

“ BRAIN TUMOR BOUNDARY DETECTION FOR PROGNOSTIC ASSESSMENT”

*Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of*

**Bachelor of Technology
in
Computer Science & Engineering**

By

DARA SRI KUMUDA VALLI	(21UECM0054)	(VTU 20578)
GOGINENI POOJITHA	(21UECM0081)	(VTU 19484)
YAZHINI G	(21UECM0265)	(VTU 19496)

*Under the guidance of
Dr.S. SRIDEVI,ME.,Ph.D.,
PROFESSOR*



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING**

**VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF
SCIENCE & TECHNOLOGY**

(Deemed to be University Estd u/s 3 of UGC Act, 1956)

**Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA**

May, 2024

“BRAIN TUMOR BOUNDARY DETECTION FOR PROGNOSTIC ASSESSMENT”

*Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of*

**Bachelor of Technology
in
Computer Science & Engineering**

By

DARA SRI KUMUDA VALLI (21UECM0054) (VTU 20578)
GOGINENI POOJITHA (21UECM0081) (VTU 19484)
YAZHINI G (21UECM0265) (VTU 19496)

*Under the guidance of
Dr.S. SRIDEVI,ME.,Ph.D.,
PROFESSOR*



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING**

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
(Deemed to be University Estd u/s 3 of UGC Act, 1956)

**Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA**

May, 2024

CERTIFICATE

It is certified that the work contained in the project report titled “BRAIN TUMOR BOUNDARY DETECTION FOR PROGNOSTIC ASSESSMENT” by DARA SRI KUMUDA VALLI (21UECM0054), GOGINENI POOJITHA (21UECM0081), YAZHINI G (21UECM2065)” has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor

Computer Science & Engineering

School of Computing

Vel Tech Rangarajan Dr. Sagunthala R&D

Institute of Science & Technology

May, 2024

Signature of Professor In-charge

Computer Science & Engineering

School of Computing

Vel Tech Rangarajan Dr. Sagunthala R&D

Institute of Science & Technology

May, 2024

DECLARATION

We declare that this written submission represents our ideas in our own words and where others ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

(DARA SRI KUMUDA VALLI)

Date: / /

(Signature)

(GOGINENI POOJITHA)

Date: / /

(Signature)

(YAZHINI G)

Date: / /

APPROVAL SHEET

This project report entitled “ BRAIN TUMOR BOUNDARY DETECTION FOR PROGNOSTIC ASSESSMENT” by (DARA SRI KUMUDA VALLI (21UECM0054), GOGINENI POOJITHA (21UECM0081), YAZHINI G (21UECM0265)) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners

Supervisor

Dr.S. SRIDEVI,ME.,Ph.D.,
PROFESSOR

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN**, for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.**, for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Head, Department of Computer Science & Engineering, Dr.M.S. MURALI DHAR, M.E., Ph.D.**, for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our Internal Supervisor **Dr.S. SRIDEVI,M.E.,Ph.D.**, for her cordial support, valuable information and guidance, she helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. U.HEMAVATHI, M.E., Ms. C. SHYAMALA KUMARI, M.E.**, for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

DARA SRI KUMUDA VALLI	(21UECM0054)
GOGINENI POOJITHA	(21UECM0081)
YAZHINI G	(21UECM0265)

ABSTRACT

The brain is the most important organ in the human body, responsible for controlling and regulating all critical life functions for the body and a tumor is a mass of tissue formed by the accumulation of abnormal cells, which keep on growing. A brain tumor is a tumor which is either formed in the brain or has migrated. No primary cause has been identified for the formation of tumors in the brain till date. Though tumors in the brain are not very common (Worldwide brain tumors make up only 1.8 of total reported tumors), the mortality rate of malignant brain tumors is very high due to the fact that the tumor formation is in the most critical organ of the body.

Hence, it is of utmost importance to accurately detect brain tumors at early stages to lower the mortality rate. We have thus proposed a computer-assisted radiology system which will assess brain tumors from MRI scans for the management of brain tumor diagnosis. In this study, we have implemented a model that segments images using Watershed and PSO algorithm, extracts features using DWT and PCA algorithms and finally classifies the tumors using CNN, SVM and Lazy IBK algorithms with a 93% of accuracy which is very high compared to other existing algorithm.

Keywords:

Brain Tumour, CNN, Progressive Human Disease, Machine Learning

LIST OF FIGURES

4.1	Architecture Diagram of Brain Tumor Detection	12
4.2	Data Flow Diagram	13
4.3	Use Case Diagram	14
4.4	Class Diagram	15
4.5	Sequence Diagram	16
4.6	Collaboration Diagram	17
4.7	Activity Diagram	18
4.8	Data Analysis	21
4.9	Convolutional Layer	22
4.10	Max pooling and Flattening	23
4.11	Inserting Dataset	24
4.12	Data Preprocessing	25
4.13	Disease Classification	26
5.1	Data Collection For Testing	27
5.2	Output of Prediction	28
5.3	Test Result of Brain Tumor Detection	30
5.4	Test Result of Brain Tumor Detection	31
5.5	Test Result of Brain Tumor Detection	32
5.6	Test Result of Training and Validation of Brain Tumor	33
6.1	Output 1 of Brain Tumor Detection	38
6.2	Output 2 of Brain Tumor Detection	39
8.1	Plagiarism Report	41
9.1	Poster Presentation	45

LIST OF ACRONYMS AND ABBREVIATIONS

AI	Artificial Intelligence
API	Application Programming Interface
CTs	Computed Tomography Scan
CNN	Convolutional Neural Network
DL	Deep Learning
GLCM	Gray Level Co-occurrence Matrix
ML	Machine Learning
MRI	Magnetic Resonance Imaging
SVM	Support Vector Machine
UI	User Interface
UX	User Experience

TABLE OF CONTENTS

	Page.No
ABSTRACT	v
LIST OF FIGURES	vi
LIST OF ACRONYMS AND ABBREVIATIONS	vii
1 INTRODUCTION	1
1.1 Introduction	1
1.2 Aim of the project	2
1.3 Project Domain	2
1.4 Scope of the Project	3
2 LITERATURE REVIEW	4
3 PROJECT DESCRIPTION	8
3.1 Existing System	8
3.2 Proposed System	8
3.3 Feasibility Study	9
3.3.1 Economic Feasibility	9
3.3.2 Technical Feasibility	9
3.3.3 Social Feasibility	10
3.4 System Specification	10
3.4.1 Hardware Specification	10
3.4.2 Software Specification	10
3.4.3 Standards and Policies	11
3.4.4 Google Colab	11
4 METHODOLOGY	12
4.1 General Architecture	12
4.2 Design Phase	13
4.2.1 Data Flow Diagram	13
4.2.2 Use Case Diagram	14

4.2.3	Class Diagram	15
4.2.4	Sequence Diagram	16
4.2.5	Collaboration diagram	17
4.2.6	Activity Diagram	18
4.3	Algorithm & Pseudo Code	19
4.3.1	Enhanced Convolutional Neural Network	19
4.3.2	Pseudo Code	19
4.4	Module Description	20
4.4.1	Dataset Used for Analysis of Tumor	20
4.4.2	Convolution Layer	21
4.4.3	Max Pooling Layer and Flatten	22
4.5	Steps to execute/run/implement the project	24
4.5.1	Inserting Dataset	24
4.5.2	Data Preprocessing	25
4.5.3	Disease Classification	26
5	IMPLEMENTATION AND TESTING	27
5.1	Input and Output	27
5.1.1	Data Collection	27
5.1.2	Output Design	28
5.2	Types of Testing	28
5.2.1	Unit Testing	28
5.2.2	Integration Testing	30
5.2.3	System Testing	31
5.2.4	Test Result	33
6	RESULTS AND DISCUSSIONS	34
6.1	Efficiency of the Proposed System	34
6.2	Comparison of Existing and Proposed System	34
6.3	Sample Code	36
7	CONCLUSION AND FUTURE ENHANCEMENTS	40
7.1	Conclusion	40
7.2	Future Enhancements	40
8	PLAGIARISM REPORT	41

9	SOURCE CODE & POSTER PRESENTATION	42
9.1	Source Code	42
9.2	Poster Presentation	45
	References	45

Chapter 1

INTRODUCTION

1.1 Introduction

The human body consists of myriad number of cells. When cell growth becomes uncontrollable the extra mass of cell transforms into tumor. CT scans and MRI scans are used for identification of the tumor. The goal of our study is to accurately detect tumors in the brain and classify it through the means of several techniques involving medical image processing, pattern analysis, and computer vision for enhancement, segmentation and classification of brain diagnosis. This system can be used by neurosurgeons, radiologists and healthcare specialists. The system is expected to improve the sensitivity, specificity, and diagnostic efficiency of brain tumor screening using industry standard simulation software tool, MATLAB. These techniques involve pre-processing of MRI scans collected from online cancer imaging archives as well as scans obtained from several pathology labs. Images are resized and then we apply the proposed algorithms for segmentation and classification. The system is expected to improve the brain tumor screening procedure currently at use, and possibly reduce health care costs by decreasing the need for followup procedures. Several processing steps are required for the accurate characterization and analysis of biomedical image data.

Similarly, the role of AI and ML in healthcare is essential. ML algorithm-based applications or tools are used in different areas of healthcare. AI-based robots are widely used to collect disease images. This saves a lot of workforce, money, and time. In precision healthcare, the AI and IoT helps the doctors as well as patients to improve the overall prediction of the disease quality and accuracy. With the help of sensors, diseases can be targeted and detected and then decide which medicines to apply to that region. In the region of the supply chain, many AI applications help in the smooth operation. The major challenge is in the broad adoption of AI applications due to the lack of simple solutions and seamless operation of the tools and applications.

This problem restricts the use of technology in the healthcare domain. The motivation behind this research is to use the AI and technology in the healthcare field because AI and technology have huge potentials to bring revolution in healthcare sector like usage of large machinery, etc. This report is a step towards solving the patient's problem of not knowing the disease they have, which takes more time and workforce. The outcome of this report will solve these diseases detection problem with higher accuracy in less time. In this paper, a mobile app is developed using the ML-based model to detect these diseases. The application is very simple to use and does not have complex UI/UX. For the model preparation, CNN-based image classification APIs are used.

1.2 Aim of the project

This project is based on detecting the tumor, from the images using Deep Learning (CNN) and implementation in application. The application is developed for the purpose to provide the users with an easy way to interact with the app to detect the disease in early stage. This app basically helps us to prevent the disease in the early stage by detecting the disease. This application basically helps us to upload the image of the affected disease that he/she wishes to diagnose.

1.3 Project Domain

This project spans lots of grounds starting from business concept to real world, and required to perform several researches to achieve the project objectives. The areas include: This includes study on how the disease detection system business is running in market, and also involves the process of how to improve the existence of the business. General users as well as the different researchers, companies will be able to use the system.

The application is developed to provide the users with an easy way to interact and detect what is going on in their health. This app basically helps everyone who wants to check their health in home, it help peoples to detect the right disease from the scanned images and help them to get a healthy life.

1.4 Scope of the Project

The main objectives of this application is to help the users to identify the diseases at the early stage, make a user friendly platform to detected disease, to ease the task for us to identify the disease with the help of technology which is better than naked eye. The android is developed to provide the users with an easy way to interact with the app to detect what is going on in their health.

Main aim is to develop an automated system for enhancement, segmentation and classification of brain tumors. The system can be used by neurosurgeons and healthcare specialists. The system incorporates image processing, pattern analysis, and computer vision techniques and is expected to improve the sensitivity, specificity, and efficiency of brain tumor screening. The proper combination and parameterization of above phases enables the development of adjunct tools that can help on the early diagnosis or the monitoring of the therapeutic procedures.

Chapter 2

LITERATURE REVIEW

[1] A. Ladgham, et al., Modified support vector machines for MR brain images recognition, International Conference on Control, Decision and Information Technologies (CoDIT). In this paper, Traditional SVM algorithms have been widely used for classification tasks due to their effectiveness in handling high-dimensional data. There are some limitations in standard SVMs when applied to MR brain image recognition, particularly in scenarios where datasets are imbalanced or non-linearly separable.

[2] P. Su, et al., Support vector machine (SVM) active learning for automated Glioblastoma segmentation, 9th IEEE International Symposium on Biomedical Imaging (ISBI). In this project, includes active learning framework based on SVM for Glioblastoma segmentation. Active learning is a machine learning paradigm that selects the most informative samples from an unlabeled dataset for annotation by an oracle with the goal of minimizing the number of labeled samples required for training while maximizing the performance of the model.

[3] Y. Liu, et al., An improved watershed algorithm based on multi-scale gradient and distance transformation, 3rd International Congress on Image and Signal Processing. The paper introduces an enhanced version of the watershed algorithm, a popular image segmentation technique, which incorporates multi-scale gradient and distance transformation to improve segmentation accuracy and robustness. The watershed algorithm is widely used for image segmentation, particularly in scenarios where objects of interest are characterized by intensity or gradient differences in the image

[4] MM. Al-Ayyoub, et al., Machine learning approach for brain tumor detection,” Proceedings of the 3rd International Conference on Information and Communication Systems - ICICS 12. The paper introduces a machine learning-based approach for the detection of brain tumors, addressing the critical need for accurate and efficient diagnostic methods in medical imaging. The proposed a machine learning approach

that leverages advanced algorithms to automatically detect brain tumors in medical images, particularly magnetic resonance imaging (MRI) scans.

[5] E. Kucukkahli, et al., Brain MRI Segmentation based on Different Clustering Algorithms, International Journal of Computer Applications, vol. 155, no. 3, pp. 37–40. Mohsen, Fahd, et al. "A new image segmentation method based on particle swarm optimization." Int. Arab J. Inf. Technol. The paper introduces a novel approach for brain MRI segmentation utilizing various clustering algorithms, aimed at accurately delineating different brain structures and abnormalities from MRI images, where accurate identification of anatomical structures and pathological regions is crucial for diagnosis and treatment planning.

[6] Mohsen, Fahd, et al. A new image segmentation method based on particle swarm optimization. Int. Arab J. Inf. Technol. 9.5 2023: 487- 493 .The paper introduces a method to address the challenge of accurate and efficient image segmentation, a fundamental task in computer vision and image processing. The proposed method involves treating image segmentation as an optimization problem, where the goal is to partition the image into meaningful regions while minimizing a segmentation objective function

[7] R. V. Rao, et al, A simple and new optimization algorithm for solving constrained and unconstrained optimization problems, International Journal of Industrial Engineering Computations. The paper presents a new approach to optimization that offers simplicity and effectiveness in solving a wide range of optimization tasks. The proposed algorithm utilizes a combination of search and update mechanisms to iteratively improve the solution quality. At each iteration, candidate solutions are generated based on exploration and exploitation of the search space.

[8] J. Han, M. Kamber, and J. Pei. Data mining: concepts and techniques. In this project the concept of data mining and its significance in extracting useful knowledge from large volumes of data. They discuss the interdisciplinary nature of data mining, which draws upon principles from statistics, machine learning, database systems, and visualization.

[9]. Maiti and M. Chakraborty .A new method for brain tumor segmentation based on watershed and edge detection algorithms in HSV colour model, National Conference On Computing And Communication Systems.he proposed method leverages the HSV (Hue, Saturation, Value) color model, which is particularly well-suited for segmenting images based on color information. The authors utilize the distinct advantages of the HSV color space, such as its ability to separate intensity (value) and color information (hue and saturation), to enhance the segmentation performance.

[10] Y. K. Dubey,et al.,Segmentation of brain MR images using intuitionistic fuzzy clustering algorithm, Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing - ICVGIP 12. The paper contributes to the field of medical image analysis by introducing a novel segmentation method based on the intuitionistic fuzzy clustering algorithm. The research findings presented in the paper have important implications for improving the accuracy and reliability of brain MR image segmentation, ultimately enhancing the diagnosis and treatment of neurological conditions.

[11]Kaur, Mandhir,et al.,Survey of Intelligent Methods for Brain Tumor Detection. International Journal of Computer Science Issues (IJCSI) 11.5. The paper begins by discussing the challenges associated with traditional brain tumor detection methods, such as manual inspection of medical images by radiologists, which can be time-consuming and prone to human error. Intelligent methods offer the potential to overcome these challenges by automating the analysis of medical images and providing accurate and efficient tumor detection.

[12]S. M. K. Hasan,et al,Perceptive Proposition of Combined Boosted Algorithm for Brain Tumor Segmentation,Proceedings of the International Conference on Advances in Information Communication Technology Computing - AICTC 16. The proposed method combines multiple boosted algorithms, leveraging their complementary strengths to improve segmentation accuracy and robustness. Boosting is an ensemble learning technique that combines the predictions of multiple weak learners to create a strong classifier or regressor. By integrating different boosted algorithms, the proposed method aims to exploit diverse features and capture complex patterns in the image data.

[13] D. Sridhar, et al., Brain Tumor Classification using Discrete Cosine Transform and Probabilistic Neural Network, International Conference on Signal Processing , Image Processing Pattern Recognition. The paper contributes to the field of medical image analysis by introducing a novel method for brain tumor classification using the Discrete Cosine Transform and Probabilistic Neural Network. The research findings presented in the paper have important implications for improving the accuracy and efficiency of brain tumor diagnosis, ultimately enhancing patient care and treatment outcomes.

[14] Q. Sun and H. Tian., Interactive image segmentation using power watershed and active contour model, 3rd IEEE International Conference on Network Infrastructure and Digital Content. This project leverages the power watershed algorithm, which extends traditional watershed segmentation by incorporating gradient information and user-defined markers to guide the segmentation process. Additionally, the method integrates an active contour model, also known as a snake model, which represents a deformable curve that evolves to minimize an energy function defined based on image features and user constraints.

[15] H. Li, et al., Label propagation with robust initialization for brain tumor segmentation, 9th IEEE International Symposium on Biomedical Imaging (ISBI). This paper contributes to the field of medical image analysis by introducing a novel approach for brain tumor segmentation. By combining label propagation with robust initialization, the proposed method achieves accurate and reliable segmentation results, providing valuable support for clinical decision-making in brain tumor diagnosis and treatment.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

Pre-processing procedures in the current system could involve standard tasks like noise reduction, image scaling, and normalization. Simple thresholding or edge detection techniques are often used for image segmentation, but they may not be able to precisely capture the complex borders of tumors, which could result in segmentation errors and inaccuracies. The intricacy of tumor traits may not be adequately captured by handcrafted features used in feature extraction approaches, such as texture or intensity histograms.

Conventional machine learning algorithms like random forests and support vector machines (SVM) are frequently used for categorization. Although these techniques can yield respectable outcomes, their efficacy in precisely categorizing tumors may be restricted by the high-dimensional and nonlinear characteristics of data.

Disadvantages:

- 1.Existing brain tumor detection systems, while advanced and effective in many ways, still have some disadvantages and limitations.
- 2.Many existing systems suffer from data imbalance, where there are significantly more samples of one class (e.g., healthy images) compared to another (e.g., tumor images).
- 3.Deep learning models used in brain tumor detection are often considered "black boxes," meaning it's difficult to interpret how they arrive at their decisions.

3.2 Proposed System

When compared to conventional methods, the suggested system for brain tumor identification and classification utilizing machine learning more particularly, CNN offers a more sophisticated and efficient way. Because CNNs can automatically learn hierarchical features from raw data, they have demonstrated extraordinary per-

formance in a variety of image processing tasks, including medical picture analysis. CNNs are used in the proposed system for tasks related to both image segmentation and classification. The CNN architecture is taught to precisely and automatically recognize and define tumor locations in MRI images in order to perform segmentation. Even in situations where tumors have irregular shapes or subtle boundaries, this method enables for more accurate and consistent tumor delineation by doing away with the requirement for manual segmentation.

Advantages:

1. Proposed systems often leverage advanced deep learning techniques, such as convolutional neural networks (CNNs), to achieve higher levels of accuracy in tumor detection.
2. Many proposed systems employ techniques to address data variability and imbalance, allowing them to generalize well to diverse datasets and imaging modalities.
3. By incorporating strategies such as data augmentation, transfer learning, and domain adaptation, these systems can better adapt to new imaging protocols and patient populations, improving their robustness and reliability.

3.3 Feasibility Study

3.3.1 Economic Feasibility

Development of this application is highly economically feasible. Whether the firm can afford to build the software, whether its benefits should substantially exceed its cost. Our project is economically feasible. Our system uses academic version of MATLAB R2017a, which was very feasible, economically since it can be viewed as a one time investment.

3.3.2 Technical Feasibility

Whether the technology needed for the system exists, how difficult it is to build. Our project is technically versatile system which can work on most platforms making it technically feasible to build requiring only few specifications. Software used for the project implementation is MATLAB. Basic technical knowledge of operating MATLAB software along with the classification toolbox is required for the developers.

3.3.3 Social Feasibility

The social feasibility of using machine learning for tumor detection is multifaceted and depends on various factors, including technological readiness, accessibility, ethical considerations, and societal acceptance. Machine learning algorithms have shown promising results in tumor detection from medical imaging such as MRI, CT scans, and X-rays. Continued advancements in machine learning techniques, coupled with the availability of large datasets for training, contribute to the feasibility of this approach. One of the critical factors in the social feasibility of tumor detection using machine learning is accessibility to the technology.

Ensuring compliance with data protection regulations and implementing robust privacy-preserving techniques is essential to address these concerns. The acceptance of machine learning-based tumor detection by patients, healthcare providers, and society at large plays a crucial role in its social feasibility. Education and awareness campaigns about the benefits, limitations, and risks associated with these technologies can help foster acceptance and mitigate apprehensions. The acceptance of machine learning-based tumor detection by patients, healthcare providers, and society at large plays a crucial role in its social feasibility.

3.4 System Specification

3.4.1 Hardware Specification

- Intel Dual Core Dual Processor or advanced version
- Minimum 8GB of RAM
- Minimum 1 GB of Hard disk Space

3.4.2 Software Specification

- Dataset obtained from pathology labs
- Online dataset from www.cancerimagingarchive.net
- Microsoft Office
- MATLAB 2017a

3.4.3 Standards and Policies

3.4.4 Google Colab

Google Colab is a free, cloud-based platform provided by Google that allows you to write and execute Python code in a collaborative environment. It is particularly useful for data analysis, machine learning, and other scientific computing tasks.

Standard Used:ISO/IEC 27001

Chapter 4

METHODOLOGY

4.1 General Architecture

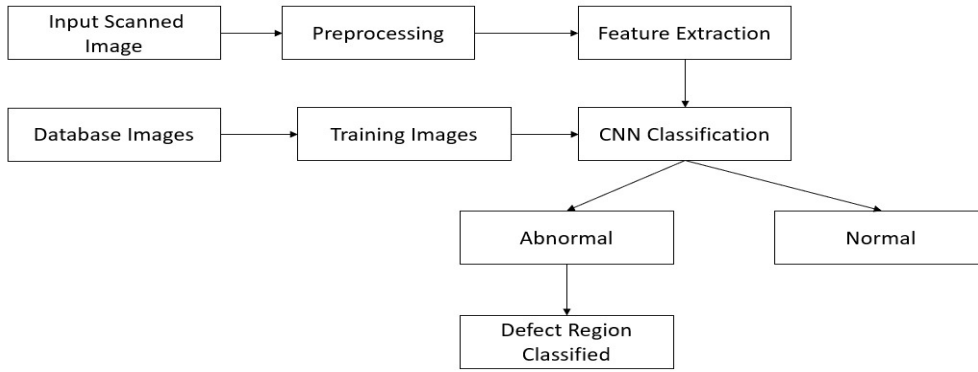


Figure 4.1: Architecture Diagram of Brain Tumor Detection

Figure 4.1 represents the Architecture Diagram, the input is given in form of image and the image is sent to preprocess data for feature extraction. Those extracted details are sent to Convolutional Neural Network (CNN), where with used of algorithm there are database image and trained images. In Convolutional Neural Network (CNN), we can classify into two Normal and Abnormal. If it is Abnormal then it shows the prediction of tumour is provided and defect region is classified.

4.2 Design Phase

4.2.1 Data Flow Diagram

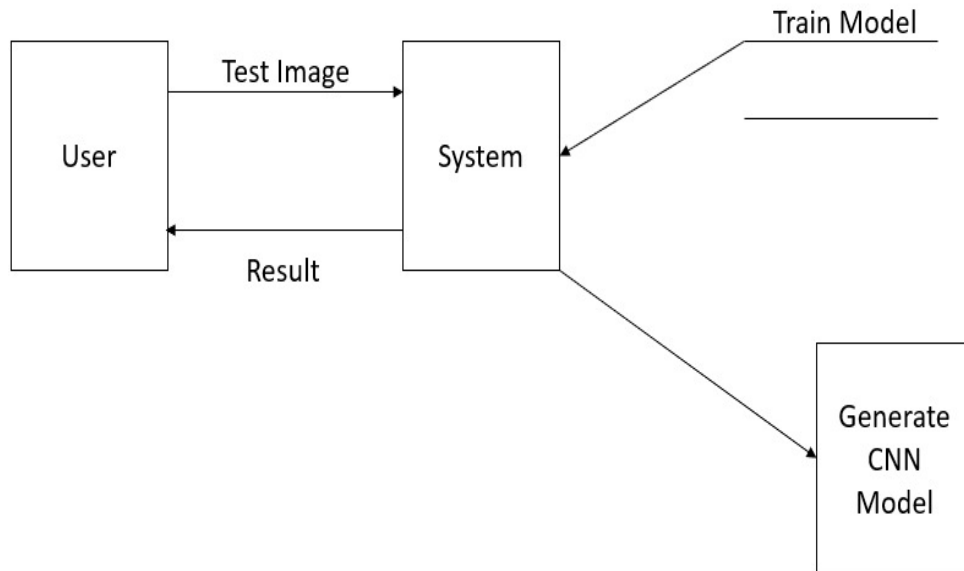


Figure 4.2: Data Flow Diagram

Figure 4.2 represents the Dataflow Diagram ,as we see the user can send and receive the information . The input is given to the system and output is generated in form of instructions.The System is a trained model to generate a appropriate result.

4.2.2 Use Case Diagram

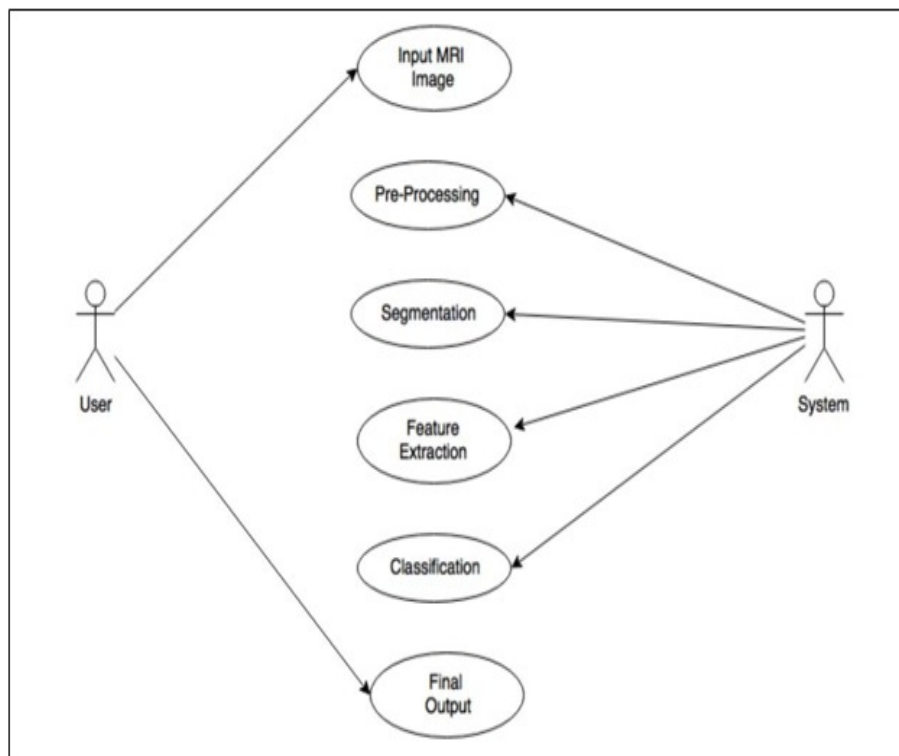


Figure 4.3: Use Case Diagram

Figure 4.3 represents the Use Case diagram for tumor detection using machine learning, you would typically identify the various actors or entities interacting with the system and the specific actions or use cases they perform. The primary actor responsible for interpreting medical imaging scans and providing diagnoses. They interact with the system to access and analyze images processed by the machine learning algorithm. The core component of the system responsible for tumor detection using machine learning algorithms. It processes medical imaging data and identifies potential tumors or abnormalities. The core use case where the machine learning algorithm analyzes the preprocessed imaging data to detect tumors or abnormal structures. This may involve segmentation of the tumor region and classification of tumor types. This use case involves uploading medical imaging data into the system for analysis. It may include functionalities for securely transmitting and storing patient data in compliance with privacy regulations.

4.2.3 Class Diagram

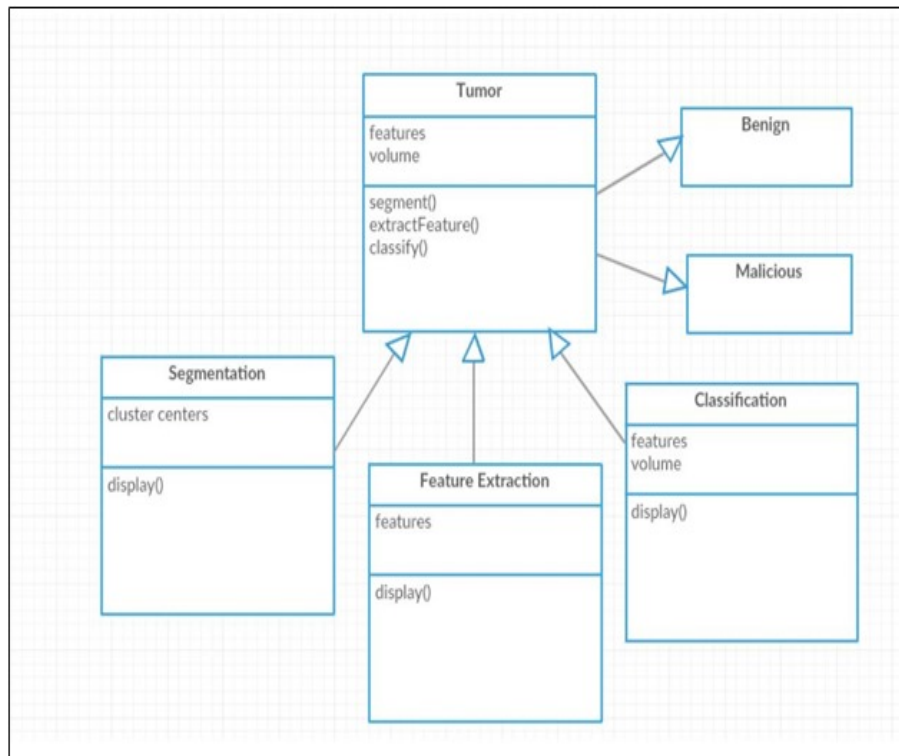


Figure 4.4: **Class Diagram**

Figure 4.4 represents the Class diagram for tumor detection using machine learning. It outlines the various classes (or entities) in the system, along with their attributes and relationships. This class diagram outlines the key entities involved in the tumor detection process using machine learning, including patients, medical images, tumor detection algorithms, detection results, healthcare providers, diagnoses, and treatment plans. It also illustrates the associations between these entities, highlighting how they are related to each other within the system. Depending on the specific requirements and functionalities of the system, you may further refine or expand upon these classes.

4.2.4 Sequence Diagram

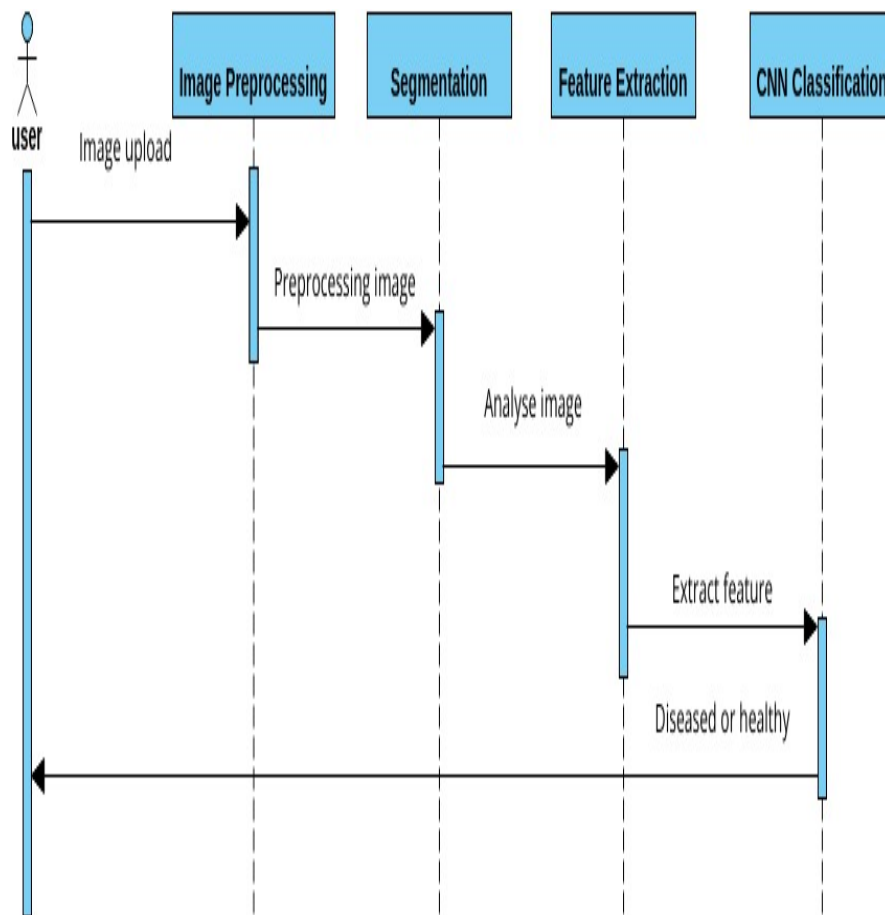


Figure 4.5: Sequence Diagram

Figure 4.5 represents the Sequence diagram for tumor detection using machine learning illustrates the interactions between different components or actors in the system over time. The sequence starts when a patient uploads a medical image for tumor detection. The Patient actor initiates the interaction by sending a request to the system to upload the medical image. The System acknowledges the request and receives the medical image data. Once the image is uploaded, the System preprocesses the medical image to enhance its quality for analysis. This step involves noise reduction, normalization, and other preprocessing techniques. The System sends a message indicating the completion of preprocessing to the Patient. After preprocessing, the System applies the tumor detection algorithm to the preprocessed medical image. The TumorDetectionAlgorithm processes the image data and generates a detection result. This sequence diagram provides a high-level overview of the interactions involved in tumor detection using machine learning, from image upload to communication of results to the patient.

4.2.5 Collaboration diagram

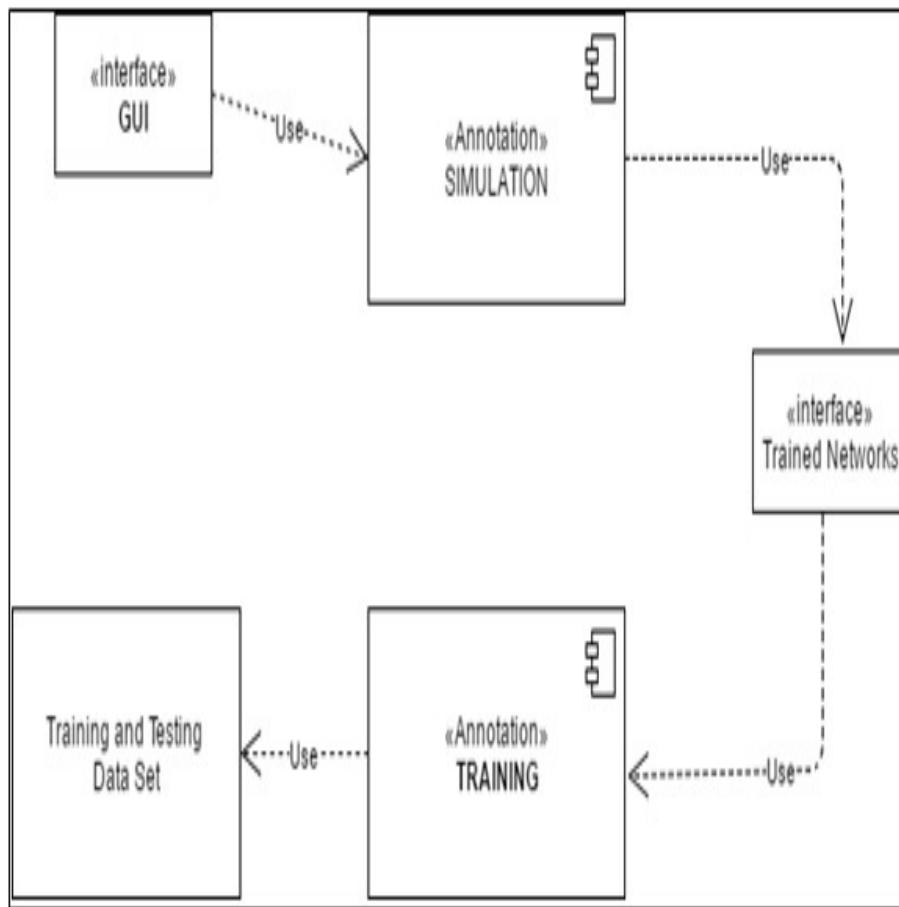


Figure 4.6: Collaboration Diagram

Figure 4.6 represents the Collaboration Diagram, also known as a communication diagram, illustrates the interactions and relationships between objects or actors in a system. In the context of tumor detection using machine learning, a collaboration diagram can depict how various components collaborate to achieve tumor detection. Initiates the process by uploading a medical image for tumor detection. Represents the overall system responsible for coordinating the tumor detection process. An object representing the uploaded medical image. It interacts with the System to undergo preprocessing and tumor detection. Represents the machine learning algorithm responsible for detecting tumors in medical images. It collaborates with the MedicalImage object to analyze the image data. This collaboration diagram illustrates the interactions between different objects and actors involved in the tumor detection process using machine learning. It highlights how these components collaborate to analyze medical images, detect tumors, make diagnoses, and formulate treatment plans for patients.

4.2.6 Activity Diagram

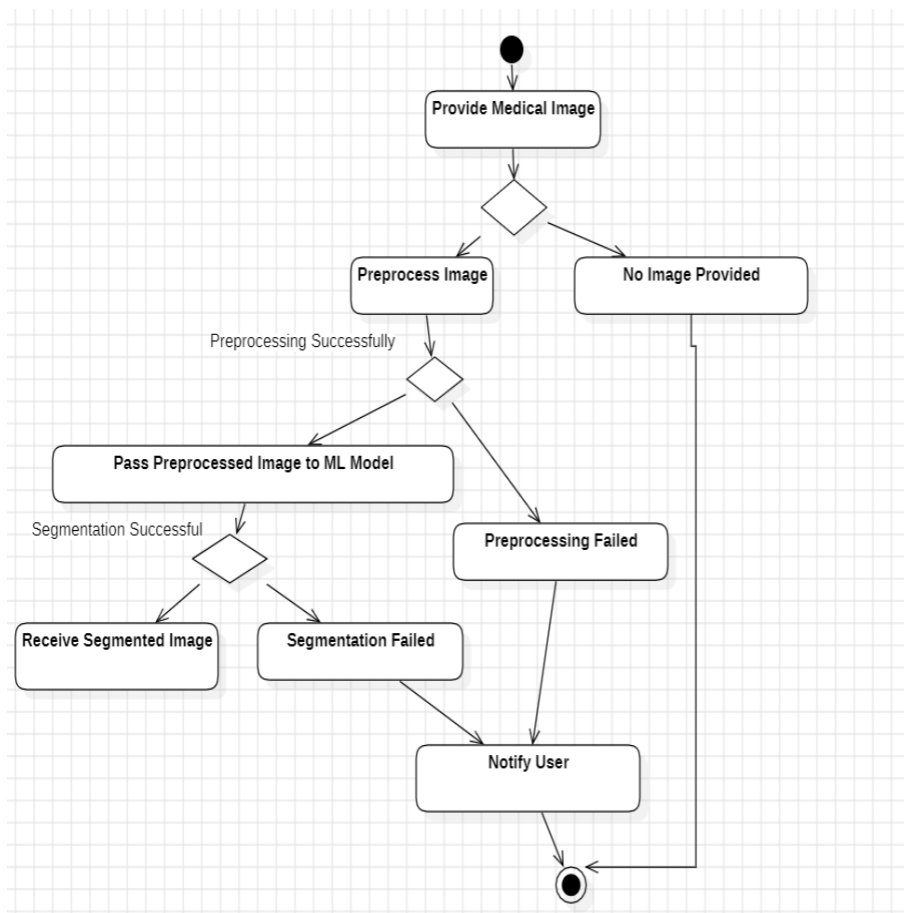


Figure 4.7: Activity Diagram

Figure 4.7 represents the Activity Diagram for tumor detection using machine learning can provide a visual representation of the workflow involved in the detection process. The process begins when a patient uploads a medical image for tumor detection. The system receives the uploaded image and proceeds to the preprocessing stage. The uploaded medical image undergoes preprocessing to enhance its quality and prepare it for tumor detection. Preprocessing steps may include noise reduction, normalization, and other image .After preprocessing, the system applies the tumor detection algorithm to the preprocessed image. The algorithm analyzes the image data to detect tumors and generates a detection result. The detection result is reviewed by a healthcare professional, such as a radiologist or oncologist. The healthcare professional interprets the results and makes further evaluations if necessary. This activity diagram provides a visual representation of the sequential workflow involved in tumor detection using machine learning, from image upload to communication of results to the patient.

4.3 Algorithm & Pseudo Code

4.3.1 Enhanced Convolutional Neural Network

Step-1 : Start

Step-2 : The CNN starts with the input layer where the raw data, usually images, is fed into the network.

Step-3 : In convolutional layer, the input image is convolved with learnable filters (also known as kernels or feature detectors). Each filter extracts different features from the input image by sliding over the image and performing element-wise multiplication and summation.

Step-4 : After the convolution operation, an activation function like Rectified Linear Unit is applied element-wise to introduce non-linearity into the network, allowing it to learn complex patterns and relationships in the data.

Step-5 : The pooling layer follows the activation function and reduces the spatial dimensions (width and height) of the input volume while retaining important information.

Step-6 : The output of the convolutional and pooling layers is flattened into a 1D vector. This step prepares the data for input into the fully connected layers.

Step-7 : The final layer of the CNN produces the desired output, such as class probabilities in classification tasks or continuous values in regression tasks. The activation function used in this layer depends on the task.

Step-8 : The gradients of the loss function with respect to the parameters of the network are computed using backpropagation.

Step-9 : The CNN is trained by iteratively feeding the training data through the network, computing the loss, and updating the parameters using backpropagation.

Step-10 : Once trained, the CNN is evaluated on a separate validation or test dataset to assess its performance and generalization ability.

step 11 : Finally, the trained CNN can be used to make predictions on new, unseen data, providing insights or making decisions based on the learned patterns in the input data.

Step-12 : Stop

4.3.2 Pseudo Code

```
import numpy as np import tensorflow as tf
```

```

2 model = tf.keras.models.load_model( Brain_Tumor_detection_model.h5 )image= tf.keras.
    preprocessing.image.load_img( brain_mri_images_for_brain_tumor_detection , target_size=(64,
    64))
3 image= tf.keras.preprocessing.image.img_to_array(image)
4 image = tf.expand_dims(image, axis=0)
5 prediction = model.predict(image)
6 predicted_class = np.argmax(prediction[0])
7 if predicted_class == 0:
8     print( Its_a_Tumor. )
9 elif predicted_class == 1:
10    print( No , Its not a tumor. )

```

4.4 Module Description

4.4.1 Dataset Used for Analysis of Tumor

Data was collected from various verified sources and then segregated into two types:

Cancerous (Malignant):

Malignant tumors, also known as cancerous tumors, are characterized by their ability to invade surrounding tissues and potentially spread to other parts of the body through a process called metastasis. These tumors typically exhibit uncontrolled growth, fueled by genetic mutations that disrupt the normal regulatory mechanisms of cell division and proliferation.

Non-Cancerous (Benign):

Benign tumors, unlike cancerous ones, are non-cancerous growths that typically do not invade nearby tissues or metastasize to other parts of the body. These tumors may develop in various organs or tissues, including the brain, breast, skin, and others. While they are generally considered less harmful than malignant tumors, benign growths can still cause health issues depending on their size and location.

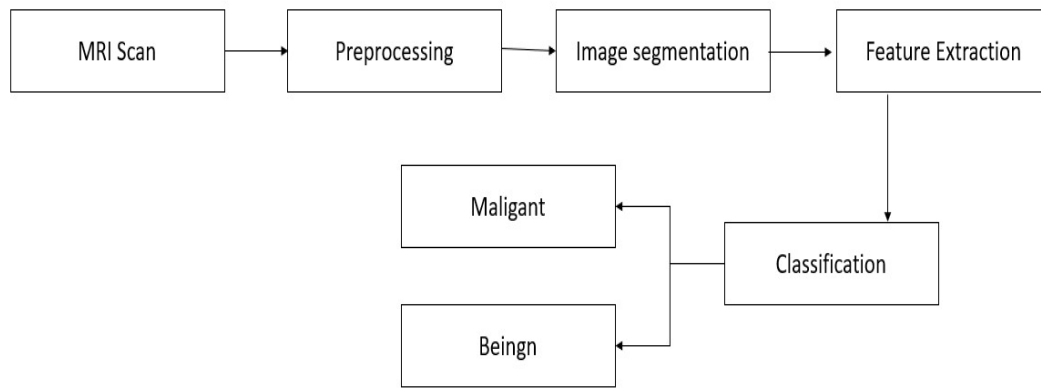


Figure 4.8: **Data Analysis**

Fig 4.8 represents the Data Analysis of Brain Tumor Detection.

4.4.2 Convolution Layer

It is the Beginning Layer and Converting all the images into $64 \times 64 \times 3$ homogeneous dimension of Convolutional kernel of 32 convolutional filters of size 3×3 with the support of 3 tensor channels and Activation function: ReLU. Feature Extraction in Convolutional layers are adept at extracting relevant features from input data. In brain tumor detection, these layers can identify patterns and structures indicative of tumors in medical images such as MRI scans. Training with Labelled Data in Convolutional layers are typically trained using supervised learning techniques, where they are fed with labelled examples of brain images containing tumors and healthy tissue. Through backpropagation, the network adjusts its parameters to minimize the difference between predicted and actual labels, improving its ability to accurately detect tumors.

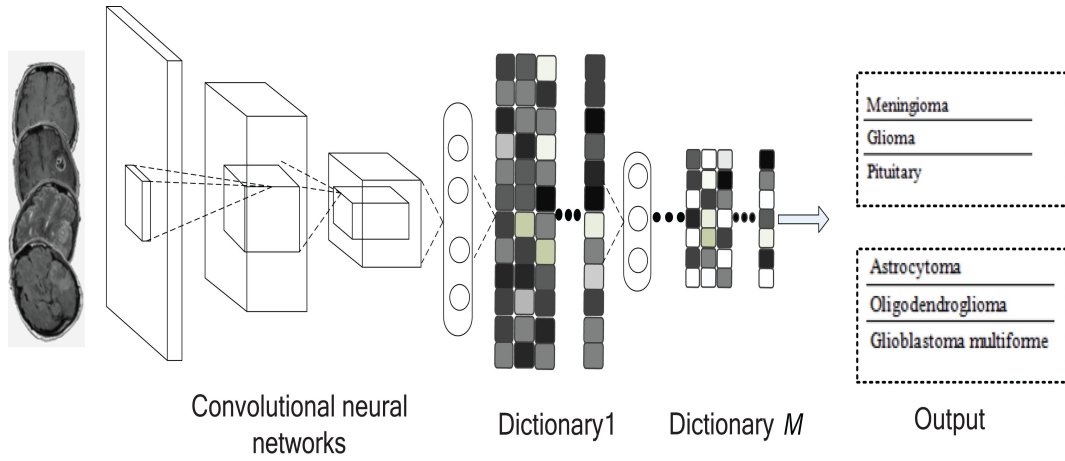


Figure 4.9: **Convolutional Layer**

Fig 4.9 represents the Convolutional Layer of Brain Tumor Detection.

4.4.3 Max Pooling Layer and Flatten

Max pooling is a downsampling operation commonly applied after convolutional layers in CNNs. It works by dividing the input feature map into non-overlapping rectangular regions and selecting the maximum value within each region. This process helps reduce the spatial dimensions of the feature maps while retaining the most important information. Max pooling helps the network become more invariant to small translations and variations in the input, making it more robust to changes in the position or orientation of features. Additionally, max pooling reduces the computational complexity of the network, aiding in faster training and inference.

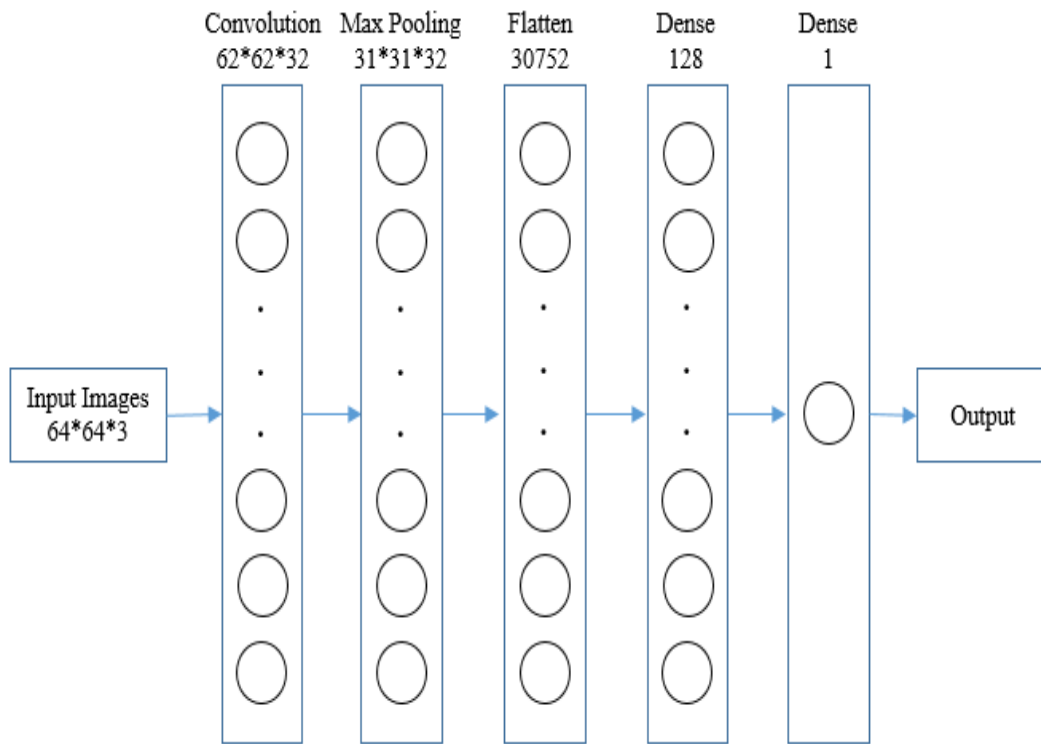


Figure 4.10: **Max pooling and Flattening**

Fig 4.10 represents the Max Pooling and Flattening of Brain Tumor. Flattening is a layer used to convert the multidimensional output of convolutional and pooling layers into a one-dimensional vector that can be fed into a fully connected (dense) layer for further processing. After the feature maps are generated by convolutional and pooling layers, flattening simply reshapes these feature maps into a single long vector by stacking the values along a single dimension. This transformation is necessary because fully connected layers require one-dimensional input vectors. Flattening enables the network to learn complex relationships between the extracted features and make predictions based on the entire input image. Both max pooling and flattening layers are crucial components of CNN architectures, allowing the network to efficiently extract and process features from input data, ultimately improving its performance in various tasks such as image classification, object detection, and medical image analysis, including brain tumor detection.

4.5 Steps to execute/run/implement the project

4.5.1 Inserting Dataset



```
tfile.write(data)
done = int(50 * dl / int(total_length))
sys.stdout.write(f"\r['=' * done]{' ' * (50-done)}] {dl} bytes downloaded")
sys.stdout.flush()
data = fileres.read(CHUNK_SIZE)
if filename.endswith('.zip'):
    with ZipFile(tfile) as zfile:
        zfile.extractall(destination_path)
else:
    with tarfile.open(tfile.name) as tarfile:
        tarfile.extractall(destination_path)
print(f'\nDownloaded and uncompressed: {directory}')
except HTTPError as e:
    print(f'Failed to load (likely expired) {download_url} to path {destination_path}')
    continue
except OSError as e:
    print(f'Failed to load {download_url} to path {destination_path}')
    continue

print('Data source import complete.')
```

Downloading brain-mri-images-for-brain-tumor-detection, 15828590 bytes compressed
[=====] 15828590 bytes downloaded
Downloaded and uncompressed: brain-mri-images-for-brain-tumor-detection
Data source import complete.

Figure 4.11: **Inserting Dataset**

Fig 4.11 shows the Data sets collection The datasets we collected are about the MRI images of the Tumour cells that are found in the different person brain scanned data.

4.5.2 Data Preprocessing

```
import matplotlib.pyplot as plt
plt.style.use('dark_background')
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder

2024-04-21 14:02:57.753258: E external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:9261] Unable to register cuDNN factory: Attempting to register factory for
2024-04-21 14:02:57.753432: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:687] Unable to register cuFFT factory: Attempting to register factory for p
2024-04-21 14:02:57.926298: E external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1515] Unable to register cuBLAS factory: Attempting to register factory for

One Hot Encoding the Target Classes

encoder = OneHotEncoder()
encoder.fit([[0], [1]])

# 0 - Tumor
# 1 - Normal

OneHotEncoder
OneHotEncoder()
```

Figure 4.12: Data Preprocessing

Fig 4.12 shows Here Data Processing is done. Analysing the above datasets and comparing data in each dataset has done and the quality of the data were before applying Machine Learning algorithms

4.5.3 Disease Classification

(BatchNormalization)		
max_pooling2d (MaxPooling2D)	(None, 64, 64, 32)	0
dropout (Dropout)	(None, 64, 64, 32)	0
conv2d_2 (Conv2D)	(None, 64, 64, 64)	8,256
conv2d_3 (Conv2D)	(None, 64, 64, 64)	16,448
batch_normalization_1 (BatchNormalization)	(None, 64, 64, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 64)	0
dropout_1 (Dropout)	(None, 32, 32, 64)	0
flatten (Flatten)	(None, 65536)	0
dense (Dense)	(None, 512)	33,554,944
dropout_2 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 2)	1,026

Total params: 33,585,602 (128.12 MB)
Trainable params: 33,585,410 (128.12 MB)
Non-trainable params: 192 (768.00 B)
None

Figure 4.13: Disease Classification

Fig 4.13 shows Predicting of output using Machine Learning Algorithms such as Convolutional Neural Networks for the given dataset of Tumour Prediction.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Data Collection

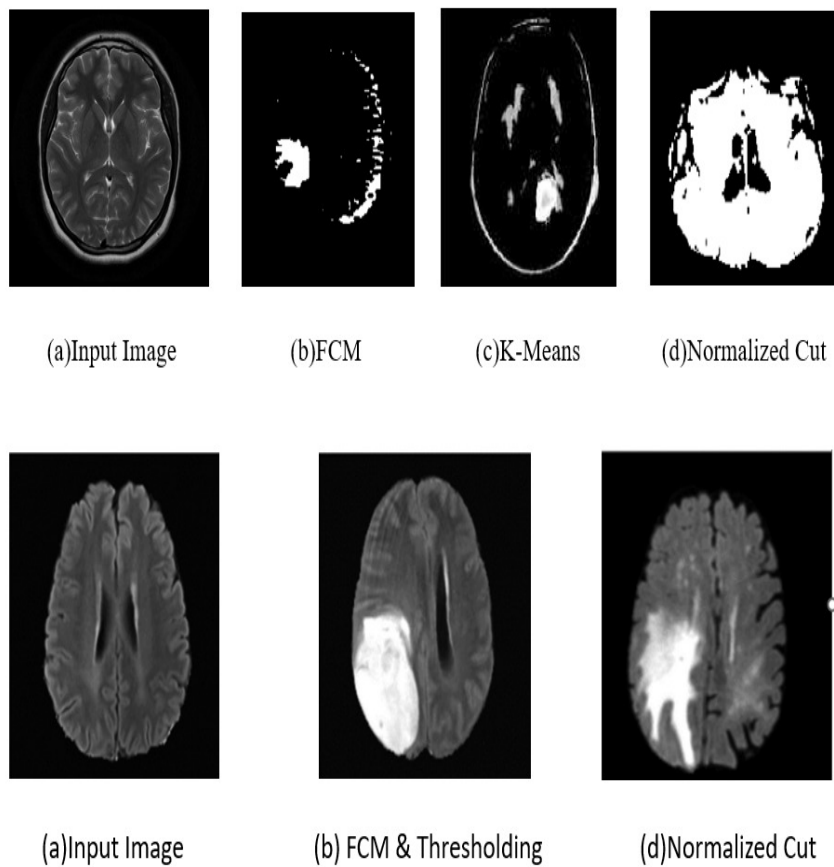


Figure 5.1: **Data Collection For Testing**

Fig 5.1 shows Input of Tumour prediction is considered in order to generate the output with Tumour dataset in which algorithm is used such as Convolutional neural networks so that by comparing all the results which was in the form of percentage we can able to find the accuracy among the algorithm we used.

5.1.2 Output Design

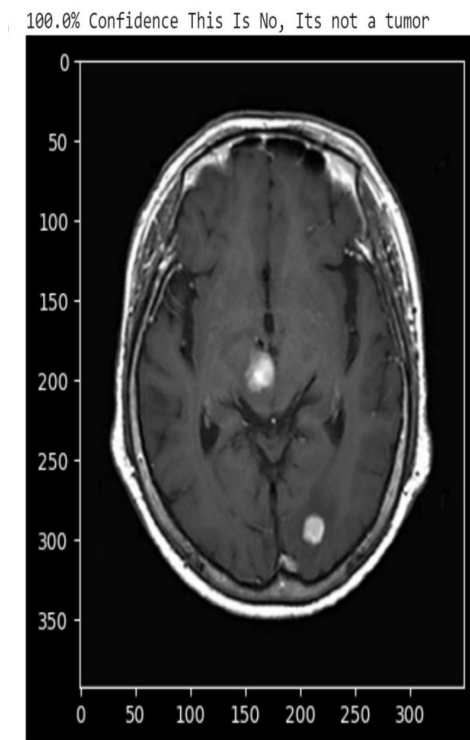


Figure 5.2: **Output of Prediction**

Fig 5.2 shows output of Tumour Detection was displayed in which for each algorithm we have different prediction algorithms to identify such that CNN algorithm shows more accuracy among the other algorithms. The efficiency among these technologies is obtained as the output by using the data collected from different datasets.

5.2 Types of Testing

5.2.1 Unit Testing

Unit testing is one of the most fundamental software testing process in which the smallest individual units of the software which are testable are scrutinized and ordered for proper execution of the software to give the desired output.

Input

```
1 import tensorflow as tf
2 from tensorflow . keras import layers
3 from tensorflow . keras . layers import Dense
4 import matplotlib . pyplot as plt
5 import numpy as np
6 import pandas as pd
7 IMAGE SIZE = 256
8 BATCH SIZE = 32
9 training_dataset = tf.keras.preprocessing.image_dataset_from_directory(
10     / content / drive / MyDrive / Train      ,
11     shuffle=True ,
12     image size = (IMAGE SIZE,IMAGE SIZE) ,
13     batch size = BATCH SIZE
14 )
15 testing_dataset = tf.keras.preprocessing.image_dataset_from_directory(
16     / content / drive / MyDrive / Test      ,
17     shuffle=True ,
18     image size = (IMAGE SIZE,IMAGE SIZE) ,
19     batch size = BATCH SIZE
20 )
21 validation_dataset = tf.keras.preprocessing.image_dataset_from_directory(
22     / content / drive / MyDrive / Valid      ,
23     shuffle=True ,
24     image size = (IMAGE SIZE,IMAGE SIZE) ,
25     batch size = BATCH SIZE
26 )
```

Test result

```
tfile.write(data)
done = int(50 * d1 / int(total_length))
sys.stdout.write(f"\r[{'=' * done}{' ' * (50-done)}] {d1} bytes downloaded")
sys.stdout.flush()
data = fileres.read(CHUNK_SIZE)
if filename.endswith('.zip'):
    with ZipFile(tfile) as zfile:
        zfile.extractall(destination_path)
else:
    with tarfile.open(tfile.name) as tarfile:
        tarfile.extractall(destination_path)
    print(f'\nDownloaded and uncompressed: {directory}')
except HTTPError as e:
    print(f'Failed to load (likely expired) {download_url} to path {destination_path}')
    continue
except OSError as e:
    print(f'Failed to load {download_url} to path {destination_path}')
    continue

print('Data source import complete.')
```

Downloading brain-mri-images-for-brain-tumor-detection, 15828590 bytes compressed
[=====] 15828590 bytes downloaded
Downloaded and uncompressed: brain-mri-images-for-brain-tumor-detection
Data source import complete.

Figure 5.3: Test Result of Brain Tumor Detection

Fig 5.3 represent the Test Result of Brain Tumor Detestion where the sample code execution in the Unit Testing.

5.2.2 Integration Testing

Integration testing is known as the second level of the software testing process, following unit testing. Integration testing involves checking individual components or units of a software project to expose defects and problems to verify that they work together as designed.

Input

```
1 model. fit (
2     training_dataset ,
3     epochs=10,
4     batch_size=32,
5     verbose=1,
6     validation_data=validation_dataset )
7 model.evaluate (test_data)
```

Test result

```
Epoch 17/30
3/3 ————— 10s 2s/step - accuracy: 0.9646 - loss: 0.1367 - val_accuracy: 0.8214 - val_loss: 3.3832
Epoch 18/30
3/3 ————— 10s 2s/step - accuracy: 0.9722 - loss: 0.5315 - val_accuracy: 0.8214 - val_loss: 3.5596
Epoch 19/30
3/3 ————— 10s 2s/step - accuracy: 1.0000 - loss: 0.0017 - val_accuracy: 0.7857 - val_loss: 3.7413
Epoch 20/30
3/3 ————— 10s 2s/step - accuracy: 1.0000 - loss: 2.2164e-04 - val_accuracy: 0.7500 - val_loss: 3.9702
Epoch 21/30
3/3 ————— 10s 2s/step - accuracy: 0.9910 - loss: 0.0373 - val_accuracy: 0.7500 - val_loss: 4.3804
Epoch 22/30
3/3 ————— 10s 2s/step - accuracy: 1.0000 - loss: 3.7460e-06 - val_accuracy: 0.7500 - val_loss: 5.1684
Epoch 23/30
3/3 ————— 10s 2s/step - accuracy: 0.9924 - loss: 0.0477 - val_accuracy: 0.7500 - val_loss: 5.1877
Epoch 24/30
3/3 ————— 10s 2s/step - accuracy: 1.0000 - loss: 0.0012 - val_accuracy: 0.7500 - val_loss: 5.0529
Epoch 25/30
3/3 ————— 5s 2s/step - accuracy: 1.0000 - loss: 3.8805e-04 - val_accuracy: 0.7500 - val_loss: 4.8808
Epoch 26/30
3/3 ————— 11s 2s/step - accuracy: 1.0000 - loss: 7.5922e-06 - val_accuracy: 0.7500 - val_loss: 4.7377
Epoch 27/30
3/3 ————— 10s 2s/step - accuracy: 1.0000 - loss: 0.0034 - val_accuracy: 0.7500 - val_loss: 4.5563
Epoch 28/30
3/3 ————— 5s 2s/step - accuracy: 0.9816 - loss: 0.0678 - val_accuracy: 0.7857 - val_loss: 4.2608
Epoch 29/30
3/3 ————— 10s 2s/step - accuracy: 1.0000 - loss: 1.8480e-04 - val_accuracy: 0.7857 - val_loss: 4.5831
Epoch 30/30
3/3 ————— 5s 2s/step - accuracy: 0.9785 - loss: 0.2007 - val_accuracy: 0.7857 - val_loss: 4.5258
```

Figure 5.4: Test Result of Brain Tumor Detection

Fig 5.4 represent the Test Result of Brain Tumor Detection in the Integration Testing.

5.2.3 System Testing

System testing is all about the testing of a complete and fully integrated software.

Input

```
1 from matplotlib.pyplot import imshow
2 img = Image.open(r"../input/brain-mri-images-for-brain-tumor-detection/no/N17.jpg")
3 x = np.array(img.resize((128,128)))
4 x = x.reshape(1,128,128,3)
5 res = model.predict_on_batch(x)
6 classification = np.where(res == np.amax(res))[1][0]
7 imshow(img)
8 print(str(res[0][classification]*100) + '% Confidence This Is ' + names[classification])
```

Test Result

```
/opt/conda/lib/python3.10/site-packages/keras/src/layers/convolutional/base_conv.py:99: Use
super().__init__(
Model: "sequential"
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 128, 128, 32)	416
conv2d_1 (Conv2D)	(None, 128, 128, 32)	4,128
batch_normalization (BatchNormalization)	(None, 128, 128, 32)	128
max_pooling2d (MaxPooling2D)	(None, 64, 64, 32)	0
dropout (Dropout)	(None, 64, 64, 32)	0
conv2d_2 (Conv2D)	(None, 64, 64, 64)	8,256
conv2d_3 (Conv2D)	(None, 64, 64, 64)	16,448
batch_normalization_1 (BatchNormalization)	(None, 64, 64, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 64)	0
dropout_1 (Dropout)	(None, 32, 32, 64)	0
flatten (Flatten)	(None, 65536)	0

Figure 5.5: Test Result of Brain Tumor Detection

Fig 5.5 represent the Test Result of Brain Tumor Detection in the System Testing.

5.2.4 Test Result

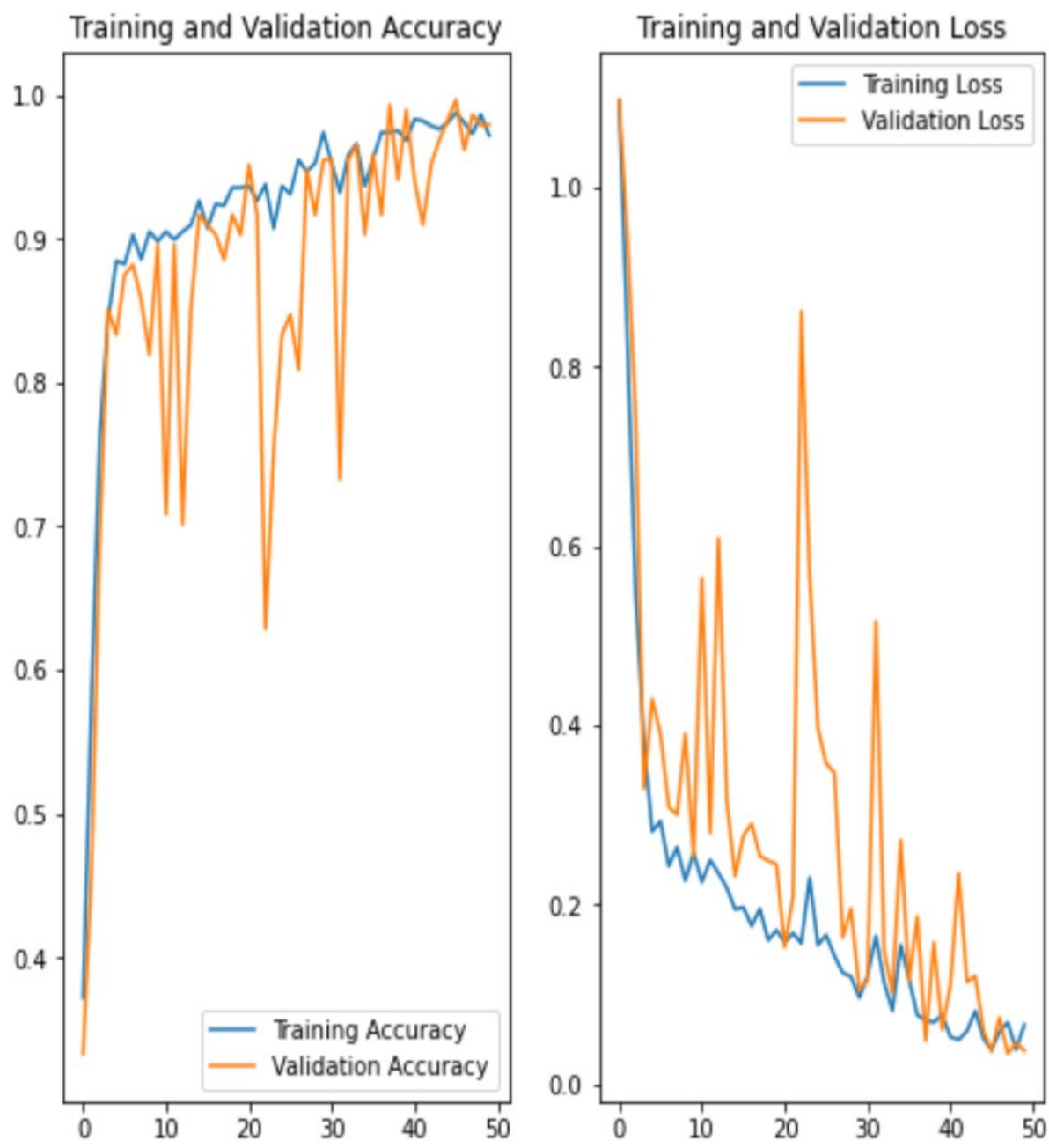


Figure 5.6: Test Result of Training and Validation of Brain Tumor

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The suggested system for using MRI scans to detect and classify brain tumors shows a thorough methodology that includes pre-processing, picture segmentation, feature extraction, and image classification. After experimenting with different segmentation techniques, it was discovered that the region-growing strategy performs better than edge- and boundary-based approaches, especially when paired with the particle swarm optimization algorithm. This combination produces tumors that are precisely segmented, which is essential for accurate diagnosis and therapy planning.

Moreover, the application of feature extraction methods, including the Gray-Level Co-occurrence Matrix (GLCM) approach, improves efficiency by utilizing a variety of features to capture minute details of tumors. Convolutional neural networks (CNNs) proved to be the most successful approach in terms of classification, attaining the highest classification accuracy. Considering how important tumor diagnosis is and how it directly affects patient outcomes,

6.2 Comparison of Existing and Proposed System

Existing system

Pre-processing procedures in the current system could involve standard tasks like noise reduction, image scaling, and normalization. Simple thresholding or edge detection techniques are often used for image segmentation, but they may not be able to precisely capture the complex borders of tumors, which could result in segmentation errors and inaccuracies. The intricacy of tumor traits may not be adequately captured by handcrafted features used in feature extraction approaches, such as texture or intensity histograms.

Conventional machine learning algorithms like random forests and support vector machines (SVM) are frequently used for categorization. Although these techniques can yield respectable outcomes, their efficacy in precisely categorizing tumors may be restricted by the high-dimensional and nonlinear characteristics of data.

Proposed system

When compared to conventional methods, the suggested system for brain tumor identification and classification utilizing machine learning more particularly, Convolutional Neural Networks (CNN) offers a more sophisticated and efficient way. Because CNNs can automatically learn hierarchical features from raw data, they have demonstrated extraordinary performance in a variety of image processing tasks, including medical picture analysis. CNNs are used in the proposed system for tasks related to both image segmentation and classification. The CNN architecture is taught to precisely and automatically recognize and define tumor locations in MRI images in order to perform segmentation. Even in situations where tumors have irregular shapes or subtle boundaries, this method enables for more accurate and consistent tumor delineation by doing away with the requirement for manual segmentation.

Moreover, CNNs are used in tumor classification, where they are trained to automatically extract pertinent features from MRI pictures without the requirement for human-crafted features. The CNN model gains the ability to discriminate between various tumor types and precisely categorize tumors into their corresponding categories by training on a sizable dataset of labeled MRI images. By using this method, classification accuracy is increased while also allowing the system to adapt and generalize effectively to data that hasn't been seen before. Additionally, the suggested approach can profit from transfer learning methods, which involve honing CNN models that have already been trained on a variety of medical imaging datasets to the particular job of brain tumor identification and categorization. By using this method, the system can further improve by utilizing the representations and knowledge acquired from jobs that are similar to

6.3 Sample Code

```
1
2 import os
3 import keras
4 from keras.models import Sequential
5 from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout, BatchNormalization
6 from PIL import Image
7 import numpy as np
8 import pandas as pd
9 import matplotlib.pyplot as plt
10 plt.style.use('dark_background')
11 from sklearn.model_selection import train_test_split
12 from sklearn.preprocessing import OneHotEncoder
13 data = []
14 paths = []
15 result = []
16
17 for r, d, f in os.walk(r'/kaggle/input/brain-mri-images-for-brain-tumor-detection/yes'):
18     for file in f:
19         if '.jpg' in file:
20             paths.append(os.path.join(r, file))
21
22 for path in paths:
23     img = Image.open(path)
24     img = img.resize((128,128))
25     img = np.array(img)
26     if(img.shape == (128,128,3)):
27         data.append(np.array(img))
28         result.append(encoder.transform([[0]]).toarray())
29 paths = []
30 for r, d, f in os.walk(r'/kaggle/input/brain-mri-images-for-brain-tumor-detection/no'):
31     for file in f:
32         if '.jpg' in file:
33             paths.append(os.path.join(r, file))
34
35 for path in paths:
36     img = Image.open(path)
37     img = img.resize((128,128))
38     img = np.array(img)
39     if(img.shape == (128,128,3)):
40         data.append(np.array(img))
41         result.append(encoder.transform([[1]]).toarray())
42 import matplotlib.pyplot as plt
43
44 # Count the number of Tumor and Normal samples
45 tumor_count = np.sum(result[:, 0] == 1)
46 normal_count = np.sum(result[:, 1] == 1)
47
```



```

48 # Create labels and counts for the bar plot
49 labels = ['Tumor', 'Normal']
50 counts = [tumor_count, normal_count]
51
52 # Example colors that are complementary and visually pleasing
53 colors = ['skyblue', 'lightcoral']
54 total_samples = len(result)
55 tumor_percentage = (tumor_count / total_samples) * 100
56 normal_percentage = (normal_count / total_samples) * 100
57
58 # Create labels and counts for the pie chart
59 labels = ['Tumor', 'Normal']
60 counts = [tumor_percentage, normal_percentage]
61 colors = ['mediumseagreen', 'lightcoral']
62
63 # Create a pie chart
64 plt.pie(counts, labels=labels, autopct='%1.1f%%', startangle=90, colors=colors)
65 plt.title('Distribution of Tumor and Normal Samples')
66 plt.show()
67 def names(number):
68     if number==0:
69         return 'Its a Tumor'
70     else:
71         return 'No, Its not a tumor'
72 from matplotlib.pyplot import imshow
73 img = Image.open(r"../input/brain-mri-images-for-brain-tumor-detection/no/N17.jpg")
74 x = np.array(img.resize((128,128)))
75 x = x.reshape(1,128,128,3)
76 res = model.predict_on_batch(x)
77 classification = np.where(res == np.amax(res))[1][0]
78 imshow(img)
79 print(str(res[0][classification]*100) + '% Confidence This Is ' + names(classification))
80 from matplotlib.pyplot import imshow
81 img = Image.open(r"../input/brain-mri-images-for-brain-tumor-detection/yes/Y3.jpg")
82 x = np.array(img.resize((128,128)))
83 x = x.reshape(1,128,128,3)
84 res = model.predict_on_batch(x)
85 classification = np.where(res == np.amax(res))[1][0]
86 imshow(img)
87 print(str(res[0][classification]*100) + '% Confidence This Is A , ' + names(classification))

```

Output

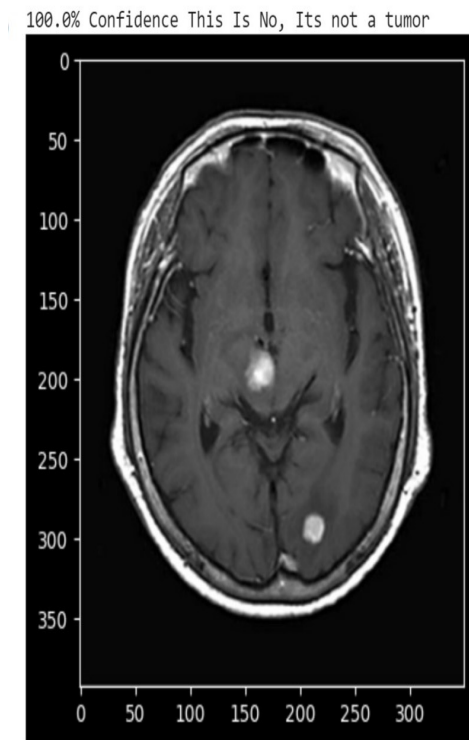


Figure 6.1: **Output 1 of Brain Tumor Detection**

Fig 6.1 Shows the output as the prediction form in which it represents the accuracy score for the algorithm i.e. Convolutional neural Network. Here we can also observe different type of predictions.

Output

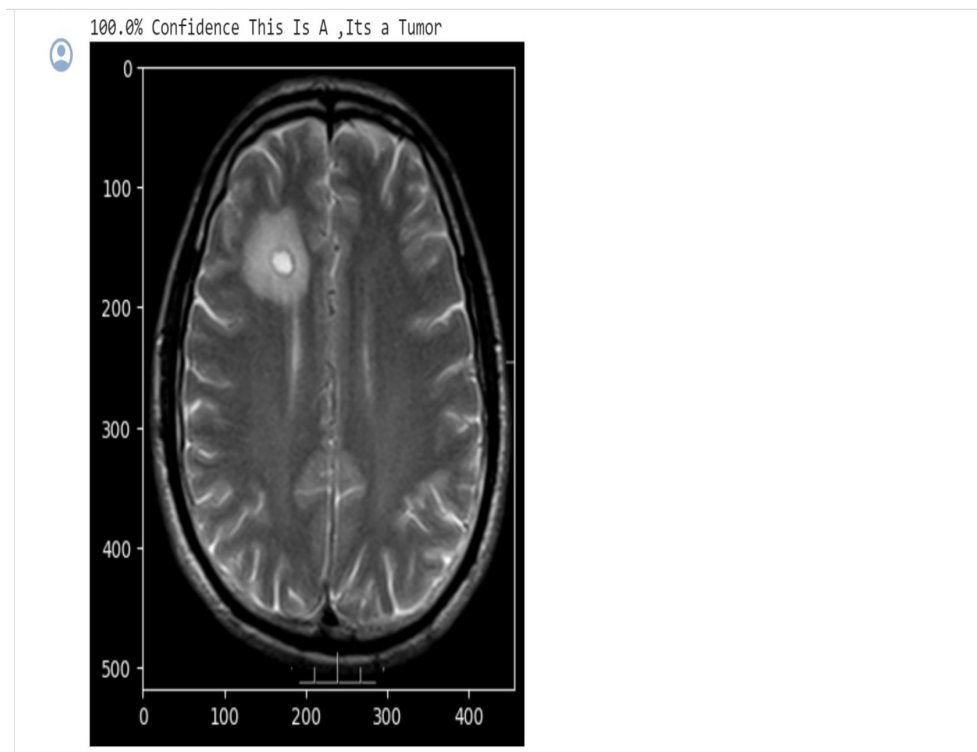


Figure 6.2: **Output 2 of Brain Tumor Detection**

Fig 6.2 Shows the output as the prediction form in which it represents the accuracy score for the algorithm i.e. Convolutional neural Network. Here we can also observe different type of predictions.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

Abnormal growth of tissue in the brain which affect the normal functioning of the brain is considered a brain tumor. The main goal of medical image processing is to identify accurate and meaningful information using algorithms with minimum error possible. Brain tumor detection and classification through MRI images can be categorized into four different sections: pre-processing, image segmentation, feature extraction and image classification. Various segmentation methodologies are explored in the project. It can be concluded that the algorithms and the parameters used in the proposed system are all meant to increase the efficiency of the system by achieving better results. Thus, the proposed methodology which helps in increasing the accuracy of 93% and obtaining the Brain Tumor Prediction

7.2 Future Enhancements

Encouraged by these results, future work will involve the improvement of classification result and overall accuracy. The number of output classes can also be increased if more data is available. With a more extensive and diverse dataset, the overall classification accuracy can be dramatically increased. Another approach to improve the result would be to increase the number of hidden layers of the neural network. By increasing the number of hidden layers, the weights will be better adjusted and thus increase the classification. One can also do fine tuning and transfer learning approaches to better tune the model on the basis of already trained models.

Chapter 8

PLAGIARISM REPORT

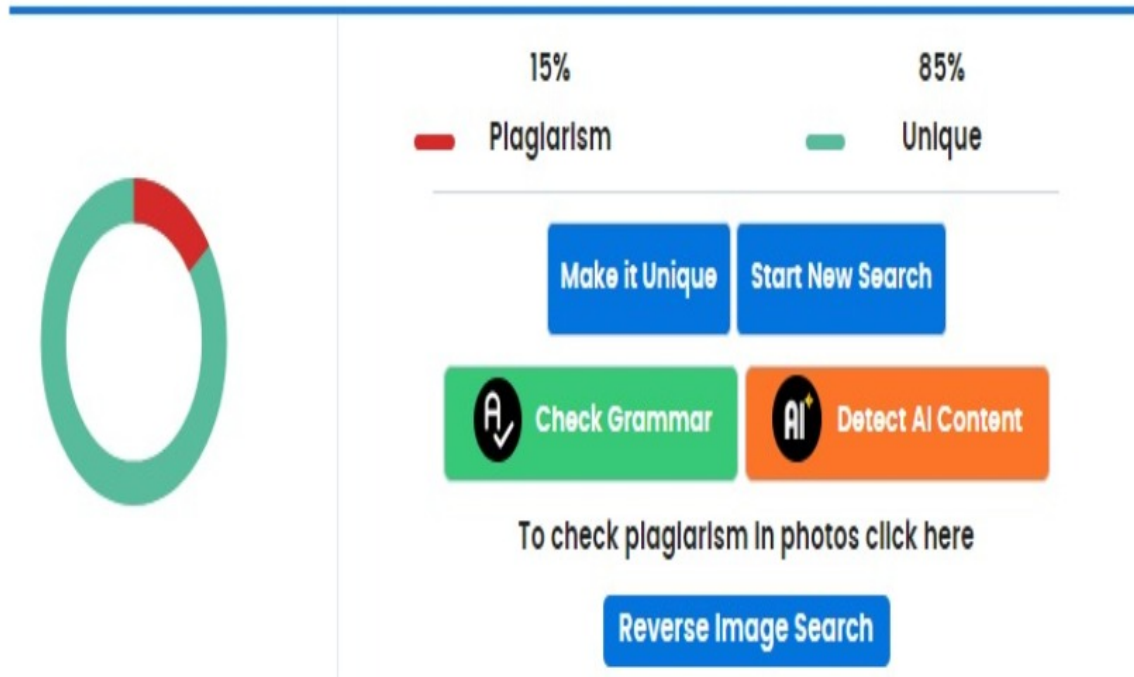


Figure 8.1: Plagiarism Report

Chapter 9

SOURCE CODE & POSTER PRESENTATION

9.1 Source Code

```
1
2 import os
3 import keras
4 from keras.models import Sequential
5 from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout, BatchNormalization
6 from PIL import Image
7 import numpy as np
8 import pandas as pd
9 import matplotlib.pyplot as plt
10 plt.style.use('dark_background')
11 from sklearn.model_selection import train_test_split
12 from sklearn.preprocessing import OneHotEncoder
13 data = []
14 paths = []
15 result = []
16
17 for r, d, f in os.walk(r'/kaggle/input/brain-mri-images-for-brain-tumor-detection/yes'):
18     for file in f:
19         if '.jpg' in file:
20             paths.append(os.path.join(r, file))
21
22 for path in paths:
23     img = Image.open(path)
24     img = img.resize((128,128))
25     img = np.array(img)
26     if (img.shape == (128,128,3)):
27         data.append(np.array(img))
28         result.append(encoder.transform([[0]]).toarray())
29 paths = []
30 for r, d, f in os.walk(r'/kaggle/input/brain-mri-images-for-brain-tumor-detection/no'):
31     for file in f:
32         if '.jpg' in file:
33             paths.append(os.path.join(r, file))
34
35 for path in paths:
```

```

36     img = Image.open(path)
37     img = img.resize((128,128))
38     img = np.array(img)
39     if (img.shape == (128,128,3)):
40         data.append(np.array(img))
41         result.append(encoder.transform([[1]]).toarray())
42 import matplotlib.pyplot as plt
43
44 # Count the number of Tumor and Normal samples
45 tumor_count = np.sum(result[:, 0] == 1)
46 normal_count = np.sum(result[:, 1] == 1)
47
48 # Create labels and counts for the bar plot
49 labels = ['Tumor', 'Normal']
50 counts = [tumor_count, normal_count]
51
52 # Example colors that are complementary and visually pleasing
53 colors = ['skyblue', 'lightcoral']
54 total_samples = len(result)
55 tumor_percentage = (tumor_count / total_samples) * 100
56 normal_percentage = (normal_count / total_samples) * 100
57
58 # Create labels and counts for the pie chart
59 labels = ['Tumor', 'Normal']
60 counts = [tumor_percentage, normal_percentage]
61 colors = ['mediumseagreen', 'lightcoral']
62
63 # Create a pie chart
64 plt.pie(counts, labels=labels, autopct='%1.1f%%', startangle=90, colors=colors)
65 plt.title('Distribution of Tumor and Normal Samples')
66 plt.show()
67 def names(number):
68     if number==0:
69         return 'Its a Tumor'
70     else:
71         return 'No, Its not a tumor'
72 from matplotlib.pyplot import imshow
73 img = Image.open(r"../input/brain-mri-images-for-brain-tumor-detection/no/N17.jpg")
74 x = np.array(img.resize((128,128)))
75 x = x.reshape(1,128,128,3)
76 res = model.predict_on_batch(x)
77 classification = np.where(res == np.amax(res))[1][0]
78 imshow(img)
79 print(str(res[0][classification]*100) + '% Confidence This Is ' + names(classification))
80 from matplotlib.pyplot import imshow
81 img = Image.open(r"../input/brain-mri-images-for-brain-tumor-detection/yes/Y3.jpg")
82 x = np.array(img.resize((128,128)))
83 x = x.reshape(1,128,128,3)
84 res = model.predict_on_batch(x)
85 classification = np.where(res == np.amax(res))[1][0]

```

```

86 imshow(img)
87 print(str(res[0][classification]*100) + '% Confidence This Is A ,' + names(classification))
88 from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score
89
90 # 'x_test', 'y_test' are test data
91 y_pred = model.predict(x_test)
92
93 # Convert probabilities to class labels
94 y_pred_classes = np.argmax(y_pred, axis=1)
95 y_true_classes = np.argmax(y_test, axis=1)
96
97 # Calculate precision, recall, and F1 score
98 accuracy = accuracy_score(y_true_classes, y_pred_classes)
99 precision = precision_score(y_true_classes, y_pred_classes)
100 recall = recall_score(y_true_classes, y_pred_classes)
101 f1 = f1_score(y_true_classes, y_pred_classes)
102
103 print("Accuracy: {:.4f}".format(accuracy))
104 print("Precision: {:.4f}".format(precision))
105 print("Recall: {:.4f}".format(recall))
106 print("F1 Score: {:.4f}".format(f1))

```


9.2 Poster Presentation

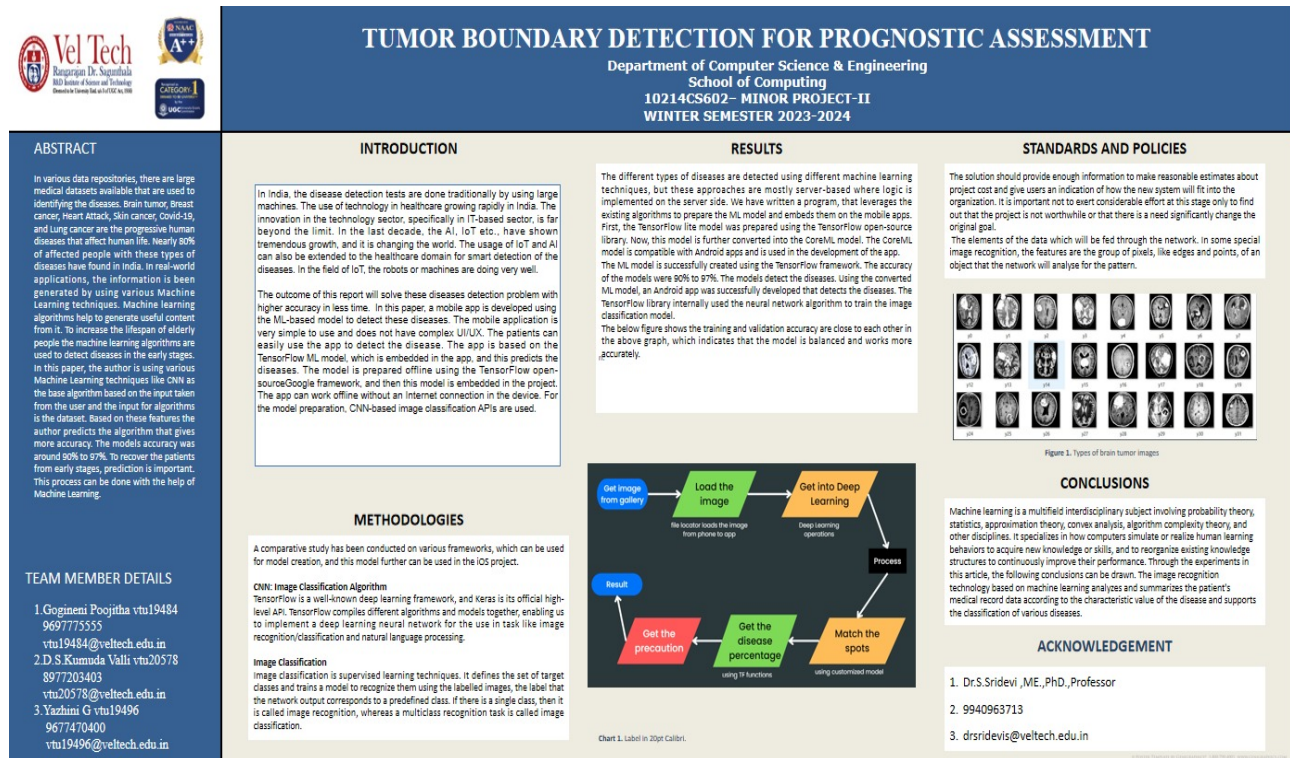


Figure 9.1: Poster Presentation

References

- [1] A. Ladgham, G. Torkhani, A. Sakly, and A. Mtibaa.(2019).Modified support vector machines for MR brain images recognition, International Conference on Control, Decision and Information Technologies (CoDIT).
- [2]P. Su, Z. Xue, L. Chi, J. Yang, and S. T. Wong.(2020). Support vector machine (SVM) active learning for automated Glioblastoma segmentation, 9th IEEE International Symposium on Biomedical Imaging (ISBI).
- [3] Y. Liu and Q. Zhao.(2020). An improved watershed algorithm based on multi- scale gradient and distance transformation, 3rd International Congress on Image and Signal Processing.
- [4] MM. Al-Ayyoub, G. Husari, O. Darwish, and A. Alabed-Alaziz.(2020). Machine learning approach for brain tumor detection,” Proceedings of the 3rd International Conference on Information and Communication Systems - ICICS 12.
- [5]E. Küçükkulahli, P. Erdoğan, and K. Polat.(2020).Brain MRI Segmentation based on Different Clustering Algorithms,International Journal of Computer Applications, vol. 155, no. 3, pp. 37–40, 2016.Mohsen, Fahd, et al. "A new image segmentation method based on particle swarm optimization." Int. Arab J. Inf. Technol. 9.5.
- [6] Mohsen, Fahd, et al.(2020).A new image segmentation method based on particle swarm optimization. Int. Arab J. Inf. Technol. 9.5 487- 493.
- [7] R. V. Rao, Jaya.(2021). A simple and new optimization algorithm for solving constrained and unconstrained optimization problems, International Journal of Industrial Engineering Computations, pp. 18–3.

- [8] J. Han, M. Kamber, and J. Pei.(2021). Data mining: concepts and techniques.
- [9] I. Maiti and M. Chakraborty .(2021).A new method for brain tumor segmentation based on watershed and edge detection algorithms in HSV colour model, National Conference On Computing And Communication Systems.
- [10] Y. K. Dubey and M. M. Mushrif.(2021).Segmentation of brain MR images using intuitionistic fuzzy clustering algorithm, Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing - ICVGIP 12.
- [11] K. Hu, G.-Y. Tang, D.-P. Xiong, and Q. Qiu.(2021). A novel image segmentation algorithm based on Hidden Markov Random Field model and Finite Mixture Model parameter estimation, International Conference on Wavelet Analysis and Pattern Recognition.
- [12] Kaur, Mandhir, and Rinkesh Mittal .(2021).Survey of Intelligent Methods for Brain Tumor Detection. International Journal of Computer Science Issues (IJCSI) 11.5.
- [13] S. M. K. Hasan, M. Ahmad, and S. D. Ghosh.(2022).Perceptive Proposition of Combined Boosted Algorithm for Brain Tumor Segmentation,Proceedings of the International Conference on Advances in Information Communication Technology Computing - AICTC 16.
- [14] D. Sridhar and I. V. M. Krishna.(2022).Brain Tumor Classification using Discrete Cosine Transform and Probabilistic Neural Network, International Conference on Signal Processing , Image Processing Pattern Recognition.
- [15]H. Li and Y. Fan.(2023).Label propagation with robust initialization for brain tumor segmentation, 9th IEEE International Symposium on Biomedical Imaging (ISBI).

