IDENTIFICATION OF BRAIN TUMOR USING IMAGE SEGMENTATION

A PROJECT REPORT

Submitted by

KISHORE.T.C	20EUIT068
KRITHIKA.N	20EUIT071
LOGA BHARATHI.M	20EUIT074
RAHUL.M.S	20EUIT114

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution, Affiliated to Anna University Chennai - 600 025)

APRIL 2023

SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY
(An Autonomous Institution. Affiliated to Anna University, Chennai)
Kuniamuthur, Coimbatore - 641 008

BONAFIDE CERTIFICATE

Certified that this project report **“IDENTIFICATION OF BRAIN TUMOR USING IMAGE SEGMENTATION”** is the bonafide work of **KISHORE.T.C (20EUIT068), KRITHIKA.N (20EUIT071), LOGA BHARATHI. M (20EUIT074), RAHUL.M.S(20EUIT114)** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
[Dr.Susila.N]	[Dr.A.Kousalya]
HEAD OF THE DEPARTMENT	**ASSOCIATE PROFESSOR**
Information Technology Sri Krishna College of Engineering & Technology, Coimbatore-641008.	Information Technology Sri Krishna College of Engineering & Technology, Coimbatore-641008.

Submitted for the Project viva-voice examination held on_______

INTERNAL EXAMINER   EXTERNAL EXAMINER

ABSTRACT

The human brain is a complex organ that controls the entire body system. However, the abnormal growth and division of cells in the brain can lead to the development of brain tumors. Early detection and diagnosis of brain tumors are crucial for successful treatment and better patient outcomes. In recent years, computer vision techniques have played a significant role in reducing human error and providing accurate results in the diagnosis of brain tumors. Magnetic resonance imaging (MRI) is the most reliable and secure imaging method for detecting brain tumors, as it can detect even the minutest of details.

The objective of our project is to focus on the use of different techniques for the detection of brain cancer using brain MRI. In this study, we have performed pre-processing of the MRI images using the bilateral filter (BF) to remove any noise present in the image. This was followed by binary thresholding and Convolutional Neural Network (CNN) segmentation techniques for the reliable detection of the tumor region. The training, testing, and validation datasets were used to train and evaluate the performance of the CNN model. Based on our model, we can predict whether the subject has a brain tumor or not.

To evaluate the performance of our proposed system, we will use various performance metrics such as accuracy, sensitivity, and specificity. It is expected that our proposed system will exhibit better performance compared to existing methods for brain tumor detection.

In conclusion, the proposed system will provide an automated and reliable tool for the early detection and diagnosis of brain tumors. This will help in better patient outcomes and treatment decisions.

ACKNOWLEDGEMENT

At this juncture, we take the opportunity to convey our sincere thanks and gratitude to the management of the college for providing all the facilities to us.

We wish to convey our gratitude to our college principal**, Dr.J.Janet**, for forwarding us to do our project and offering adequate duration to complete our project.

We would like to express our grateful thanks to **Dr.N.Susila**, Head of the department, Department of Information Technology for her encouragement and valuable guidance to this project.

We extend my gratitude to our beloved guide **Dr.A.KOUSALYA** Associate Professor, Department of Information Technology for her constant support and immense help at all stages of the project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

SYMBOLS	EXPANSION
CNN	CONVOLUTIONAL NEURAL NETWORK
NPDR	NON-PROLIFERATIVE DIABETIC RETINOPATHYPROLIFERATIVE
RGB	RED GREEN BLUE
SVM	SUPPORT VECTOR MACHINES
VOG	VIDEO-OCULOGRAPHY
LBP	LOCAL BINARY PATTERN
ANN	ARTIFICIAL NEURAL NETWORK
CT	COMPUTED TOMOGRAPHY
MRI	MAGNETIC RESONANCE IMAGING
CNS	CENTRAL NEURAL SYSTEM
DFD	DATA FLOW DIAGRAM
HTML	HYPERTEXT MARKUP LANGUAGE
OS	OPERATING SYSTEM
ML	MACHINE LEARNING

CHAPTER - 1

INTRODUCTION

Medical imaging is the technique and process of creating visual representations of the interior of a body for clinical analysis and medical intervention, as well as visual representation of the function of some organs or tissues. Medical imaging seeks to reveal internal structures hidden by the skin and bones, as well as to diagnose and treat disease. Medical imaging also establishes a database of normal anatomy and physiology to make it possible to identify abnormalities. The medical imaging processing refers to handling images by using the computer. This processing includes many types of techniques and operations such as image gaining, storage, presentation, and communication. This process pursues the disorder identification and management. This process creates a data bank of the regular structure and function of the organs to make it easy to recognize the anomalies. This process includes both organic and radiological imaging which used electromagnetic energies (X-rays and gamma), sonography, magnetic, scopes, and thermal and isotope imaging. There are many other technologies used to record information about the location and function of the body. Those techniques have many limitations compared to those modulates which produce images. An image processing technique is the usage of a computer to manipulate the digital image. This technique has many benefits such as elasticity, adaptability, data storing, and communication. With the growth of different image resizing techniques, the images can be kept efficiently. This technique has many sets of rules to perform in the images synchronously. The 2D and 3D images can be processed in multiple dimensions

1.1 OVERVIEW :

A brain tumor is a growth of cells in the brain or near it. Brain tumors can happen in the brain tissue. Brain tumors also can happen near the brain tissue. Nearby locations include nerves, the pituitary gland, the pineal gland, and the membranes that cover the surface of the brain. Brain tumors can begin in the brain. These are called primary brain tumors. Sometimes, cancer spreads to the brain from other parts of the body. These tumors are secondary brain tumors, also called metastatic brain tumors. Many different types of primary brain tumors exist. Some brain tumors aren't cancerous. These are called noncancerous brain tumors or benign brain tumors. Noncancerous brain tumors may grow over time and press on the brain tissue. Other brain tumors are brain cancers, also called malignant brain tumors. Brain cancers may grow quickly. The cancer cells can invade and destroy the brain tissue. Brain tumors range in size from very small to very large. Some brain tumors are found when they are very small because they cause symptoms that you notice right away. Other brain tumors grow very large before they're found. Some parts of the brain are less active than others. If a brain tumor starts in a part of the brain that's less active, it might not cause symptoms right away. The brain tumor size could become quite large before the tumor is detected. Brain tumor treatment options depend on the type of brain tumor you have, as well as its size and location.

1.2 LITERATURE REVIEW:

1. **A NOVEL BASED APPROACH FOR EXTRACTION OF BRAIN TUMOR IN MRI IMAGES USING SOFT COMPUTING TECHNIQUES:**

Authors: A. Sivaramakrishnan And Dr. M. Karnan

Abstract:

A. Sivaramakrishnan et al. (2013) [1] projected an efficient and innovative discovery of the brain tumor vicinity from an image that turned into finished using the Fuzzy Capproach grouping algorithm and histogram equalization. The disintegration of images is achieved by the usage of principal factor evaluation is done to reduce the extent of the wavelet coefficient. The outcomes of the anticipated FCM clustering algorithm accurately withdrew tumor area from the MR images.

2. **IMPROVED EDGE DETECTION ALGORITHM FOR BRAIN TUMOR SEGMENTATION:**

Authors: Asra Aslam, Ekram Khan, M.M. Sufyan Beg

Abstract:

M. M. Sufyan et al. [2] has presented a detection using enhanced edge technique for brain-tumor segmentation that mainly relied on Sobel feature detection. Their presented work associates the binary thresholding operation with the Sobel approach and excavates diverse extents using a secure contour process. After the completion of that process, cancer cells are extracted from the obtained picture using intensity values.

3. **IMAGE SEGMENATION USING CLUSTERING METHODS: PERFORMANCE ANALYSIS.**

Authors: B.Sathya and R.Manavalan,

Abstract:

Sathya et al. (2011) [3], provided a different clustering algorithm such as K-means, Improvised K-means, C-means, and improvised C-means algorithms. Their paper presented an experimental analysis for massive datasets consisting of unique photographs. They analyzed the discovered consequences using numerous parametric tests.

4. IMAGE SEGMENTATION FOR EARLY STAGE BRAIN TUMOR DETECTION USING MATHEMATICAL MORPHOLOGICAL RECONSTRUCTION:

Authors: Devkota, B. & Alsadoon, Abeer & Prasad, P.W.C. & Singh, A.K. & Elchouemi,

Abstract:

B. Devkota et al. [4] have proposed that a computer-aided detection (CAD) approach is used to spot abnormal tissues via Morphological operations. Amongst all different segmentation approaches existing, the morphological opening and closing operations are preferred since it takes less processing time with the utmost efficiency in withdrawing tumor areas with the least faults.

5. INTELLIGENT BRAIN TUMOR LESION CLASSIFICATION AND IDENTIFICATION FROM MRI IMAGES USING A K-NN TECHNIQUE:

Authors: K. Sudharani, T. C. Sarma and K. Satya Rasad

Abstract:

K. Sudharani et al. [5] presented a K- nearest neighbor algorithm to the MR images to identify and confine the hysterically full-fledged part within the abnormal tissues. The proposed work is a sluggish methodology but produces exquisite effects. The accuracy relies upon the sample training phase.

6. A COMPARATIVE ANALYSIS OF THRESHOLDING AND EDGE DETECTION SEGMENTATION TECHNIQUES:

Authors: Kaur, Jaskirat & Agrawal, Sunil & Renu

Abstract:

Jaskirat Kaur et al. (2012) [6] defined a few clustering procedures for the segmentation process and executed an assessment on distinctive styles for those techniques. Kaur represented a scheme to measure selected clustering techniques based on their steadiness in exceptional tenders. They also defined the diverse performance metric tests, such as sensitivity, specificity, and accuracy.

7. **FUSING IMAGES WITH DIFFERENT FOCUSES USING SUPPORT VECTOR MACHINES**

Authors: Li, Shutao, JT-Y. Kwok, IW-H. Tsang and Yaonan Wang

Abstract:

J.T. Kwok et al. [7] delivered wavelet-based photograph fusion to easily cognizance at the object with all focal lengths as several vision-related processing tasks can be carried out more effortlessly when wholly substances within the images are bright. In their work Kwok et al. investigated with different datasets, and results show that presented work is extra correct as it does not get suffering from evenness at different activity stages computations.

8. **A TEXTURE BASED TUMOR DETECTION AND AUTOMATIC SEGMENTATION USING SEEDED REGION GROWING METHOD:**

Authors: M. Kumar and K. K. Mehta,

Abstract:

They highlighted the effects of segmentation if the tumor tissue edges aren't shrill. The performance of the proposed technology may get unwilling results due to those edges. The texture evaluation and seeded region approach turned into executed inside the MATLAB environment.

9. **BRAIN TUMOR DETECTION USING ARTIFICIAL NEURAL NETWORKS. JOURNAL OF SCIENCE AND TECHNOLOGY:**

Authors: Mahmoud, Dalia & Mohamed Eltaher.

Abstract:

They implemented a computerized recognition system for MR imaging the use of Artificial Neural Networks. That was observed that after the Elman community was used during the recognition system, the period time and the 9 accuracy level were high, in comparison with other ANNs systems. This neural community has a sigmoid characteristic which elevated the extent of accuracy of the tumor segmentation.

10. **AN ACCURATE AND EFFICIENT BAYESIAN METHOD FOR AUTOMATIC SEGMENTATION OF BRAIN MRI:**

Authors: Marroquin J.L., Vemuri B.C., Botello S., Calderon F

Abstract:

Using a separate parametric model in preference to a single multiplicative magnificence will lessen the impact on the intensities of a grandeur. Brain atlas is hired to find nonrigid conversion to map the usual brain. This transformation is further used to segment the brain from nonbrain tissues, computing prior probabilities and finding automatic initialization and finally applying the MPM-MAP algorithm to find out optimal segmentation. Major findings from the study show that the MPM-MAP algorithm is comparatively robust than EM in terms of errors while estimating the posterior marginal. For optimal segmentation, the MPM-MAP algorithm involves only the solution of linear systems and is therefore computationally efficient.

11. MR IMAGE CLASSIFICATION USING ADABOOST FOR BRAIN TUMOR TYPE:

Authors: Minz, Astina, and Chandrakant Mahobiya.

Abstract:

Astina minz et al. implemented an operative automatic classification approach for brain image that projected the usage of the AdaBoost gadget mastering algorithm. The proposed system includes three main segments. Pre-processing has eradicated noises in the datasets and converted images into grayscale. Median filtering and thresholding segmentation are implemented in the pre-processed image

12. BRAIN MR IMAGE SEGMENTATION FOR TUMORDETECTION USING ARTIFICIAL NEURAL NETWORKS:

Authors: Monica Subashini.M, Sarat Kumar Sahoo

Abstract:

Monica Subashini and Sarat Kumar Sahoo has suggested a technique for detecting the tumor commencing the brain MR images. They also worked on different techniques, which include pulse-coupled Neural Network and noise removal strategies for reinforcing the mind MRI images and backpropagation network for classifying the brain MRI images from tumor cells. They observed image enhancement and segmentation of the usage of their proposed technique, and the backpropagation network helps in the identification of a tumor in a brain MR image.

13. FUSING IMAGES WITH DIFFERENT FOCUSES USING SUPPORT VECTOR MACHINES:

Authors: S. Li, J.T. Kwok, I.W Tsang, and Y. Wang

Abstract:

Li et al. report that edge detection, image segmentation, and matching are not easy to achieve in optical lenses that have long focal lengths. Previously, researchers have proposed many techniques for this mechanism, one of which is wavelet-based image fusion. The wavelet function can be improved by applying a discrete wavelet frame transform (DWFT) and a support vector machine (SVM). In this paper, the authors experimented with five sets of 256-level images. Experimental results show that this technique is efficient and more accurate as it does not get affected by consistency verification and activity level measurements. However, the paper is limited to only one task related to fusion, and dynamic ranges are not considered during the calculation.

14. THREE-LEVEL IMAGE SEGMENTATION BASED ON MAXIMUM FUZZY PARTITION ENTROPY OF 2-D HISTOGRAM AND QUANTUM GENETIC ALGORITHM, ADVANCED INTELLIGENT COMPUTING THEORIES, AND APPLICATIONS. WITH ASPECTS OF ARTIFICIAL INTELLIGENCE:

Authors: H. Yu and J.L. Fan

Abstract:

That image segmentation is used for extracting meaningful objects from an image. They propose segmenting an image into three parts, including dark, grey and white. Z-function and s-function are used for the fuzzy division of the 2D histogram. Afterward, QGA is used for finding a combination of 12 membership parameters, which have a maximum value. This technique is used to enhance image segmentation and the significance of their work is that three-level image segmentation is used by following the maximum fuzzy partition of 2D Histograms. QGA is selected 11 for the optimal combination of parameters with the fuzzy partition entropy. The proposed method of fuzzy partition entropy of 2D histogram generates better performance than one-dimensional 3-level thresh holding method. Somehow, a large number of possible combinations of 12 parameters in a multi-dimensional fuzzy partition are used, and it is practically not feasible to compute each possible value; therefore, QGA can be used to find the optimal combination.

15. SEGMENTATION AND CLASSIFICATION OF MRI BRAIN TUMOR:

Authors: P.S. Mukambika, K Uma Rani,

Abstract:

Mukambika et al proposed methodology for the subsequent stage's classification of the tumor, whether it is present or not. Their proposed work represents the comparative study of strategies used for tumor identification from MR images, namely the Level set approach and discrete wavelength transforms (DWT) and K-method segmentation algorithms. After that phase, feature extraction is done followed SVM classification.

16. BRAIN TUMOR GRADING BASED ON NEURAL NETWORKS AND CONVOLUTIONAL NEURAL NETWORKS:

Authors: Pan, Yuehao & Huang, Weimin & Lin, Zhiping & Zhu, Wanzheng & Zhou, Jiayin & Wong, Jocelyn & Ding, Zhongxiang.

Abstract:

Yuehao Pan et al., [16] has used brain MRI pix for getting useful statistics for classifying brain tumor. In their proposed method, they used Convolutional Neural Networks (CNN) algorithms for developing a brain tumor detection system. The performance of their CNN report is measured primarily based on sensitivity and specificity parameters, which have stepped forward when in comparison to the Artificial Neural Networks (ANN).

17. BRAIN TUMOR SEGMENTATION USING CONVOLUTIONAL NEURAL NETWORKS IN MRI IMAGES

Author: S. Pereira, A. Pinto, V. Alves, and C. A. Silva

Abstract:

S. Pereira et al. [17] presented that magnetic resonance prevents physical segmentation time in the medical areas. So, an automatic and reliable segmentation technique for identifying abnormal tissues by using Convolutional Neural Network (CNN) had been proposed in the research work. The massive three-dimensional and underlying roughness amongst brain images makes the process of segmenting the image a severe issue, so a robust methodology such as CNN is used.

18. DETECTION AND QUALIFICATION OF BRAIN TUMOR FROM MRI OF BRAIN AND SYMMETRIC ANALYSIS :

Authors: S. Roy And S. K. Bandyopadhyay

Abstract:

Roy et al. (2012) [18] calculated the tumor affected area for proportioned analysis. They confirmed its software with numerous statistics groups with distinctive tumor sizes, intensities, and location. They showed that their algorithm could robotically hit upon and phase the brain tumor from the given photo. Image pre-processing consists of fleeting that pictures to the filtering technique to remove distractors found in given pictures. They first detect the tumor, segment it and then find out the area of tumor. One of the important aspects is that after performing the quantitative analysis, we can identify the status of an increase in the disease. They have suggested multi-step and modular approach to solve the complex MRI segmentation problem. Tumor detection is the first step in tumor segmentation. They have obtained good results in complex situations. The authors claim that MRI segmentation is one of the essential tasks in the medical area but boring and time-consuming if it is performed manually, so visually study of MRI is more interesting and faster.

19. AUTOMATIC TUMOR SEGMENTATION USING CONVOLUTIONAL NEURAL NETWORKS:

Authors: Sankari Ali, and S. Vigneshwari

Abstract:

A. Sankari and S. Vigneshwari [19] has proposed a Convolutional Neural Network (CNN) segmentation, which principally based on the brain tumor classification method. The proposed work used the non-linearity activation feature that's a leaky rectified linear unit (LReLU). They primarily focused on necessary capabilities, which include mean and entropy of the image and analyzed that the CNN algorithm is working higher for representing the complicated and minute capabilities of brain tumor tissues present in the MR Images.

20. SEGMENTATION OF BRAIN TUMOR FROM BRAIN MRI IMAGES REINTRODUCING K – MEANS WITH ADVANCED DUAL LOCALIZATION :

Author: T.U Paul and S.K. Bandyopadhyay

Abstract:

T.U Paul and S.K. Bandyopadhyay [20] has presented the brain segmentation that has automated the use of the Dual Localization technique. In the initial phase, the skull masks are generated for the brain MR images. The tumor areas are improvised using the K-manner procedure. In the final step of their proposed work, they evaluated by its dimensions such as length and breadth.

CHAPTER – 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM:

Support Vector Machine (SVM) is a powerful machine learning algorithm that has been used in various fields, including medical image analysis. One of the most important applications of SVM in the medical field is the prediction of brain tumors using MRI images. This approach is gaining more popularity due to the non-invasive nature of MRI and its high sensitivity in detecting brain tumors.

The process of predicting brain tumors using SVM involves a series of steps. Firstly, a large dataset of MRI images is collected, which includes both normal brain images and images of brains with tumors. Then, these images are preprocessed to remove any noise and enhance their contrast. The next step involves the extraction of relevant features from these images. This is usually done by segmenting the images and identifying the regions of interest, such as the tumor or the surrounding tissue. Features like texture, shape, and intensity are then extracted from these regions.

Once the feature extraction is complete, the data is split into two parts: the training set and the testing set. The training set is used to train the SVM classifier, while the testing set is used to evaluate the performance of the classifier. The SVM algorithm is then used to build a model that can predict whether a new MRI image contains a brain tumor or not. The model is trained to classify the MRI images into two categories: normal brain images and brain images with tumors.

The SVM algorithm works by finding the optimal hyperplane that separates the two classes of data with the largest margin. In other words, the SVM tries to find the decision boundary that maximizes the distance between the two classes of data. This decision boundary is then used to classify new MRI images into either one of the two classes.

The accuracy of the SVM algorithm in predicting brain tumors using MRI images can vary depending on several factors, such as the size and quality of the dataset, the feature extraction method used, and the choice of hyperparameters. Nevertheless, SVM has been shown to be a very effective tool in predicting brain tumors using MRI images.

In conclusion, SVM can be a useful tool in the prediction of brain tumors using MRI images. It has the potential to assist radiologists in accurately diagnosing brain tumors, leading to better treatment outcomes for patients. Future research in this field could focus on improving the accuracy of the algorithm by exploring different feature extraction methods and optimizing the hyperparameters of the SVM.

2.1.1 DRAWBACKS OF EXISTING SYSTEM:

Support Vector Machines (SVMs) have some drawbacks that should be considered when using them for the prediction of brain tumors using MRI images.

One of the main drawbacks of SVM is its sensitivity to the choice of hyperparameters, such as the kernel function and the regularization parameter. These hyperparameters can have a significant impact on the performance of the SVM, and selecting the optimal values requires expertise and trial-and-error. Moreover, SVMs are prone to overfitting if the hyperparameters are not properly tuned, which can result in poor generalization performance when applied to new, unseen data.

Another limitation of SVMs is their computational complexity. SVMs have a quadratic time complexity in the number of training samples, which can make them computationally expensive when dealing with large datasets. This complexity can be further exacerbated by the need for feature extraction and preprocessing steps, which can also be time-consuming.

Furthermore, the accuracy of SVMs is highly dependent on the quality and quantity of the training data. If the dataset is imbalanced, with a much larger proportion of one class than the other, SVMs can be biased towards the majority class and have poor performance on the minority class. Additionally, if the dataset is noisy or contains outliers, the SVM may not perform well, as it tries to find a hyperplane that separates the two classes with the largest margin, and outliers can disrupt this process.

Finally, SVMs are often criticized for being black-box models, meaning that it can be difficult to interpret the decision-making process of the algorithm. While some methods exist to visualize the decision boundaries and identify important features used by the SVM, it can still be challenging to explain the rationale behind the algorithm's predictions.

In summary, while SVMs have proven to be a useful tool in the prediction of brain tumors using MRI images, they do have some limitations that should be taken into account, such as their sensitivity to hyperparameters, computational complexity, and susceptibility to imbalanced or noisy data.

2.2 PROPOSED SYSTEM:

Convolutional Neural Networks (CNNs) have shown to be a more effective tool for the prediction of brain tumors using MRI images compared to Support Vector Machines (SVMs). CNNs can learn more complex and abstract features from the images, and can also automatically extract features from the raw input data, without the need for explicit feature extraction steps. This makes CNNs more efficient and less prone to overfitting compared to SVMs. Additionally, CNNs can also handle images of different sizes and resolutions, which can be important when dealing with medical images that may vary in quality and acquisition settings. Overall, CNNs have the potential to provide more accurate and robust results for the prediction of brain tumors using MRI images.

2.2.1 ADVANTAGES:

- Ability to learn spatial hierarchies of features directly from the input data, without the need for explicit feature engineering.
- Ability to handle large datasets with high-dimensional inputs, such as medical images, through the use of pooling layers that down-sample the output of the convolutional layers.
- High accuracy and performance on a wide range of computer vision tasks, including image classification, object detection, and segmentation.
- Ability to generalize well to new and unseen data, thanks to their ability to learn complex and relevant features from the training data.
- Robustness to noise and variations in the input data, thanks to the use of multiple layers of convolution and pooling.
- Ability to be trained using transfer learning, where a pre-trained model on a large dataset can be fine-tuned on a smaller, more specific dataset, saving computational resources and time.
- Flexibility to be applied to a wide range of applications, including medical image analysis, autonomous driving, and natural language processing.
- Ability to provide real-time results on low-power devices, such as mobile phones and embedded systems, due to their lightweight and efficient architecture.

Overall, CNNs have become a powerful tool for computer vision tasks, offering significant advantages over traditional machine learning algorithms, such as SVMs, and providing high accuracy and performance on a wide range of applications.

2.3 FEASIBILTY STUDY

2.3.1 TECHNICAL FEASIBILITY:

Technical feasibility is a standard practice for companies to conduct feasibility studies before commencing work on a project. Businesses undertake a technical feasibility study to assessthe practicality and viability of a product or service before launching it. Whether you are workingas a product engineer, product designer or team manager, there may be plenty of situations in yourcareer where you are required to prepare a technical feasibility study. In this article, we discuss what technical feasibility is, explain how to conduct one and share tips on writing a feasibility study report. What is technical feasibility, can be described as the formal process of assessing whether it is technically possible to manufacture a product or service. Before launching a new offering or takingup a client project, it is essential to plan and prepare for every step of the operation. Technical feasibility helps determine the efficacy of the proposed plan by analyzing the process, including tools, technology, material, labor and logistics.

A technical feasibility study helps organizations determine whether they have the technical resources to convert the idea into a fully functional and profitable working system. It helps in troubleshooting the project before commencing work. The study identifies potential challenges and uncovers ways to overcome them. It also helps in long-term planning, as it can serve as a flowchart for how products and services evolve before they reach the market.

2.3.2 OPERATIONAL FEASIBILITY:

Operational feasibility studies are generally utilized to answer the following questions:

- **Process** – How do the end-users feel about a new process that may be implemented?
- **Evaluation** – Whether or not the process within the organization will work but also if it *can* work.
- **Implementation** – Stakeholder, manager, and end-user tasks.
- **Resistance** – Evaluate management, team, and individual resistance and how that resistance will be handled.
- **In-House Strategies** – How will the work environment be affected? How much will it change?

- **Adapt & Review** – Once change resistance is overcome, explain how the new process will be implemented along with a review process to monitor the process change.

Example of an Operational Feasibility Study

If an operational feasibility study must answer the six items above, how is it used in the real world?A good example might be if a company has determined that it needs to totally redesign the workspace environment.

After analyzing the technical, economic, and scheduling feasibility studies, next would come the operational analysis. In order to determine if the redesign of the workspace environment would work, an example of an operational feasibility study would follow this path based on six elements:

- **Process** – Input and analysis from everyone the new redesign will affect along with a data matrix on ideas and suggestions from the original plans.
- **Evaluation** – Determinations from the process suggestions; will the redesign benefit everyone? Who is left behind? Who feels threatened?
- **Implementation** – Identify resources both inside and out that will work on the redesign. How will the redesign construction interfere with current work?
- **Resistance** – What areas and individuals will be most resistant? Develop a **change resistance plan**.
- **Strategies** – How will the organization deal with the changed workspace environment? Do new processes or structures need to be reviewed or implemented in order for the redesign tobe effective?

2.3.3 ECONOMICAL FEASIBILITY:

Economic feasibility is a kind of cost-benefit analysis of the examined project, which assesses whether it is possible to implement it. This term means the assessment and analysis of a project's potential to support the decision-making process by objectively and rationally identifying its strengths, weaknesses, opportunities and risks associated with it, the resources that will be needed to implement the project, and an assessment of its chances of success. It consists of market analysis, economic analysis, technical and strategic analysis.

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.

2.3.4 LEGAL FEASIBILITY

This assessment investigates whether any aspect of the proposed project conflicts with legal requirements like zoning laws, data protection acts or social media laws. Let’s say an organization wants to construct a new office building in a specific location. A feasibility study might reveal the organization’s ideal location isn’t zoned for that type of business. That organization has just saved considerable time and effort by learning that their project was not feasible right from the beginning.

CHAPTER – 3

SYSTEM REQUIREMENTS

Defending requirements to establish specifications is the first step in the development of the system. However, in many situations not enough care is taken in establishing correct requirements up front.

3.1 HARDWARE REQUIREMENTS:

Processor	:	Intel Core i3
RAM Capacity	:	4 GB DDR2 RAM
Hard Disk	:	100 GB
Mouse	:	Logical Optical Mouse
Keyboard	:	108 Keys
Monitor	:	17 inch
Speed	:	2GHZ
Floppy Disk Drive	:	2MB

3.2 SOFTWARE REQUIREMENTS:

Back End	:	Flask
OS	:	Windows 07
Front End	:	HTML, CSS

3.3 PLATFORM DESCRIPTION:

3.3.1 Front End : HTML CSS

HTML (Hypertext Markup Language) is a markup language used to create web pages and other documents that can be displayed in a web browser. HTML provides a standard structure for organizing content on the web, including text, images, videos, links, and other elements.
HTML uses tags to mark up different parts of a web page. Tags are enclosed in angle brackets and usually come in pairs, with the opening tag indicating the start of a section of content and the closing tag indicating the end.
CSS (Cascading Style Sheets) is a stylesheet language used to describe the presentation and layout of web pages written in HTML. CSS provides a way to separate the structure and content of a web page from its visual design, allowing developers to create pages that are more flexible and easier to maintain.

FEATURES INTERFACE-html

HTML (Hypertext Markup Language) can be used as an interface for creating web pages and web applications. HTML provides the structure and organization to web content, which can be styled with CSS (Cascading Style Sheets) and made interactive with JavaScript. HTML is versatile and easy to use, making it a popular choice for building user interfaces on the web. The HTML interface can include various elements like text, images, videos, audio, forms, buttons, links, and more. With HTML, developers can create intuitive and user-friendly interfaces for web applications and websites that are accessible to users across different devices and platforms.

3.3.2 Back End: Flask

Flask is a popular Python web framework used for building web applications and APIs. It is a lightweight framework that is easy to learn and use, making it a popular choice for beginners and small projects.
When it comes to the back-end development of a Flask application, Flask provides a number of built-in features that make it easy to handle data, manage user sessions, and create APIs. Flask supports various databases like SQLite, MySQL, and PostgreSQL, making it easy to store and retrieve data.
Flask also supports popular web technologies like HTML, CSS, and JavaScript, and allows

developers to integrate these technologies with Python to build dynamic web applications. Additionally, Flask allows developers to add extensions and plugins to their applications, making it easy to add new functionality and integrate with third-party services.

Overall, Flask provides a simple and efficient way to build back-end functionality for web applications and APIs using Python, and is a popular choice for developers looking to build web applications quickly and easily.

CHAPTER - 4

SYSTEM DESIGN AND DEVELOPMENT

4.1 SYSTEM ARCHITECTURE:

Sure, here's an example of how a CNN can be used for brain tumor detection:

- Input Layer: The input layer takes as input the MRI images of the brain, which are represented as 2D matrices of pixel values. Each pixel value represents the intensity of the corresponding point in the image.
- Convolutional Layer: The convolutional layer applies a set of filters (kernels) to the input MRI images, resulting in a set of feature maps that highlight different parts of the image. For example, some filters might highlight the edges of the tumor, while others might highlight the texture or contrast of the tumor.
- ReLU Layer: The ReLU layer applies the ReLU activation function to the output of the convolutional layer, which introduces non-linearity to the network and helps it learn more complex patterns.
- Pooling Layer: The pooling layer downsamples the output of the ReLU layer, reducing the size of the feature maps and making the network more computationally efficient. For example, max pooling might be used to keep only the maximum value in each pooling region, discarding the rest of the values.
- Dropout Layer: The dropout layer randomly drops out some of the neurons in the network during training, which prevents overfitting and improves generalization.
- Flatten Layer: The flatten layer takes the output of the previous layers and flattens it into a 1-dimensional array.
- Fully Connected Layer: The fully connected layer is a traditional neural network layer where each neuron is connected to every neuron in the previous layer. This layer learns to classify the input MRI images based on the features learned in the previous layers.
- Output Layer: The output layer consists of one or more neurons, depending on the number of classes in the classification task. In this case, there are two classes: brain tumors and normal brain tissue. The output layer therefore consists of two neurons, one for each class. The neuron with the highest output value is chosen as the predicted class for the input MRI

image.

During training, the weights of the filters and neurons in the network are adjusted to minimize the error between the predicted output and the actual output. Once the network is trained, it can be used to classify new MRI images as either brain tumors or normal brain tissue.

4.2 SYSTEM DEVELOPMENT:

A system development lifecycle (SDLC) adheres to important faces that are essential for developers, such as planning, analysis, and implementation and are explained in the section below.A number of system development life cycle models have been created: waterfall, spiral, build andfix, rapid prototyping, incremental, synchronize and stabilize. The oldest of these and the best known is the waterfall model: A sequence of stage in which the output of each becomes the inputfor the next.

4.2.1 Data Pre-processing

Data pre-processing is a crucial step in preparing the MRI image for brain tumor detection. The MRI image is preprocessed to remove any noise or artifacts using various techniques like the bilateral filter to smooth out the image while preserving the edges. The goal is to enhance the image quality and remove any irrelevant features that may interfere with the detection of brain tumors.

In this step, the image is converted into grayscale to simplify the image data and to reduce the computational complexity of the subsequent steps. The grayscale image is then smoothed using the bilateral filter which preserves the edges in the image while removing the noise. The filtered image is then normalized to reduce any intensity variations across the image.

4.2.2 Image Segmentation

Image segmentation is the process of separating the tumor region from the rest of the brain. This can be done using a variety of techniques like thresholding, region-growing, and edge detection. In this step, the CNN model is used to classify each pixel in the image as either tumor or non-tumor.
In the case of thresholding, the filtered image is converted into a binary image by selecting a

threshold value that separates the tumor region from the rest of the brain. The threshold value can be determined by trial and error or by using mathematical techniques like Otsu's method. Once the threshold value is determined, the image is binarized, and the tumor region is extracted.

4.2.3 Model Training

In this step, the CNN model is trained using a dataset of MRI images that have been annotated with tumor regions. The model is trained to recognize the patterns and features that are characteristic of brain tumors in MRI images.

The CNN model is designed to learn the features of the MRI images that distinguish the tumor region from the rest of the brain. The model is trained using a large dataset of MRI images that have been annotated with the tumor region. The annotated images are fed into the CNN model, which learns the features and patterns of the tumor region from the images. The model is trained using various optimization techniques like stochastic gradient descent and backpropagation.

4.2.4 Model Validation

The trained model is then validated using a separate dataset of MRI images that were not used during training. The model's performance is evaluated based on metrics such as accuracy, sensitivity, and specificity.

The validation dataset is used to evaluate the model's performance on new and unseen data. The performance of the model is compared against other state-of-the-art models to determine its accuracy and effectiveness in detecting brain tumors. The model is fine-tuned based on the validation results to improve its performance.

4.3 METHODOLOGY

The methodology used in detecting brain tumors using CNN involves the following steps:

1. Data Collection and Pre-processing: Medical imaging datasets, such as MRI or CT scans, are collected and pre-processed to ensure consistency in image quality and resolution.
2. Data Augmentation: Additional synthetic images are generated by applying transformations to the existing images, such as rotation or scaling, to increase the size of the dataset and improve the generalizability of the model.

3. Model Architecture Design: A CNN architecture is designed with several layers, including convolutional, pooling, and fully connected layers, and the specific hyperparameters, such as learning rate and number of filters, are chosen.
4. Model Training: The designed CNN architecture is trained on the pre-processed dataset, using backpropagation to update the weights of the network and minimize the loss function.
5. Model Evaluation: The trained model is evaluated on a separate validation set to assess its accuracy and generalizability.
6. Model Optimization: The model architecture and hyperparameters are fine-tuned to improve performance on the given dataset, using techniques such as cross-validation and regularization.
7. Testing: The optimized model is then tested on a separate testing set to assess its performance in real-world scenarios.

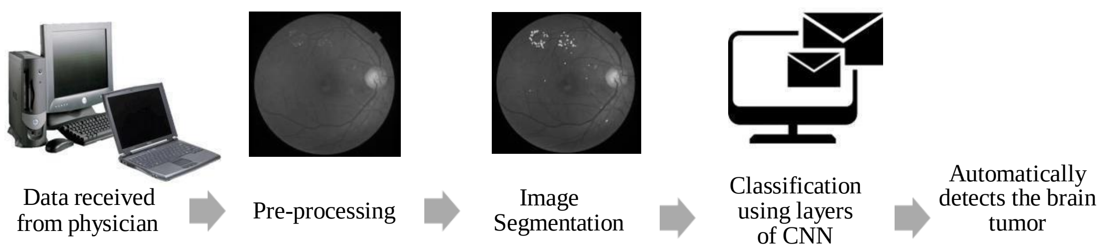
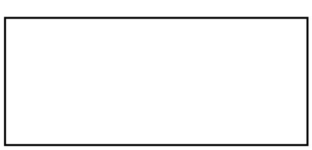


Fig.4.1. Flow Diagram of Methodology

4.4 DATA FLOW DIAGRAM:

A Data Flow Diagram (DFD) is a diagram that describes the flow of data and the processesthat change data throughout a system. It's a structured analysis and design tool that can be used for flowcharting in place of or in association with information. Oriented and process-oriented system flowcharts. Four basic symbols are used to construct data flow diagrams. They are symbolsthat represent data source, data flows, and data transformations and Entity relationship. The points at the data are transformed are represented by enclosed figures, usually circles, they are called nodes.



- **Source or Destination of data**

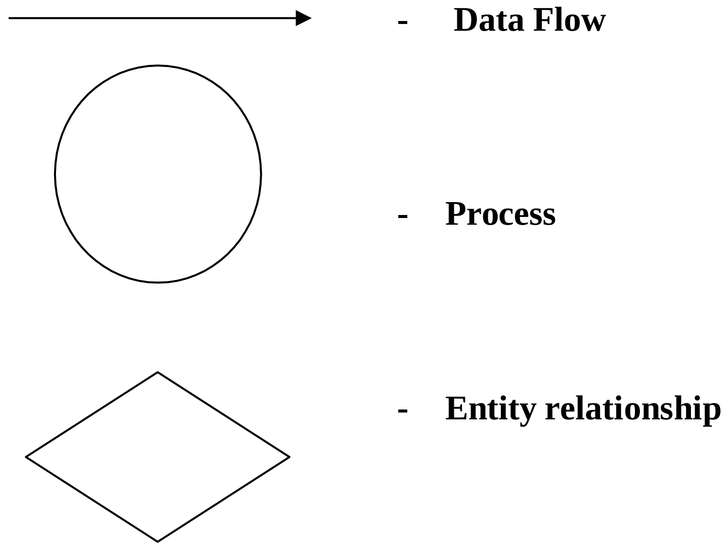


Fig.4.2 Data Flow Diagram Symbols

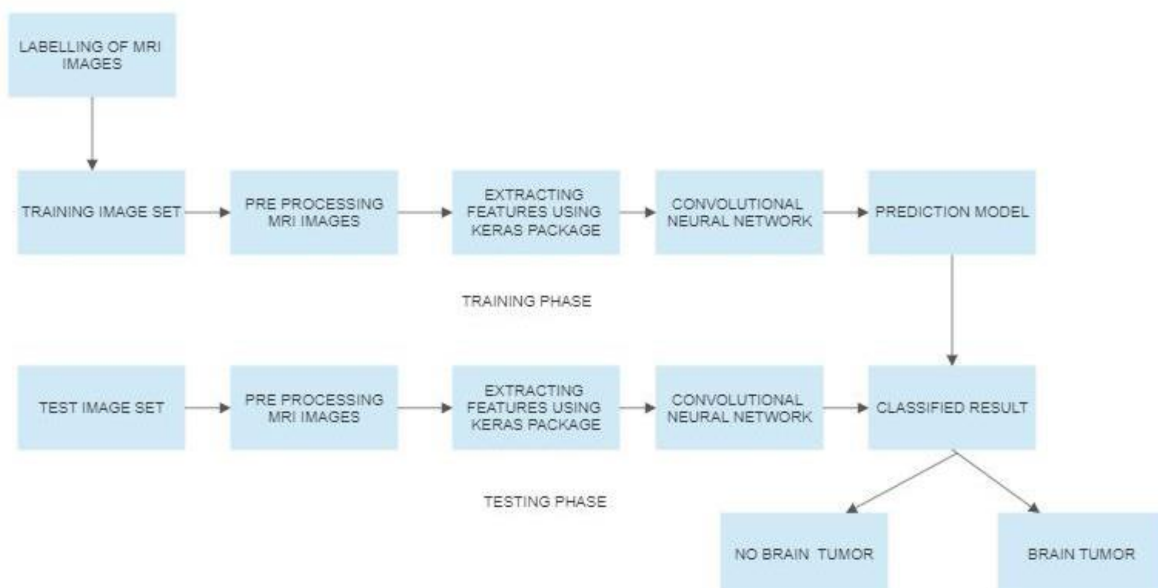


Fig.4.3 Data Flow Diagram

CHAPTER - 5

EXPERIMENTAL RESULTS

5.1 SAMPLE INPUT:

Fig.5.1.1. Home page

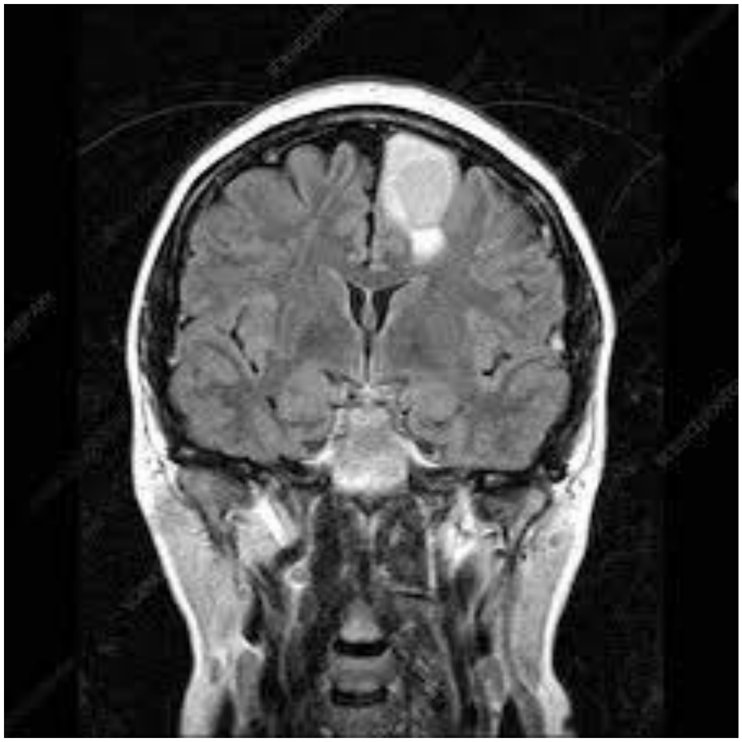


Fig.5.1.2 Sample input

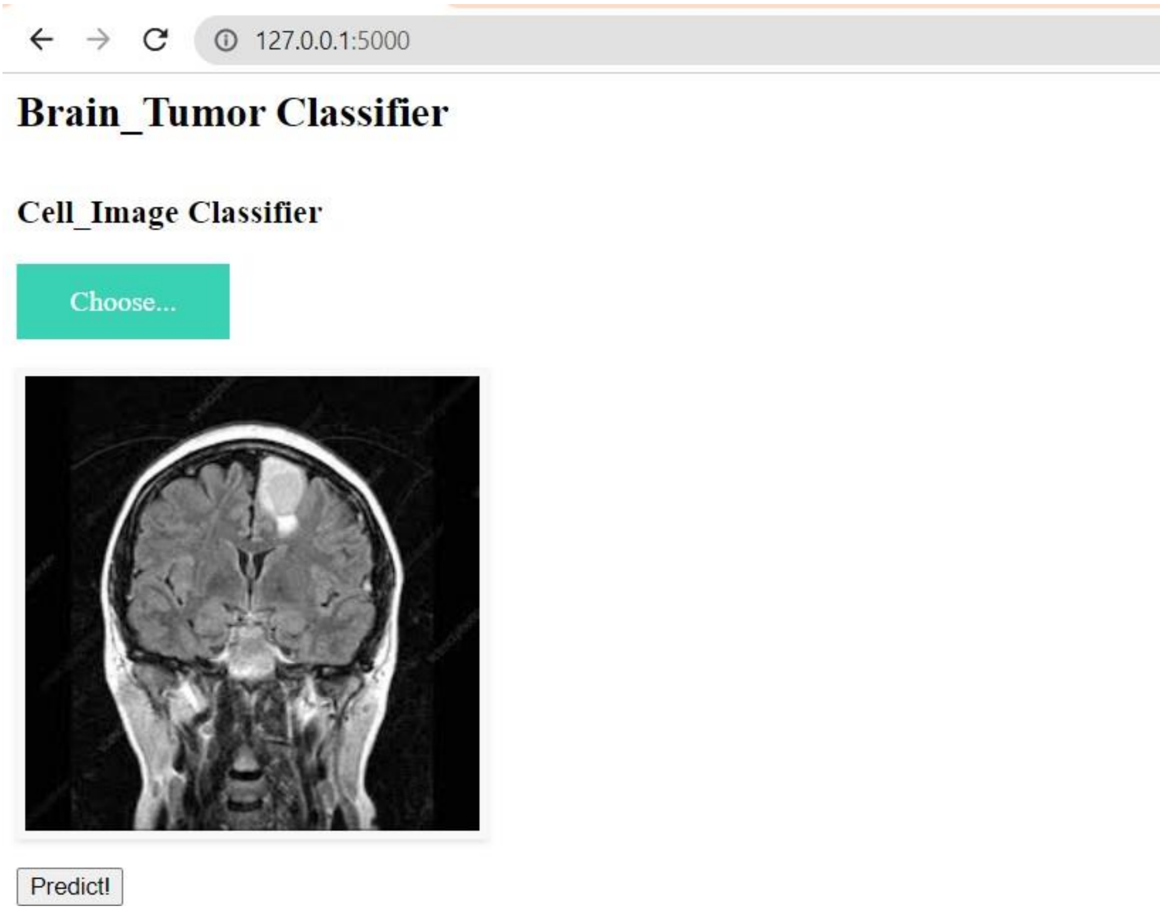


Fig.5.1.3. Analyzing the input

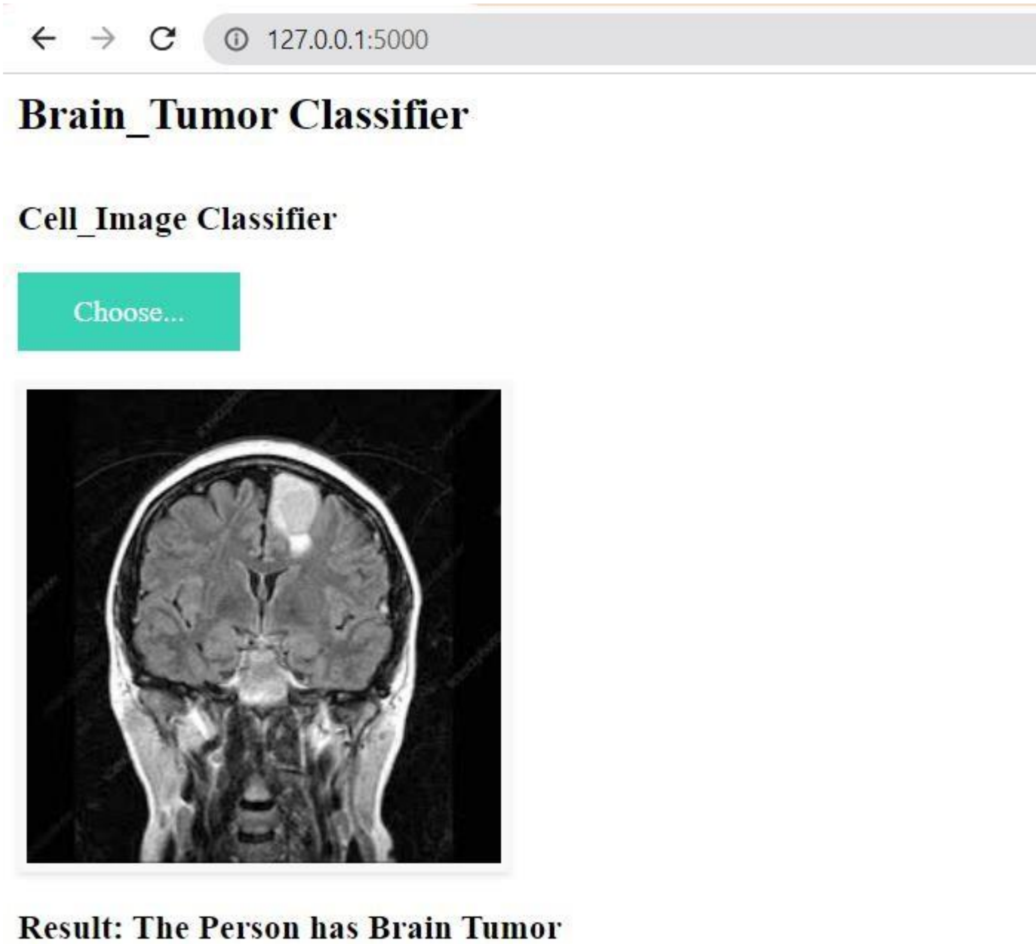


Fig.5.1.4.Output Produced

CHAPTER – 6

TESTING DESIGN

The common view of testing held by users is that it is performed to prove that there are noerrors in a program. It is extremely difficult since designer cannot prove to be one hundred percentaccurate. Therefore, the most useful and practical approach is with the understanding that testing is the process of executing a program with explicit intention of finding errors that make the program fail.

Testing has its own cycle. The testing process begins with the product requirements phaseand from there parallels the entire development process. In other words, for each phase of the development process there is an important testing activity. Successful testing requires a methodicalapproach. It requires focusing on basic critical factors:

- Planning
- App and process control
- Risk management
- Inspections
- Measurement tools
- Organization and professionalism

6.1 Test Plan:

Before going for testing, first we have to decide upon the type of testing to be carried out.

The following factors are taken into consideration:

- To ensure that information properly flows into and out of program.
- To find out whether the local data structures maintain its integrity during all steps inan algorithm execution.
- To ensure that the module operate properly at boundaries established to limit or restrict processing.
- To find out whether error - handling paths are working correctly or not.
- To find out whether the values are correctly updated or not.
- Check for validations

6.1.1 Black Box Testing:

It is a software testing approach in the tester doesn't know the internal working of the item being tested. For example in a Black box test, on software design the tester only knows the input and the expected outputs. Tester doesn't know how the program derives the output. Tester doesn't even imagine as to how, the coding is done. Tester need to know only the specifications.

The advantages of black box testing approach are

- The test is unbiased because the designer and the tester is independent of each other
- The tester needs no specific knowledge on any programming language
- The test is done from the point of view of the user, not the designer.
- The test can be designed as soon as the specifications are complete

The disadvantages of black box testing approach are

- The test can be redundant if the software designer has already run a test case.
- The test can be difficult to design
- Testing every possible input stream is unrealistic.

6.1.2 Unit Testing:

Unit or module testing is the process of testing the individual components (subprograms or procedures) of a program. The purpose is to discover discrepancies between the modules interface specification and its actual behavior. In our system each module must be tested independently for validation.

6.1.3 Integration Testing

Integration testing is the process of combining and testing multiple components together. The primary objective of integration testing is to discover errors in the interfaces between the components. In our system each of the modules mentioned above, are tested for checking the integration between them, after each of them are tested individually.

6.2 SYSTEM IMPLEMENTATION:

System implementation is the important stage of app when the theoretical design is tunes into practical system. The main stages in the implementation are as follows:

- Planning
- Training
- System testing and
- Changeover planning

Planning is the first task in the app implementation. Planning is deciding on the method and the time scale to be adapted. At the time of implementation of any system people from different departments and system analysis involve. They are confirmed to practical problem of controlling various activities of people outside their own data processing departments. The line manager controlled through an implementation co-ordinate committee. The committee consists of ideas, Problems and complaints of user department. It must also consider,

- The implementation of system environment.
- Self selection and allocation for implementation tasks.
- Consultation with unions and resources available.
- Standby facilities and channels of communication.

CHAPTER -7

CONCLUSION AND FUTURE WORK

7.1 CONCLUSION:

Detection of brain tumor using image processing is proposed. The proposed system is helpful in detection of brain tumors automatically. Here we applied Convolutional neural network (CNN) which is a machine learning method. This system identifies the abnormalities in the brain which is detected in the MR image. The system requires less training set and helps in faster detection of the tumors and provides 96.39% accurate results.

7.2 FUTURE ENHANCEMENT:

In future following enhancements can be made:

- The performance of the system should be evaluated to determine whether system achieves the results that are expected and whether the predicted benefits of the system are realized.
- Enhance the model by adding features that indicate the location of the brain tumor within the MRI image.
- Increase the accuracy of the classification algorithm to identify not only the presence of a brain tumor, but also the specific type and stage of the tumor.

APPENDIX

CODING

base.html

```
<html lang="en">
 <head>
   <meta charset="UTF-8" />
   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
   <meta http-equiv="X-UA-Compatible" content="ie=edge" />
   <title>Brain_Tumor Classifier</title>
   <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
   <script src="http://code.jquery.com/jquery-1.10.2.js"></script>
   <script src="http://code.jquery.com/ui/1.11.2/jquery-ui.js"></script>
   <script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>
   <link
     href="{{ url_for('static', filename='css/main.css') }}"
     rel="stylesheet"
   />
 </head>

 <body>
   <nav class="navbar navbar-dark bg-dark">
     <div class="container">
       <h1>Brain_Tumor Classifier</h1>
     </div>
   </nav>
   <div class="container">
     <div id="content" style="margin-top: 2em">
       {% block content %}{% endblock %}
     </div>
   </div>
 </body>
```

```
  <footer>
    <script
      src="{{ url_for('static', filename='js/main.js') }}"
      type="text/javascript"
    ></script>
  </footer>
</html>
```

index.html

```
{% extends "base.html" %} {% block content %}

<h3>Cell_Image Classifier</h3>

<div>
  <form id="upload-file" method="post" enctype="multipart/form-data">
    <label for="imageUpload" class="upload-label"> Choose... </label>
    <input
      type="file"
      name="file"
      id="imageUpload"
      accept=".png, .jpg, .jpeg"
    />
  </form>

  <div class="image-section" style="display: none">
    <div class="img-preview">
      <div id="imagePreview"></div>
    </div>
    <div>
      <button type="button" class="btn btn-primary btn-lg" id="btn-predict">
        Predict!
      </button>
    </div>
```

```
  </div>

  <div class="loader" style="display: none"></div>

  <h3 id="result">
    <span> </span>
  </h3>
</div>

{% endblock %}
```

main.css

```
.img-preview {
    width: 256px;
    height: 256px;
    position: relative;
    border: 5px solid #F8F8F8;
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);
    margin-top: 1em;
    margin-bottom: 1em;
}

.img-preview>div {
    width: 100%;
    height: 100%;
    background-size: 256px 256px;
    background-repeat: no-repeat;
    background-position: center;
}

input[type="file"] {
    display: none;
}
```

```
.upload-label{
    display: inline-block;
    padding: 12px 30px;
    background: #39D2B4;
    color: #fff;
    font-size: 1em;
    transition: all .4s;
    cursor: pointer;
}

.upload-label:hover{
    background: #34495E;
    color: #39D2B4;
}

.loader {
    border: 8px solid #f3f3f3; /* Light grey */
    border-top: 8px solid #3498db; /* Blue */
    border-radius: 50%;
    width: 50px;
    height: 50px;
    animation: spin 1s linear infinite;
}

@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}
```

main.js

```
$(document).ready(function () {
    // Init
    $('.image-section').hide();
    $('.loader').hide();
```

```
$('#result').hide();

//      Upload       Preview
function readURL(input) {
    if (input.files && input.files[0]) {
        var reader = new FileReader();
        reader.onload = function (e) {
            $('#imagePreview').css('background-image', 'url(' + e.target.result + ')');
            $('#imagePreview').hide();
            $('#imagePreview').fadeIn(650);
        }
        reader.readAsDataURL(input.files[0]);
    }
}
$("#imageUpload").change(function () {
    $('.image-section').show();
    $('#btn-predict').show();
    $('#result').text('');
    $('#result').hide();
    readURL(this);
});

// Predict
$('#btn-predict').click(function () {
    var form_data = new FormData($('#upload-file')[0]);

    // Show loading animation
    $(this).hide();
    $('.loader').show();

    // Make prediction by calling api /predict
    $.ajax({
        type: 'POST',
        url: '/predict',
        data: form_data,
```

```
            contentType: false,
            cache: false,
            processData: false,
            async: true,
            success: function (data) {
                // Get and display the result
                $('.loader').hide();
                $('#result').fadeIn(600);
                $('#result').text(' Result:  ' + data);
                console.log('Success!');
            },
        });
    });

});
```

app.py

```
import os
import numpy as np

# Keras
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

# Flask utils
from flask import Flask, request, render_template
from werkzeug.utils import secure_filename
import tensorflow_hub as hub

# Define a flask app
app = Flask(_name_)

# Model saved with Keras model.save()
model = load_model(('brain_tumor.h5'), custom_objects={
```

```
                'KerasLayer': hub.KerasLayer})

def model_predict(img_path, model):
    test_image = image.load_img(img_path, target_size=(256, 256))
    test_image = image.img_to_array(test_image)
    test_image = test_image / 255
    test_image = np.expand_dims(test_image, axis=0)
    result = model.predict(test_image)
    result

    if result <= 0.5:
        result = "The Person has no Brain Tumor"
    else:
        result = "The Person has Brain Tumor"

    return result

@app.route('/', methods=['GET'])
def index():
    # Main page
    return render_template('index.html')

@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == 'POST':
        # Get the file from post request
        f = request.files['file']

        # Save the file to ./uploads
        basepath = os.path.dirname(_file_)
        file_path = os.path.join(
            basepath, 'uploads', secure_filename(f.filename))
        f.save(file_path)

        # Make prediction
```

```
        preds = model_predict(file_path, model)
        result = preds
        return result
    return None
if___name___== '_main_':
    app.run(debug=False)
```

brain tumor predictor.ipynb

```
import tensorflow as tf
import os
import cv2
import imghdr
```

Removing Dodgy images

```
data_dir = 'testers'
image_exts = ['jpeg','png','jpg','bmp']
os.listdir(data_dir)

for image_class in os.listdir(data_dir):
    for image in os.listdir(os.path.join(data_dir,image_class)):
        image_path = os.path.join(data_dir,image_class,image)
        try:
            img = cv2.imread(image_path)
            tip = imghdr.what(image_path)
            if tip not in image_exts:
                print("Image not in expected format - {}".format(image_path))
                os.remove(image_path)
        except Exception as e:
            print("Issue in image - {}".format(image_path))
```

Loading Data

```
import numpy as numpy
```

```
from matplotlib import pyplot as plt

data = tf.keras.utils.image_dataset_from_directory(data_dir)

data_iterator = data.as_numpy_iterator()
batch = data_iterator.next()
batch[0].shape

tot_images = 5
fig,ax = plt.subplots(ncols=tot_images,figsize=(20,20))
for idx,img in enumerate(batch[0][:tot_images]):
    ax[idx].imshow(img.astype(int))
    ax[idx].title.set_text(batch[1][idx])
```

Data Pre-processing

Scale Data

```
data = data.map(lambda x, y: (x/255, y))

scaled_iterator = data.as_numpy_iterator()
batch = scaled_iterator.next()
batch
```

Split Data

```
train_size = int(len(data)*.7)
val_size = int(len(data)*.2)+1
test_size = int(len(data)*.1)+1
val_size+train_size+test_size

train = data.take(train_size)
val = data.skip(train_size).take(val_size)
test = data.skip(train_size+val_size).take(test_size)
```

Deep Learning

Build Model

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Dense, Flatten

model = Sequential()

model.add(Conv2D(16,(3,3),1,activation='relu',input_shape = (256,256,3)))
model.add(MaxPooling2D())

model.add(Conv2D(32,(3,3),1,activation='relu'))
model.add(MaxPooling2D())

model.add(Conv2D(16,(3,3),1,activation='relu'))
model.add(MaxPooling2D())

model.add(Flatten())

model.add(Dense(256,activation='relu'))
model.add(Dense(1,activation='sigmoid'))

model.compile('adam', loss=tf.losses.BinaryCrossentropy(), metrics=['accuracy'])

model.summary()
```

Train

```
logdir='logs'
tensorCallback = tf.keras.callbacks.TensorBoard(log_dir = logdir)

hist = model.fit(train,epochs=5,validation_data=val,callbacks=[tensorCallback])
```

Evaluate

```
from tensorflow.keras.metrics import Precision, Recall, BinaryAccuracy

pre = Precision()
re = Recall()
acc = BinaryAccuracy()

for batch in test.as_numpy_iterator():
    X, y = batch
    yhat = model.predict(X)
    pre.update_state(y, yhat)
    re.update_state(y, yhat)
    acc.update_state(y, yhat)

print(f'Precision = {pre.result().numpy()}, Recall = {re.result().numpy()}, Accuracy = {acc.result().numpy()}')
```

Save the Model

```
from tensorflow.keras.models import load_model
model.save('brain_tumor.h5')
```

REFERENCES

1. A.Sivaramakrishnan And Dr.M.Karnan "A Novel Based Approach For Extraction Of Brain Tumor In Mri Images Using Soft Computing Techniques," International Journal Of Advanced Research In Computer And Communication Engineering, Vol. 2, Issue 4, April 2013.
2. Asra Aslam, Ekram Khan, M.M. Sufyan Beg, Improved Edge Detection Algorithm for Brain Tumor Segmentation, Procedia Computer Science, Volume 58,2015, Pp 430-437, ISSN 1877-0509.
3. B.Sathya and R.Manavalan, Image Segmentation by Clustering Methods: Performance Analysis, International Journal of Computer Applications (0975 – 8887) Volume 29– No.11, September 2011.
4. Devkota, B. & Alsadoon, Abeer & Prasad, P.W.C. & Singh, A.K. & Elchouemi, A.. (2018). Image Segmentation for Early Stage Brain Tumor Detection using Mathematical Morphological Reconstruction. Procedia Computer Science. 125. 115-123. 10.1016/j.procs.2017.12.017.
5. K. Sudharani, T. C. Sarma and K. Satya Rasad, "Intelligent Brain Tumor lesion classification and identification from MRI images using k-NN technique," 2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kumaracoil, 2015, pp. 777-780. DOI: 10.1109/ICCICCT.2015.7475384
6. Kaur, Jaskirat & Agrawal, Sunil & Renu, Vig. (2012). A Comparative Analysis of Thresholding and Edge Detection Segmentation Techniques. International Journal of Computer Applications.vol. 39.pp. 29-34. 10.5120/4898-7432.
7. Li, Shutao, JT-Y. Kwok, IW-H. Tsang and Yaonan Wang. "Fusing images with different focuses using support vector machines." IEEE Transactions on neural networks 15, no. 6 (2004): 1555-1561.
8. M. Kumar and K. K. Mehta, "A Texture based Tumor detection and automatic Segmentation using Seeded Region Growing Method," International Journal of Computer Technology and Applications, ISSN: 2229-6093, Vol. 2, Issue 4, PP. 855-859 August 2011.
9. Mahmoud, Dalia & Mohamed, Eltaher. (2012). Brain Tumor Detection Using Artificial Neural Networks. Journal of Science and Technology. 13. 31-39.
10. Marroquin J.L., Vemuri B.C., Botello S., Calderon F. (2002) An Accurate and Efficient Bayesian Method for Automatic Segmentation of Brain MRI. In: Heyden A., Sparr G., Nielsen M., Johansen P. (eds) Computer Vision — ECCV 2002. ECCV 2002. Lecture Notes in Computer Science, vol 2353. Springer, Berlin, Heidelberg.