include an Oracle database, a Google website, the Windows operating system, or even your own custom program. You can test these applications using black box testing by focusing just on their inputs and outputs and ignoring any awareness of how their underlying code is implemented.

The following locations are checked for faults by this technique:

1. Insufficiency or absence of capacities.
2. Interaction errors.
3. Inadequate information architecture.
4. Behaviour or execution errors.
5. Starting and finishing errors.

**7.2. STRATEGIES FOR SOFTWARE TESTING**

* A unit test
* Integrity Checks
* Validation Examination
* System Evaluation
* Security Checks
* Performance Evaluation

### 7.2.1. Unit Testing

The module is the smallest piece of software architecture that is tested as part of unit testing. Within the constraints of the module, significant control channels are analysed using the procedural design description as a guide. The smallest testable parts of a programme, called units, are reviewed separately and independently during unit testing to guarantee proper operation. This testing process is used by software engineers and, on occasion, QA staff throughout the development phase. The main objective of unit testing is to test and validate written code separately to ensure that it operates as intended.

When done correctly, unit testing can help detect coding flaws that would otherwise be difficult to locate. TDD is a practical technique that regularly tests and enhances the product development process in a complete manner. One of the elements of TDD is unit testing. This method of testing serves as the initial phase of software testing and includes tests that come before integration testing and other types of testing. Unit testing verifies a unit's independence from any external code or functionalities. Manual testing is still an option even if automation testing is more popular.

### 7.2.2. Integration Testing

Integration testing is the process of creating a program's structure while running tests to find interface problems. To create a design-based programme structure, unit-tested methods are to be used. Integration testing is a testing procedure that conceptually connects and puts software components to the test. A typical software project consists of several software modules created by different programmers. This level of testing aims to find problems in the interactions between different software components when they are integrated. The interactions between these modules are examined during integration testing.

**Top-Down Integration:**

The next step in the testing process is top-down integrations, a method for building and testing a program's structure progressively. Different modules in a software, product, or application are integrated by moving downward through the systematic control hierarchy between the modules, starting with the main control or home control or index program. The project's framework includes a variety of breadth- or depth-first activities or modules related to the primary program.

**Bottom-up Integration:**

The construction and testing of a few atomic modules, or the product's most basic features, is the first step in the subsequent testing methodology. Since all processes or modules are integrated bottom-up, there is no need for residual, and processing for modules tied to a certain level is always available.

### 7.2.3. Validation testing

Validation testing assures that the software developed and tested satisfies the client's or user's needs. Logic or scenarios for business requirements need to be thoroughly tested. Here, it is necessary to test every significant component of the application. You must always be able to validate the business logic or scenarios that are given to you as a tester. One such method that encourages a careful examination of functioning is the validation process.

Validation testing ensures that the programme has been tested and built to meet user or customer requirements. The justifications or scenarios for business demands must be thoroughly tested. Every key component of the application must be tested in this situation. As a tester, you will always be provided with scenarios or business logic that can be independently checked. One such process that helps in a detailed analysis of performance is the validation process.

### 7.2.4. System Testing

System testing's main goal is to rigorously test computer-based systems. Even though each test has a distinct goal, they all check to make sure that each system part is properly integrated in order to reach the objectives. Examining an entirely integrated software system is a component of system testing. A computer system is typically constructed by mixing software (any Software is the sole component of a computer system. The program is made up of modules that, when placed together with other pieces of software and hardware, form a complete computer system. In other words, a computer system is made up of numerous software program that perform various jobs. Software, however, is unable to carry out these duties alone.

System Testing requires the appropriate hardware must be used to help. System testing is a set of processes used to verify the overall functionality of a computer system that uses integrated software. The practise of system testing involves examining an application's or software's end-to-end flow from the viewpoint of a user. Each module required for an application is examined in detail, and systemic product testing is done to ensure that the final features and functionality function as planned. Since the testing environment mirrors the production environment, it is known as end-to-end testing.

### 7.2.5. Security testing

Security testing is an essential component of software testing since it enables us to identify vulnerabilities, risks, and hazards in software applications and protects our program from malevolent outsiders. Security testing's primary objective is to identify all of a program's potential ambiguities and vulnerabilities, which maintains the application operating. When we perform security testing, we might uncover any potential security risks and assist the programmer in resolving any issues. It is a method for ensuring data security while preserving software usability.

### 7.2.6. Performance Evaluation

Performance testing is a technique for assessing a system's responsiveness and stability under changing workloads. Performance testing assesses the dependability, scalability, and resource use of the system.

**Performance Evaluation Method:**

Load testing is the simplest technique for evaluating how well a system will perform under a particular load. A load test's findings will show how much work is put on the application server, database, and other systems as well as the importance of key business transactions. Stress testing is carried out to ascertain the system's maximum capacity and how it will operate if the present load is greater than the predicted maximum.

Soak tests, often called endurance tests, are used to evaluate a system's performance under a steady load. During soak testing, memory usage is monitored to identify performance issues like memory leaks. Monitoring the system's performance over time is the main objective. When testing during a "spike," the user base is rapidly expanded and the system's performance is swiftly examined. The main objective is to assess the system's workload management capabilities.

**7.3 TEST CASES:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | **Output** |  | **Result** |
| Input Image | Providing Application | Success |  |

**Test cases Model building:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **Test cases** | **I/O** | **Expected O/T** | **Actual O/T** | **P/F** |
| 1 | View page | Images dataset | Images | Showed  Successfully | P |
| 2 | Model page | Applying algorithms | Fitting the model | Applied  Successfully | P |
| 3. | Prediction page | Entering Image- classify | P>N>N | Showed  Successfully | P |
| 4. | View page | Images dataset | Rows/columns | Showed  Successfully | P |
| 5 | Model page | Applying algorithms | Fitting the model | Applied  Successfully | P |
| 6 | Prediction page | Entering input features | Output Classes | Showed  Successfully | P |

**CHAPTER-8**

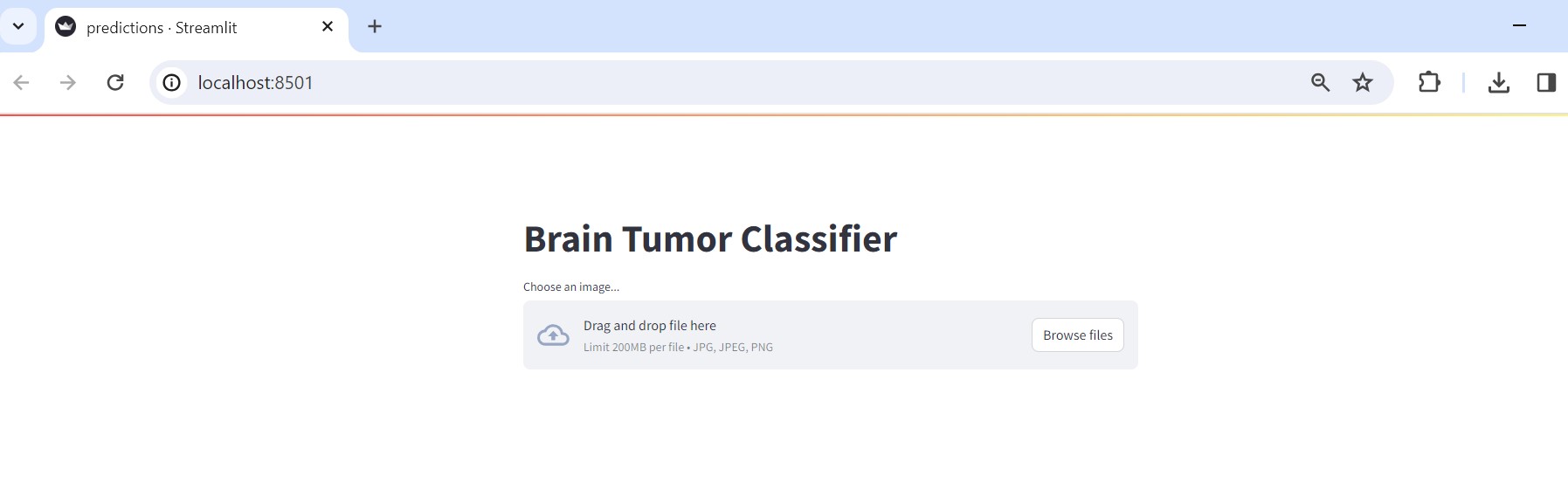
**RESULTS**

**8. RESULTS**

**OUTPUT SCREEN SHOTS WITH DESCRIPTION:**

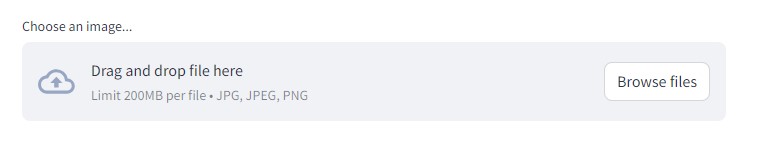
**HOME PAGE:**

The user is currently seeing the web application's home page.



**Figure 8.1**: **Home Page**

**UPLOAD IMAGE:** The user can upload the images.



**Figure 8.2**: **Upload Page**

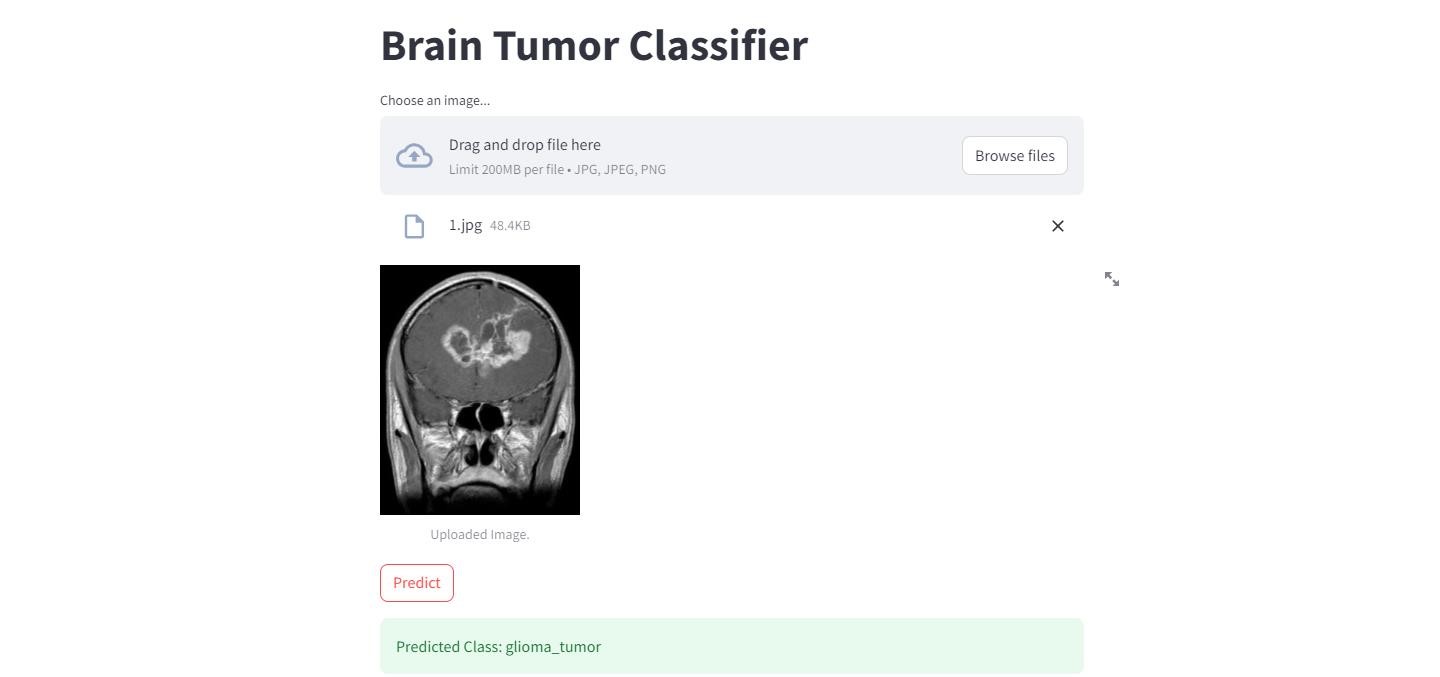
**PREDICTIONS PAGE:**

The user can predicts the outputs.

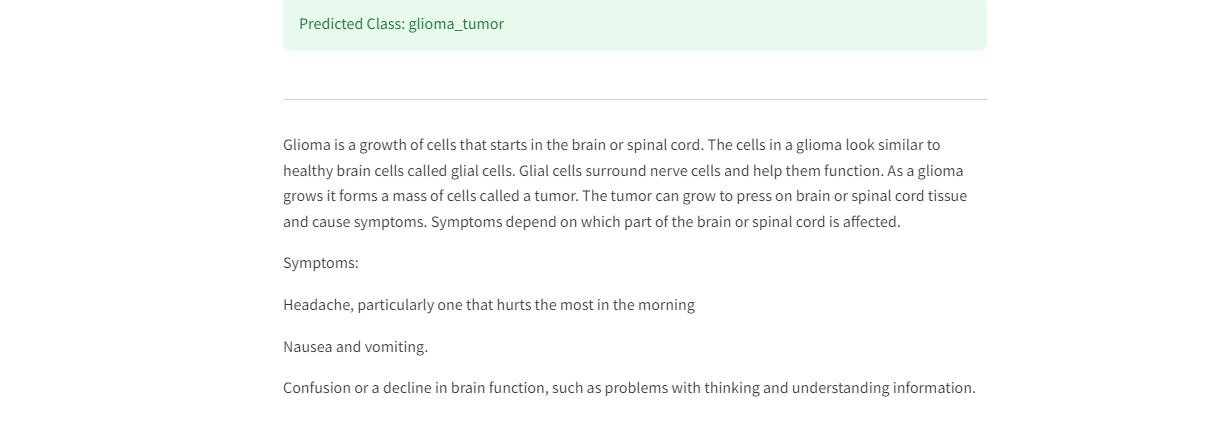


**Figure 8.3 Predictions Page**

The user can upload and predicts results.



**Figure 8.4 Predictions Page**



**Fig.8.5 Predictions Page**



**Fig.8.6 Predictions Page**

**CHAPTER 9**

**CONCLUSION**

**AND**

**FUTURE ENHANCEMENTS**

**9. CONCLUSION AND FUTURE ENHANCEMENT**

In summary, a major breakthrough in medical diagnostics has been made with the creation of an accurate brain tumor detection system that combines Convolutional Neural Networks (CNNs) with an intuitive Streamlit user interface (UI). This novel methodology provides automatic, accurate, and easily accessible tumor categorization, therefore addressing the shortcomings of current techniques. The device improves patient outcomes, increases productivity, and simplifies the diagnostic process by fusing cutting-edge technology with an easy design. Moreover, it gives patients the power to actively engage in their healthcare decisions by offering thorough information about tumor kinds and ramifications. All things considered, the suggested method has enormous potential to transform the identification of brain tumors, aiding in early diagnosis, efficient treatment planning, and ultimately, improved patient care.

In terms of upcoming improvements, there are several opportunities for growth and development. First off, the CNN model may be further improved and optimized to increase its resilience and accuracy. This will allow it to detect a wider variety of tumor kinds and subtypes with even higher precision. Furthermore, the amalgamation of multimodal imaging data, such merging MRI and PET or CT scans, might furnish supplementary insights and enhance the precision of diagnosis. Furthermore, using cutting-edge imaging methods like diffusion tensor imaging or functional magnetic resonance imaging might provide insights into tumor features and their effects on brain function, enabling customized treatment plans. Additionally, the system's accessibility may be expanded to remote or resource-constrained areas through the creation of mobile or cloud-based versions, guaranteeing fair healthcare delivery. To further improve the system's impact on brain tumor diagnosis and patient care, further collaboration with medical experts and patient advocacy organizations can help customize it to users' changing requirements and preferences. Fundamentally, brain tumor detection in the future will depend on ongoing innovation, teamwork, and adaptability to new medical trends and technology.

UGC CARE Group-1, Sr. No.-155 (Sciences) 8