AUTOMATED DETECTION OF BRAIN TUMOR USING DEEP LEARNING TECHNIQUE AND ASSESSING THE PERFORMANCE USING DIFFERENT OPTIMIZERS

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

MASTER OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

*By*

[NAME]

[REGD. NO]

UNDER THE ESTEEMED GUIDANCE OF

Dr. B S PANDA

M.Tech, PhD

Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

**RAGHU ENGINEERING COLLEGE**

**(Autonomous) Affiliated to JNTU-GV VIZIANAGARAM**

**Approved by AICTE, accredited by NBA, Accredited by NAAC with A+ grade**

[**www.raghuenggcollege.com**](http://www.raghuenggcollege.com/)

**2024**

**RAGHU ENGINEERING COLLEGE**

##### (Autonomous) Affiliated to JNTUGV VIZIANAGARAM

**Approved by AICTE, accredited by NBA, Accredited by NAAC with A+ grade**

##### [www.raghuenggcollege.com](http://www.raghuenggcollege.com/) 2024



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**CERTIFICATE**

This is to certify that this project entitled **“Automated detection of Brain Tumor using Deep learning technique and assessing the performance using different Optimizers”** done by

**[NAME],** bearing Regd.No: **[ ]** are student of M.Tech in the Department of Computer Science and Engineering, Raghu Engineering College, during the period 2022-2024 ,in partial fulfillment for the award of the Degree of Master of Technology in Computer Science & Engineering to the Jawaharlal Nehru Technological University, Vizianagaram is a record of bonafide work carried out under my guidance and supervision.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree.

|  |  |  |
| --- | --- | --- |
| **Internal Guide** |  | **Head of the Department** |
| **Dr. B S PANDA** |  | **Mr. S. SRINADH RAJU,** |
| Dept of CSE, |  | Dept of CSE, |
| Raghu Engineering college, |  | Raghu Engineering College, |
| Dakamarri (V), |  | Dakamarri (V), |
| Visakhapatnam. | **External Examiner** | Visakhapatnam. |

#### DISSERTATION APPROVAL SHEET

*This is to certify that the dissertation titled*

Automated detection of Brain Tumor using Deep learning technique and assessing the performance using different Optimizers

*By*

[name]

[regd no]

*Is approved for the degree of* ***Master of Technology***

##### Dr. B S Panda, (Guide)

**Internal Examiner**

##### External Examiner

**HOD**

Date.

### DECLARATION

This is to certify that this project titled **“Automated detection of Brain Tumor using Deep learning technique and assessing the performance using different Optimizers”** is bonafide work done by my me, in partial fulfillment of the requirements for the award of the degree M.Tech. and submitted to the Department of Computer Science & Engineering, **Raghu Engineering College, Dakamarri**.

I also declare that this project is a result of my team effort and that has not been copied from anyone and I have taken only citations from the sources which are mentioned in the references.

This work was not submitted earlier at any other University or Institute for the award of any degree.

##### Date:

**Place:**

[name]

[regd no]

##### ACKNOWLEDGEMENT

I express our sincere gratitude to our esteemed Institute “Raghu Engineering College”, which has provided us an opportunity to fulfill the most cherished desire to reach our goal.

I take this opportunity with great pleasure to put on record our ineffable personal indebtedness to **Mr Raghu Kalidindi, Chairman of Raghu Engineering College** for providing necessary departmental facilities.

I would like to thank the Principal **Dr. Ch. Srinivasu Rao**, Administration and Management of “**Raghu Engineering College**”, for providing the requisite facilities to carry out project in Campus.

My sincere thanks to Associate Professor **Sri. S. Srinadh Raju, Head of the Department,** Department of Computer Science and Engineering, Raghu Engineering College, for his kind support in the successful completion of this work.

I sincerely express our deep sense of gratitude to **Dr. B S Panda**, Professor, Department of Computer Science and Engineering, Raghu Engineering college, for his perspicacity, wisdom and sagacity coupled with compassion and patience. It is our great pleasure to submit this work under his wing.

I extend my deep hearted thanks to all the faculty members of Computer Science department for their value-based imparting of theory and practical subjects, which were use in the project.

I thankful to the non-teaching staff of the Department of Computer Science and Engineering, Raghu Engineering College, for their inexpressible support.

##### Regards

[name. Regd no]

### ABSTRACT

A brain tumor is the abnormal cell development in the brain, some of which can progress to cancer. Magnetic Resonance Imaging (MRI) scans are commonly used to diagnose brain tumors. From The MRI Images, information about the abnormal tissue growth in the brain is found. One of the most challenging issues in tumor segmentation is variation in brain tumor size, shape, and form. The tumor area is retrieved from all volume slices using the segmentation approach. Automated defect identification in medical imaging has emerged as an emerging topic in numerous medical diagnostic applications, one of which is determining whether or not a brain tumor exists. In this research, we propose a Deep learning algorithm such as convolution neural networks (CNN) to detect a tumor in the brain with different optimizers. The results of applying the suggested approach to MRI scans from the Kaggle dataset determine whether or not a brain tumor is present. The accuracy of 98.8% using the Adam optimizer was observed best among the three optimizers considered. We created a GUI based user interface in which we can upload MRI scans and get the results whether there is a tumor or not.

**Keywords:** Brain tumor, Medical Imaging, Deep Learning, Convolutional Neural Network, Optimizers

## TABLE OF CONTENTS

|  |  |
| --- | --- |
| **CONTENT** | **PAGE**  **NUMBER** |
| Certificate | i |
| Dissertation Approval Sheet | ii |
| Declaration | iii |
| Acknowledgement | iv |
| Abstract | v |
| Table of Contents | vi |
| List of figures | viii |
| **CHAPTER 1: INTRODUCTION** |  |
| 1.1 Purpose | 2 |
| 1.2 Scope | 2 |
| 1.3 Motivation | 3 |
| 1.4 Machine learning overview | 3 |
| 1.4.1 Deep learning | 6 |
| 1.4.2 CNN | 7-8 |
| 1.5 Proposed System | 9 |
| **CHAPTER 2: LITERATURE SURVEY** |  |
| 2.1 Introduction to literature survey | 11 |
| 2.2 Literature survey | 11-12 |
| **CHAPTER 3**: **SYSTEM ANALYSIS** |  |
| 3.1 Introduction | 14-15 |
| 3.2 Problem statement | 15 |
| 3.3 Existing System | 15 |
| 3.4 Modules Description | 15-16 |
| **CHAPTER 4: SYSTEM REQUIREMENTS** |  |
| 4.1 Software requirements | 18 |
| 4.2 Hardware requirements | 18 |

|  |  |
| --- | --- |
| 4.3 Project Perquisites | 18-20 |
| **CHAPTER 5 : SYSTEM DESIGN** |  |
| 5.1 Introduction | 32-23 |
| 5.2 System Model | 23-24 |
| 5.3 System Architecture | 24-26 |
| 5.4 UML Diagrams | 27-29 |
| **CHAPTER 6** : **IMPLEMENTATION** |  |
| 6.1 Technology Description | 31-34 |
| 6.2 Sample Code | 34-49 |
| **CHAPTER 7 : SCREEN SHOTS** |  |
| 7.1 Output screenshots | 51-53 |
| 7.2 Model Training Output | 54-66 |
| **CHAPTER 8 : SYSTEM TESTING** |  |
| 8.1 Introduction to Testing | 68 |
| 8.2 Types of Testing | 68-72 |
| **CHAPTER 9**: **CONCLUSION AND FUTURE**  **ENHANCEMENTS** |  |
| 9.1 Conclusion | 74 |
| 9.2 Future Scope | 74 |
| **CHAPTER 10: REFERENCES** | 76-77 |
| **Paper Publication** | 78 |
| **Paper** | 79-85 |

**List Of Figures**

|  |  |
| --- | --- |
| **Figure** | **Page Number** |
| Figure - 1.1 AI vs Machine Learning vs Neural Network vs Deep Learning vs Convolution Neural Network | 6 |
| Figure - 1.2 Convolutional neural network | 8 |
| Figure - 5.1 Workflow diagram | 26 |
| Figure - 5.2 Activity diagram | 27 |
| Figure - 5.3 Use case diagram | 28 |
| Figure - 5.4 Sequence diagram | 29 |
| Figure - 7.1 User Interface | 51 |
| Figure - 7.2 MRI Scan with brain tumor uploaded | 51 |
| Figure - 7.3 MRI Scan with brain tumor result | 52 |
| Figure - 7.4 MRI Scan with no brain tumor uploaded | 52 |
| Figure - 7.5 MRI Scan with no brain tumor result | 53 |
| Figure - 7.6 Adam optimizer model Epoch Result | 54 |
| Figure - 7.7Adam optimizer model Epoch Result | 55 |
| Figure - 7.8 Adam optimizer model Epoch Result | 55 |
| Figure - 7.9 Adam optimizer model summary | 56 |
| Figure - 7.10 Adam optimizer Confusion matrix | 57 |

**CHAPTER-1**

# INTRODUCTION

### Introduction:

In medical science, an anomalous and uncontrollable cell growth inside the brain is recognised as a tumor. A tumor in the brain can affect such sensory information and muscle movements or even results in a more dangerous situation which includes death. Depending upon the place of commencing, tumors can be categorised into primary tumors and secondary tumors. If the tumor originated inside the skull, then the tumor is known as primary brain tumor; otherwise if the tumor‘s initiation place is somewhere else in the body and moved towards the brain, then such tumors are called secondary tumors. Magnetic Resonance Imaging (MRI) has become the standard non-invasive technique for brain tumor diagnosis over the last few decades, due to its improved soft tissue contrast that does not use harmful radiations unlike other methods like CT(Computed Tomography), X-ray, PET (Positron Emission Tomography) scans etc. The MRI image is basically a matrix of pixels having characteristic features.

### Purpose

Manual detection of a brain tumor is a time-consuming and tedious process for health care professionals. So, the modern computer algorithms comprising Machine Learning and deep learning are used for the present work. When these techniques are applied on the MRI images the prediction of brain tumor is done fast and with higher accuracy helps in providing the treatment to the patients.

### Scope

The objective of brain tumor detection is to detect if tumor is present or not when MRI image is uploaded. Thus, our work aims to a develop technique that can accurately detect brain tumor

### Motivation

The main motivation behind Brain tumor detection is to detect tumor in less time with high accuracy. So, it can be useful in cases such as we have to sure the tumor is present or not, it can detect tumor from image and return the result tumor is present or not. This project deals with such a system, which uses computer, based procedures to detect tumor using Convolution Neural Network Algorithm for MRI images of different patients.

### Machine Learning Overview

Tom Mitchell states machine learning as “A computer program is said to learn from experience and from some tasks and some performance on, as measured by, improves with experience”. Machine Learning is combination of correlations and relationships, most machine learning algorithms in existence are concerned with finding and/or exploiting relationship between datasets. Once Machine Learning Algorithms can pinpoint on certain correlations, the model can either use these relationships to predict future observations or generalize the data to reveal interesting patterns. In Machine Learning there are various types of algorithms such as Regression, Linear Regression, Logistic Regression, Naive Bayes Classifier, Bayes theorem, KNN (K-Nearest Neighbor Classifier), Decision Tress, Entropy, ID3, SVM (Support Vector Machines), K-means Algorithm, Random Forest and etc. The name machine learning was coined in 1959 by Arthur Samuel. Machine learning explores the study and construction of algorithms that can learn from and make predictions on data Machine learning is closely related to (and often overlaps with) computational statistics, which also focuses on prediction- making through the use of computers. It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field. Machine learning is sometimes conflated with data mining, where the latter sub- field focuses more on exploratory data analysis and is known as unsupervised learning.

Within the field of data analytics, machine learning is a method used to devise complex models and algorithms that lend themselves to prediction; in commercial use, this is known as predictive analytics. These analytical models allow

researchers, data scientists, engineers, and analysts to "produce reliable, repeatable decisions and results" and uncover "hidden insights" through learning from historical relationships and trends in the data.

Machine learning tasks are typically classified into several broad categories:

##### Supervised learning:

The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs. As special cases, the input signal can be only partially available, or restricted to special feedback.

##### Semi-supervised Learning:

The computer is given only an incomplete training signal: a training set with some (often many) of the target outputs missing.

##### Active Learning:

The computer can only obtain training labels for a limited set of instances (based on a budget), and also has to optimize its choice of objects to acquire labels for. When used interactively, these can be presented to the user for labelling.

##### Unsupervised Learning:

No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

##### Reinforcement Learning:

Data (in form of rewards and punishments) are given only as feedback to the program's actions in a dynamic environment, such as driving a vehicle or playing a game against an opponent.

##### Deep Learning:

Deep Learning is known as deep structured learning or hierarchical learning. Most of the deep learning architectures and algorithms are built with the Artificial neural network (ANN) framework. Initially the input layer gets an input and it passes on a modified version of the input to the next layer. The layers between the input and output named as hidden layers and composed of multiple linear and non- linear transformation.

##### Open CV:

Open CV (Open Source Computer Vision Library) is a collection of algorithms for computer vision. It focus on real time image processing it is free for commercial and research use under a BSD license.

##### Tensorflow:

Tensor Flow is a mathematical computation library for training and building your machine learning and deep learning model with a simple touse high level APIs.

##### Keras:

Keras is a neural network API. It is library written specifically in python. In addition, it works with other libraries and packages such as tensorflow which makes deep learning easier. Keras was developed to allow for quick experimentation.

### DEEP LEARNING

Deep learning (also known as deep structured learning) is part of a broader family of [machine learning](https://en.wikipedia.org/wiki/Machine_learning) methods based on [artificial neural](https://en.wikipedia.org/wiki/Artificial_neural_network) [networks](https://en.wikipedia.org/wiki/Artificial_neural_network) with [representation learning.](https://en.wikipedia.org/wiki/Representation_learning) Learning can be [supervised](https://en.wikipedia.org/wiki/Supervised_learning), [semi](https://en.wikipedia.org/wiki/Semi-supervised_learning) [supervised](https://en.wikipedia.org/wiki/Semi-supervised_learning) or [unsupervised.](https://en.wikipedia.org/wiki/Unsupervised_learning)

Deep-learning architectures such as [deep neural networks](https://en.wikipedia.org/wiki/Deep_learning#Deep_neural_networks), [deep belief](https://en.wikipedia.org/wiki/Deep_belief_network) [networks,](https://en.wikipedia.org/wiki/Deep_belief_network) [deep reinforcement learning,](https://en.wikipedia.org/wiki/Deep_reinforcement_learning) [recurrent neural](https://en.wikipedia.org/wiki/Recurrent_neural_networks) [networks](https://en.wikipedia.org/wiki/Recurrent_neural_networks) and [convolutional neural networks](https://en.wikipedia.org/wiki/Convolutional_neural_networks) have been applied to fields including [computer vision](https://en.wikipedia.org/wiki/Computer_vision), [speech recognition](https://en.wikipedia.org/wiki/Speech_recognition), [natural language](https://en.wikipedia.org/wiki/Natural_language_processing) [processing,](https://en.wikipedia.org/wiki/Natural_language_processing) [machine translation](https://en.wikipedia.org/wiki/Machine_translation), [bioinformatics](https://en.wikipedia.org/wiki/Bioinformatics), [drug design,](https://en.wikipedia.org/wiki/Drug_design) [medical image](https://en.wikipedia.org/wiki/Medical_image_analysis) [analysis,](https://en.wikipedia.org/wiki/Medical_image_analysis) [climate science](https://en.wikipedia.org/wiki/Climatology), material inspection and [board game](https://en.wikipedia.org/wiki/Board_game) programs, where they have produced results comparable to and in some cases surpassing human expert performance.

Most modern deep learning models are based on [artificial neural networks](https://en.wikipedia.org/wiki/Artificial_neural_network), specifically [convolutional neural networks](https://en.wikipedia.org/wiki/Convolutional_neural_network) (CNN)s, although they can also include [propositional formulas](https://en.wikipedia.org/wiki/Propositional_formula) or latent variables organized layer-wise in deep [generative models](https://en.wikipedia.org/wiki/Generative_model) such as the nodes in [deep belief networks](https://en.wikipedia.org/wiki/Deep_belief_network) and deep [Boltzmann machines](https://en.wikipedia.org/wiki/Boltzmann_machine).

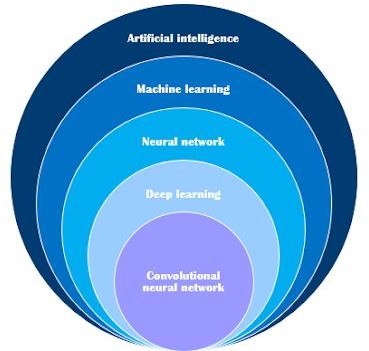


Figure 1.1 AI vs Machine Learning vs Neural Network vs Deep Learning vs Convolution Neural Network[1]

### CNN

Convolutional Neural Network are designed to process data through multiple layers of arrays. This type of neural networks is used in application like image recognition of face recognition. The primary difference between CNN and other ordinary neural network is that CNN takes input as a two-dimensional array and operates directly on the images rather than focusing on feature extraction which other neural network focus on.

The dominant approach of CNN includes solutions for problems of recognition. Top companies like google and Facebook have invested in research and developments towards recognition projects to get activities done with greater speed.

Convolutional neural network processes closely knitted data used for image classification, image processing, face detection etc. It is a specialized 3D structure with specialized NN analyzing RGB layers of an image. Unlike others, it analyses one image at a time, identifies and extracts important features and uses them to classify the image. Convolutional Neural Networks (ConvNets) automatically learns mid-level and high-level representations or abstractions from the input training data. The main building block used to construct a CNN architecture is the convolutional layer. It also consists of several other layers, some of which are described as below:

Input Layer: It takes in the raw pixel value of input image

Convolutional Layer: It is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel to generate a feature map Convolution of an image with different filters can perform operations such as edge detection, blur and sharpen by applying filters.

Activation Layer: It produces a single output based on the weighted sum of inputs. Pooling Layer-Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling (also called subsampling or down

sampling) reduces the dimensionality of each map but retains important information. Spatial pooling can be of different types:

Max Pooling – taking the largest element in the feature map of Average Pooling - taking the average of elements in the feature map Sum Pooling – taking the sum of all elements in the feature map

Fully Connected Layer: The layer we call the FC layer, we flattened our matrix into a vector and fed it into a fully connected layer like a neural network. The feature map matrix will be converted as a column vector (x1, x2, x3, …). With the fully connected layers, we combined these features together to create a model. For Classifying input images into various classes based on training set.

Dropout Layer: It prevents nodes in a network from co-adapting to each other.

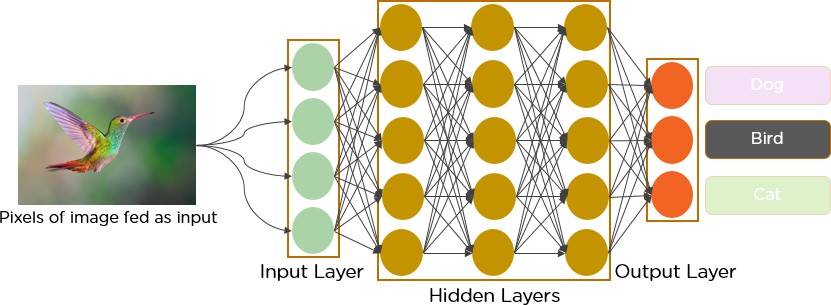


Figure 1.2 Convolutional neural network [2]

### Proposed System

In this project, we propose Machine learning and Deep learning to overcome the drawbacks of traditional classifiers where tumors are detected in brain MRI using machine learning and Deep learning. Machine learning and image classifiers can be used to efficiently detect cancer cells in the brain through MRI Neural Networks (NN) form the base of deep learning, a subfield of machine learning. Convolutional neural network processes closely knitted data used for image classification, image processing, face detection etc. It is a specialised 3D structure with specialised NN analysing RGB layers of an image. Unlike others, it analyses one image at a time, identifies and extracts important features and uses them to classify the image. Convolutional Neural Networks (ConvNets) automatically learns mid-level and high-level representations or abstractions from the input training data.

**CHAPTER-2**

# LITERATURE SURVEY

### Introduction to Literature Survey

In the present day, the field of Medical Imaging diagnosis using AI has gained a lot of importance, and it is the need of the hour to overwhelm the burden of health professionals. Most researchers have proposed various approaches to medical diagnosis earlier, and some of them are listed below.

### Literature Survey

* 1. Devkota et al. [3] suggested a computer-aided detection (CAD) method for detecting aberrant tissues using Morphological procedures. Among the several segmentation procedures available, morphological opening and closure operations are recommended because they need less processing time while extracting tumor regions with minor defects.

Astina Minz et al. [4] used the AdaBoost gadget mastering method to build an effective automated classification solution for brain images. The suggested system is divided into three parts. Pre-processing removed noise from the datasets and transformed photos to grayscale. The pre-processed picture includes median filtering and thresholding segmentation.

Mukambika et al. [5] suggested a technique for the later stage categorization of the tumor, whether it is existent or not. Their proposed work is a comparative evaluation of tumor detection methodologies using MR images, namely the Level set approach, discontinuous wavelength transformations (DWT), and K-method segmentation algorithms. Following that, feature extraction is performed, followed by SVM classification.

Shikha Gitte et al. [6] described various techniques to classify brain tumors, therapies improving patient's quality of life

Anjaly Antony et al. [7] present the implementation of a CNN for diagnosing brain tumors from MRI images. If the brain tumor is predicted, the position and the size of the tumor can be identified.

Ashwini patne et al. [8] present the process of detecting a tumor in the brain with image processing techniques.

Pranav Shetty et al. [9] present the segmentation technique of MRI based on CNN for spatial and structural changeability of brain tumors

Dr. K. Kranthi Kumar et al. [10] predict the magnitude of brain tumors using a Convolution Neural Network algorithm that provides us with reliable results.

Shreyash Gupta et al. [11] describe the importance of artificial intelligence in detecting tumor regions using computer vision and image segmentation.

Khalifa shantta et al. [12] reviews the state-of-the-art brain tumor detection techniques and highlights their pros and cons.

All papers are used to find tumor in the brain, but different techniques are used to find tumor. The methods include optimized k-means clustering with genetic algorithm. Traditional k-means algorithm is sensitive to the initial cluster centers. Another paper they required 2400 MRI images to detect the brain tumor.

**CHAPTER-3**

# SYSTEM ANALYSIS

### Introduction

A brain tumor is a mass or development of abnormal brain cells. There are several forms of brain tumors. Some brain tumors are noncancerous (benign), whereas others are cancerous (malignant). Brain tumors can start in your brain (primary brain tumors) or move from other body regions to your brain (metastatic brain tumors). Although the etiology of brain tumors is unknown, there are specific patterns among those who get them. It may impact any human being, whether a juvenile or an adult.

The tumor area must first be recognized to reduce the chance of death. Patients are physically assessed by utilizing Computed Tomography (CT scan) and Magnetic Resonance Imaging (MRI). Because MRI scans indicate the brain structure, size, and location, they will make it easier to diagnose the tumor and design the surgical technique for its removal. In medical imaging, machine learning-based automated defect identification has emerged as an active research area in numerous medical diagnostic applications. Its use in diagnosing brain tumors in MRI is critical because it offers information about aberrant tissues required for treatment planning.

According to current research, automatic computerized detection and diagnosis of disease based on medical image analysis might be a valuable strategy since it would reduce radiologist time while also obtaining tested accuracy. Furthermore, if computer algorithms can offer robust and reliable tumor portrayal measures, these automated measurements would considerably assist in the therapeutic care of brain tumors by relieving clinicians of the burden of manual tumor depiction. Machine Learning (ML) is a subset of AI that is built to think for itself, engage socially, learn new information from supplied data, and adapt and improve with experience.

Manual detection of a brain tumor is a time-consuming and tedious process for health care professionals. So, the modern computer algorithms comprising Artificial Intelligence are used for the present work. Artificial intelligence (AI) refers to modeling human intelligence processes by machines, particularly computer systems, to mimic human behavior. Deep Learning is a branch of machine

learning dealing with artificial neural networks that are inspired by the structure and function of the brain.

We employed the Convolutional Neural Network architecture for Brain Tumor Detection and Classification with tested accuracy in our study. Convolutional neural networks analyze densely connected data for image classification, image processing, face identification, and other applications. It is a specialized 3D structure with specialized NN that analyses an image's RGB layers. Also, we assess the performance of CNN with different optimizers such as Adam, Root Mean Square, and Stochastic Gradient Descent.

### Problem statement

Automated detection of Brain Tumor using Deep learning techniques and assessing the performance by testing different Optimizers and providing a user interface where the user can upload a brain MRI scan image and check whether the brain tumor is present or brain tumor is not present.

### Existing System

In existing work when an MRI scan image is taken as input then it validates and provides the output whether tumor is there or not.

### Modules Description

##### Image Pre processing

Image pre-processing [13]is the term for operations on images at the lowest level of abstraction. These operations do not increase image information content but they decrease it if entropy is an information measure. The aim of pre-processing is an improvement of the image data that suppresses undesired distortions or enhances some image features relevant for further processing and analysis task.

##### Feature Extraction

Feature Extraction is the mathematical statistical procedure that extracts the quantitative parameter of resolution changes/abnormalities that are not visible to the naked eye.

Feature Extraction is identifying abnormalities. We need to extract some features from images as we need to do classification of the images using a classifier which needs these features to get trained on.

##### Model Training and Testing

Train/Test is a method to measure the accuracy of your model. It is called Train/Test because you split the data set into two sets: a training set and a testing set. Train the model means create the model. Test the model means test the accuracy of the model.

##### Flask

Flask [14] is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

**CHAPTER-4**

# SYSTEM REQUREMENTS

### Software Requirements

* + - Language: Python 3
    - API: Keras
    - Data set: Kaggle
    - Python packages: OpenCv, numpy, Tensorflow-Keras, imutil, matplotlib.

### Hardware Requirements

* + - Processor: Intel Core
    - CPU: 2.30GHz
    - Installed memory (RAM):4.00GB
    - System Type: 64-bit Operating System

### Project Perquisites

* + - CV2 module in python
    - OS module in python
    - Tensorflow module in python
    - Numpy module in python
    - Pillow module in python
    - Keras module in python
    - flask module in python
    - matlab module in python
    - werkzeug module in python
    - Visual studio with Python Extention
    - python
    - html, css, bootstrap
    - javascript

##### Open CV

OpenCV-Python [15] makes use of **Numpy**, which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays. This also makes it easier to integrate with other libraries that use Numpy such as SciPy and Matplotlib.

##### OS Module

The OS [16] module in Python provides functions for interacting with the operating system. OS comes under Python’s standard utility modules. This module provides a portable way of using operating system-dependent functionality. The \*os\* and \*os.path\* modules include many functions to interact with the file system.

##### Tensorflow

TensorFlow [17] provides a collection of workflows to develop and train models using Python or JavaScript, and to easily deploy in the cloud, on-prem, in the browser, or on-device no matter what language you use. TensorFlow is an end-to- end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.

##### Numpy

NumPy is a Python library used for working with arrays.It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently.

##### Pillow

The [18] Python Imaging Library adds image processing capabilities to your Python interpreter. This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities. The core image

library is designed for fast access to data stored in a few basic pixel formats. It should provide a solid foundation for a general image processing tool.

**Keras**

Keras is an open-source high-level Neural Network library, which is written in Python is capable enough to run on Theano, TensorFlow, or CNTK. It was developed by one of the Google engineers, Francois Chollet. It is made user- friendly, extensible, and modular for facilitating faster experimentation with deep neural networks. It not only supports Convolutional Networks and Recurrent Networks individually but also their combination.

**Flask**

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

##### MATLAB

The MATLAB Engine API for Python provides a package for Python to call MATLAB as a computational engine. The engine supports the reference implementation (CPython).

##### Werkzeug

Werkzeug [19] doesn’t enforce any dependencies. It is up to the developer to choose a template engine, database adapter, and even how to handle requests. It can be used to build all sorts of end user applications such as blogs, wikis, or bulletin boards.

**CHAPTER-5**

# SYSTEM DESIGN

### 5.1. INTRODUCTION

A brain tumor is a cancerous or non-cancerous growth of abnormal cells in the brain, which leads to benign or malignant brain tumors. Most of the researchers are engaging in the primary type of tumor such as Gliomas. We have some ways to treat gliomas such as chemotherapy, radiotherapy, and surgery. Automation by computer-aided devices can be used to obtain the necessary clinical data such as tumor presence, location, and type. However, it is still a very challenging task in assessing their shape, volume, boundaries, tumor detection, size, segmentation, and classification. Also, brain tumor intensity varies from individual to individual. Magnetic Resonance Imaging (MRI) is preferred over other treatment and diagnosis methods because it gives superior image contrast in soft tissues and has non- invasive property. It is a difficult task to do annotation of brain tumors from MRI scans manually. Hence, there is a strong need for automation of brain tumor segmentation and classification with the help of computer vision and machine learning algorithms.

According to current research, automatic computerized detection and diagnosis of disease based on medical image analysis might be a valuable strategy since it would reduce radiologist time while also obtaining tested accuracy. Furthermore, if computer algorithms can offer robust and reliable tumor portrayal measures, these automated measurements would considerably assist in the therapeutic care of brain tumors by relieving clinicians of the burden of manual tumor depiction. Machine Learning (ML) is a subset of AI that is built to think for itself, engage socially, learn new information from supplied data, and adapt and improve with experience.

Manual detection of a brain tumor is a time-consuming and tedious process for health care professionals. So, the modern computer algorithms comprising Artificial Intelligence are used for the present work. Artificial intelligence (AI) refers to modeling human intelligence processes by machines, particularly computer systems, to mimic human behavior. Deep Learning is a branch of machine learning dealing with artificial neural networks that are inspired by the structure and function of the brain

Systems design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing and designing systems which satisfies the specific needs and requirements of a business or organization. Systems design implies a systematic approach to the design of a system. It may take a bottom-up or top-down approach, but either way the process is systematic wherein it takes into account all related variables of the system that needs to be created—from the architecture, to the required hardware and software, right down to the data and how it travels and transforms throughout its travel through the system. Systems design then overlaps with systems analysis, systems engineering and systems architecture.

### 5.2 SYSTEM MODEL:

**Deep Learning:**

Deep learning is an artificial intelligence (AI) function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network. It is extremely beneficial to data scientists who are tasked with collecting, analyzing and interpreting large amounts of data; deep learning makes this process faster and easier. In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning AI is able to learn without human supervision, drawing from data that is both unstructured and unlabeled.

### Open CV:

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection.

### TensorFlow:

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. Like similar platforms, it's designed to streamline the process of developing and executing advanced analytics applications for users such as data scientists, statisticians and predictive modelers.

### Keras:

Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. Keras is based on minimal structure that provides a clean and easy way to create deep learning models based on TensorFlow or Theano. Keras is designed to quickly define deep learning models. Well, Keras is an optimal choice for deep learning application.

### SYSTEM ARCHITECTURE

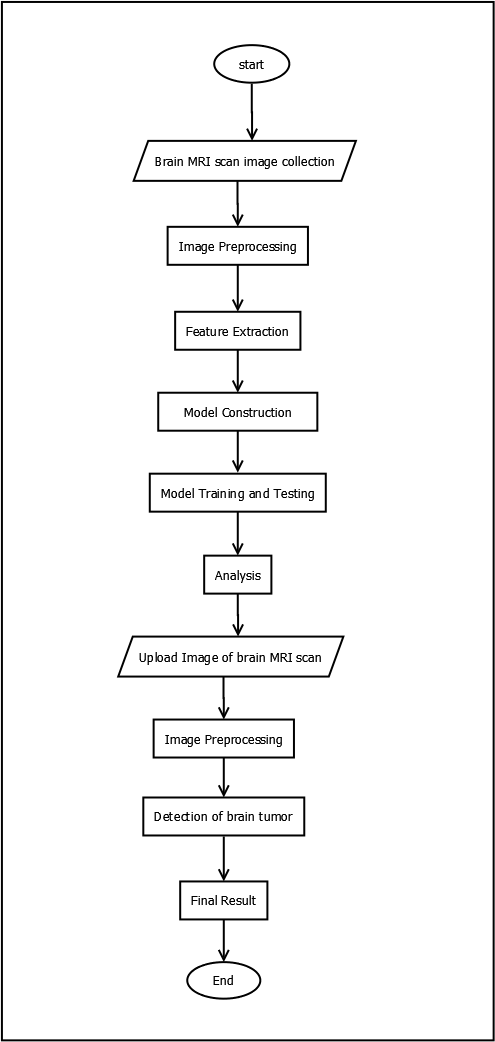
In our project we used Convolution Neural Networks to design a sequential model and used the Kaggle data set which contains images of MRI scans of brain. Which includes both the images of MRI scans which are effected and not effected by brain tumor.

The Dataset was colleted from Kaggle which consists of 1500 yes brain tumor images, 1500 no brain tumor images. we import OS package to read the images form the folder and CV2 package to convert image to array with RGB pixels and resize the images to 128 X 128 and covert the array into numpy array. If image label is 0 is for no brain tumor. If the image label is 1 is for yes brain tumor. We divided the dataset in which 80 percent of dataset for training and 20 percent of dataset for testing.

In out Project we constructed a sequential model with 4 con2d layers with different filters, with ‘relu’ activation function, 2 batch normalization layers, 2 maxpooling2d layers, 3 dropout layers, 2 dense layers and 1 flatten layer.

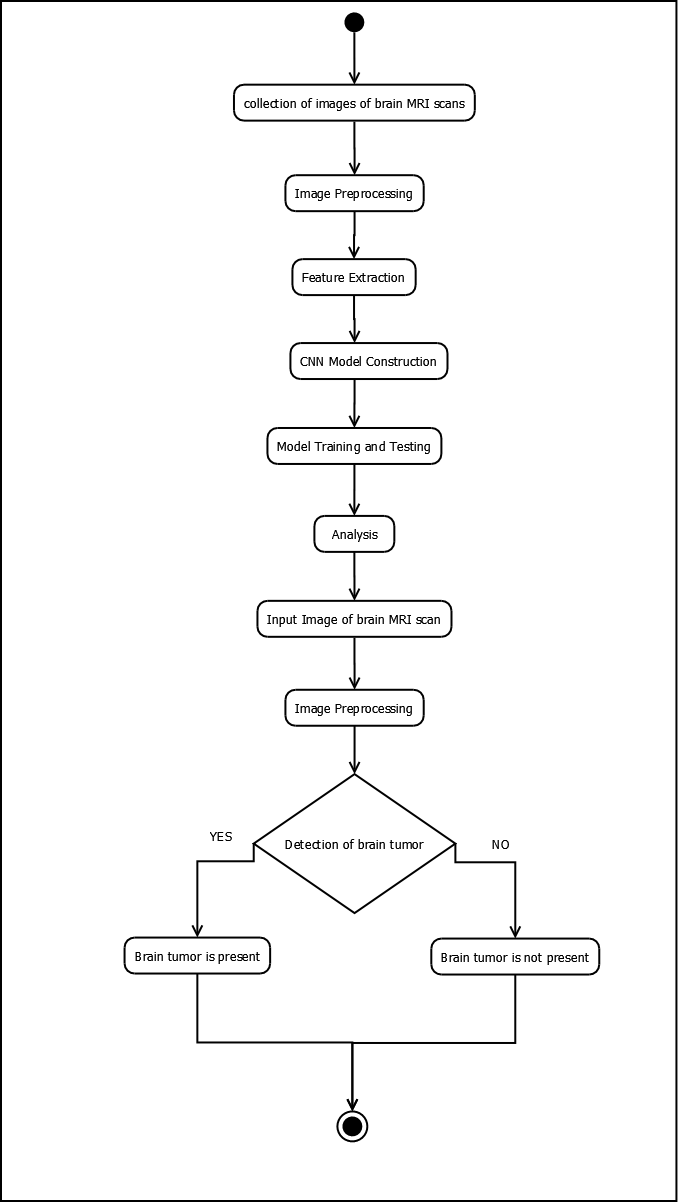
An optimizer is a strategy or algorithm that changes a neural network's weights and learning rate. As a result, it aids in the reduction of total loss and the improvement of accuracy. The main focus of our project is to assess the performance of the model with different optimizers and we used three different optimizers. They are Adam, RMSprop and SGD optimizer. We got the best results with Adam optimizer and the accuracy is 98.83%. We created a user interface to upload an image of MRI scan by using flask. Flask is a micro web framework written in python It is classified as a micro frame work because it does not require particular tools or libraries. However, Flask supports extensions that can add application features as if they were implemented in Flask itself.

In our project the User interface takes MRI scan image as input and when clicked on check results button the image undergoes the image pre-processing and gives the final result. The final result is if tumor is present, it gives the result as Brain Tumor is present else Brain Tumor is not present message will be displayed on the user interface.



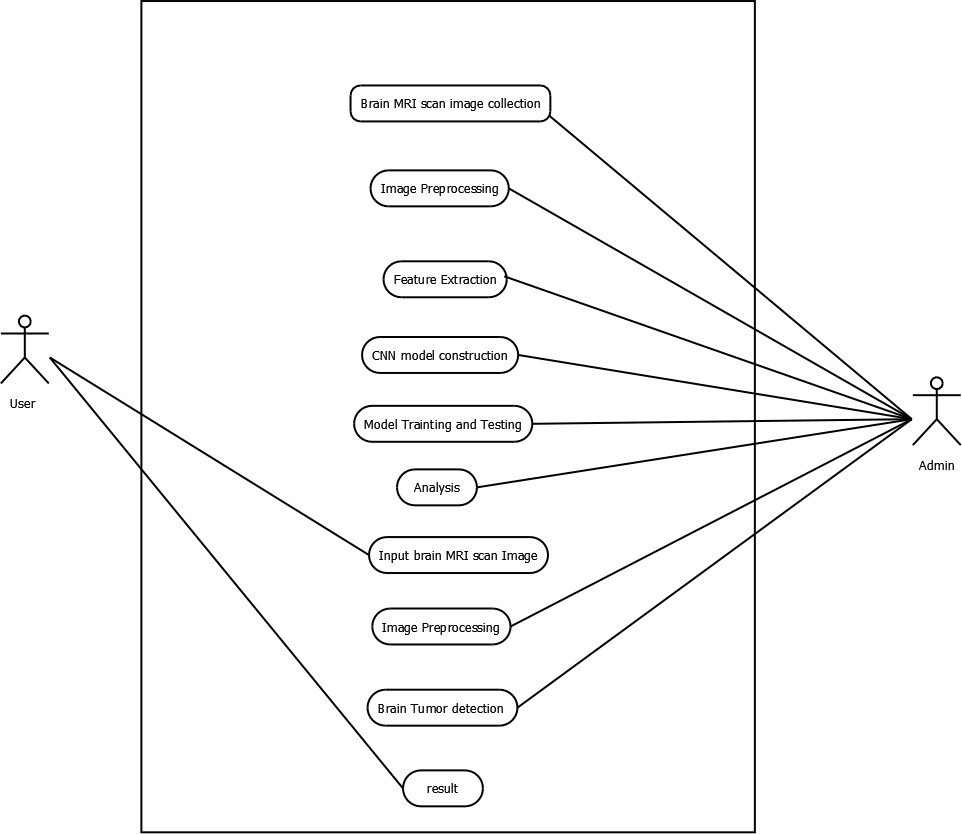
5.1. Work flow diagram

### UML DIAGRAMS Activity diagram



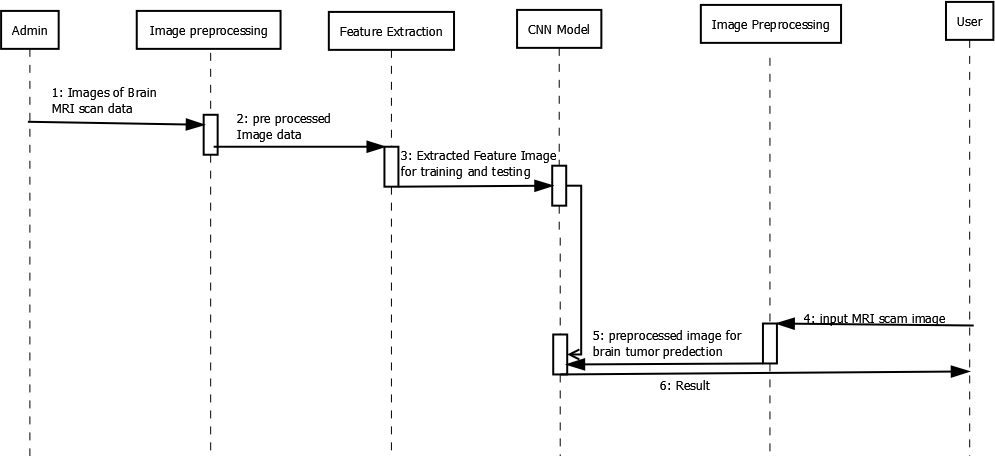
* 1. Activity diagram

### Use case diagram



* 1. Use case diagram

### Sequence diagram



5.4. sequence diagram

**CHAPTER-6**

# IMPLEMENTATIONS

## Technology description

### PYTHON

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by met programming and Meta objects. Many other paradigms are supported via extensions, including design by contract and logic programming. Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Python's developers strive to avoid premature optimization, and reject patches to non- critical parts of Python that would offer marginal increases in speed at the cost of clarity. When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or use PyPy, a just-in-time compiler. Python is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter. An important goal of Python's developers is keeping it fun to use. Python's design offers some support for functional programming in the Lisp tradition. It has filter, map, and reduce functions, list comprehensions, dictionaries, sets, and generator expressions. The standard library has two modules (iter tools and func tools) that implement functional tools borrowed from Haskell and Standard ML.

##### BENEFITS OF PYTHON

* + - Presence of Third-Party Modules
    - Extensive Support Libraries
    - Open Source and Community Development
    - Learning Ease and Support Available
    - User-friendly Data Structures
    - Productivity and Speed

### Model Training:

ML models can be trained to benefit manufacturing processes in several ways. The ability of ML models to process large volumes of data can help manufacturers identify anomalies and test correlations while searching for patterns across the data feed. It can equip manufacturers with predictive maintenance capabilities and minimize planned and unplanned downtime. We train the model using OpenCV, keras (python library).

### Optimizers:

An optimizer is a strategy or algorithm that changes a neural network's weights and learning rate. As a result, it aids in the reduction of total loss and the improvement of accuracy. We analyzed three different optimizers explained below.

The existing work is a forthright image processing approach towards detection and localization of brain tumor region The approach consists of a few stages such as pre-processing, edge detection and segmentation. The pre- processing stage converts the original image into a greyscale image, and noise removal if necessary. Next, the image is enhanced using image enhancement techniques. It is then followed by edge detection using Sobel and Canny algorithms.

##### 1 Adam Optimizer:

Adaptive moment estimate is the source of the word adam. To update network weights during training, this optimization approach further develops stochastic gradient descent (SGD). Unlike SGD, Adam optimizer modifies the learning rate for each network weight independently, rather than keeping a single learning rate entire training.

Adma Deep Learning Optimizer--

- (1)

The adam optimizer's operation is represented by equation (1) above. The decay rates of the average of the gradients are represented by B1 and B2.

##### 2 RMS Prop (Root Mean Square) Optimizer:

Among deep learning aficionados, the RMS prop is a popular optimizer. RMS prop is a natural extension of RPPROP's work. The problem of fluctuating gradients is solved by RPPROP. The issue with the gradients is that some were modest while others may be rather large. As a result, establishing a single learning rate may not be the ideal option. RPPROP adjusts the step size for each weight based on the gradient sign, and the RMS Prop optimizer is represented by equation (2) below.

RMS Prop(Root Mean Square) Deep Learning Optimizer

(2)

where gamma is the forgetting factor. Weights are updated by the below equation (3)

##### RMS Prop(Root Mean Square) Deep Learning Optimizer-----

**(3)**

##### 3 Stochastic Gradient Descent:

SGD is a version of Gradient Descent. It tries to update the parameters of the model more often. After each training example's loss has been computed, the model parameters are changed. As a result, if the dataset has 1000 rows, SGD will update

the model parameters 1000 times in one dataset cycle, rather than once as with Gradient Descent.

##### θ=θ−α⋅∇J(θ;x(i);y(i)) ----

**(4)**

##### where {x(i) ,y(i)} are the training examples.

Because model parameters are continuously changed, loss functions contain a lot of variability and fluctuation at different intensities.

### Sample Code

##### modelAdam.py :

import os import cv2

from PIL import Image import numpy

from sklearn.model\_selection import train\_test\_split from tensorflow import keras

from keras.models import Sequential

from keras.layers import Conv2D, MaxPooling2D, BatchNormalization, Dropout,

Flatten, Dense

import matplotlib.pyplot as plt

from sklearn.metrics import confusion\_matrix, ConfusionMatrixDisplay, classification\_report

tumor\_yes\_no\_images= [] yes\_no\_flag = []

c=0

for (root, directory, file) in os.walk(r'./brain\_tumor\_no\_yes\_images\_data'): c=c+1

if(c>1):

for f in file:

if '.jpg' in f:

path = os.path.join(root, f) img = cv2.imread(path)

img = Image.fromarray(img,'RGB') img = img.resize((128,128))

img = numpy.array(img) tumor\_yes\_no\_images.append(img) if(c==2):

yes\_no\_flag.append(0)

else:

yes\_no\_flag.append(1)

tumor\_yes\_no\_images = numpy.array(tumor\_yes\_no\_images) yes\_no\_flag = numpy.array(yes\_no\_flag)

x\_train, x\_test, y\_train, y\_test = train\_test\_split(tumor\_yes\_no\_images, yes\_no\_flag, test\_size=0.2, random\_state=0)

model = Sequential()

model.add(Conv2D(32, kernel\_size=(2, 2), input\_shape=(128, 128, 3)))

model.add(Conv2D(32, kernel\_size=(2, 2), kernel\_initializer='he\_uniform', activation ='relu', padding = 'Same'))

model.add(BatchNormalization()) model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Dropout(0.25))

model.add(Conv2D(64, kernel\_size = (2,2), kernel\_initializer='he\_uniform', activation ='relu', padding = 'Same'))

model.add(Conv2D(64, kernel\_size = (2,2), kernel\_initializer='he\_uniform', activation ='relu', padding = 'Same'))

model.add(BatchNormalization()) model.add(MaxPooling2D(pool\_size=(2,2), strides=(2,2))) model.add(Dropout(0.25))

model.add(Flatten()) model.add(Dense(128, activation='relu')) model.add(Dropout(0.5)) model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary\_crossentropy', optimizer='adam',metrics = ['accuracy'])

model.fit(x\_train, y\_train, batch\_size=32, verbose =1, epochs=50, validation\_data

= (x\_test,y\_test), shuffle = False)

y\_predict = (model.predict(x\_test)>0.5).astype("int32")

confusion\_matrix = confusion\_matrix(y\_test,y\_predict,labels=[0,1])

print("Confusion Matrix : ") print(confusion\_matrix) print()

confusion\_matrix\_display = ConfusionMatrixDisplay(confusion\_matrix)

confusion\_matrix\_display.plot()

plt.title("Confusion Matrix of Brain Tumor Detection") plt.show()

classification = classification\_report(y\_test,y\_predict,labels=[0,1]) print("Classification Report: ")

print(classification) print()

scores = model.evaluate(x\_test, y\_test, verbose=1) print("Accuracy: %.2f%%" % (scores[1]\*100)) print()

print(model.summary())

model.save('BrainTumorDetectionModel.h5')

##### modelRMSProp.py:

import os import cv2

from PIL import Image import numpy

from sklearn.model\_selection import train\_test\_split from tensorflow import keras

from keras.models import Sequential

from keras.layers import Conv2D, MaxPooling2D, BatchNormalization, Dropout,

Flatten, Dense

import matplotlib.pyplot as plt

from sklearn.metrics import confusion\_matrix, ConfusionMatrixDisplay, classification\_report

tumor\_yes\_no\_images= [] yes\_no\_flag = []

c=0

for (root, directory, file) in os.walk(r'./brain\_tumor\_no\_yes\_images\_data'): c=c+1

if(c>1):

for f in file:

if '.jpg' in f:

path = os.path.join(root, f) img = cv2.imread(path)

img = Image.fromarray(img,'RGB') img = img.resize((128,128))

img = numpy.array(img) tumor\_yes\_no\_images.append(img) if(c==2):

yes\_no\_flag.append(0) else:

yes\_no\_flag.append(1)

tumor\_yes\_no\_images = numpy.array(tumor\_yes\_no\_images) yes\_no\_flag = numpy.array(yes\_no\_flag)

x\_train, x\_test, y\_train, y\_test = train\_test\_split(tumor\_yes\_no\_images, yes\_no\_flag, test\_size=0.2, random\_state=0)

model = Sequential()

model.add(Conv2D(32, kernel\_size=(2, 2), input\_shape=(128, 128, 3)))

model.add(Conv2D(32, kernel\_size=(2, 2), kernel\_initializer='he\_uniform', activation ='relu', padding = 'Same'))

model.add(BatchNormalization()) model.add(MaxPooling2D(pool\_size=(2, 2))) model.add(Dropout(0.25))

model.add(Conv2D(64, kernel\_size = (2,2), kernel\_initializer='he\_uniform', activation ='relu', padding = 'Same'))

model.add(Conv2D(64, kernel\_size = (2,2), kernel\_initializer='he\_uniform', activation ='relu', padding = 'Same'))

model.add(BatchNormalization()) model.add(MaxPooling2D(pool\_size=(2,2), strides=(2,2))) model.add(Dropout(0.25))

model.add(Flatten()) model.add(Dense(128, activation='relu')) model.add(Dropout(0.5)) model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary\_crossentropy', optimizer='rmsprop',metrics = ['accuracy'])

model.fit(x\_train, y\_train, batch\_size=32, verbose =1, epochs=50, validation\_data

= (x\_test,y\_test), shuffle = False)

y\_predict = (model.predict(x\_test)>0.5).astype("int32")

confusion\_matrix = confusion\_matrix(y\_test,y\_predict,labels=[0,1])

print("Confusion Matrix : ") print(confusion\_matrix) print()

confusion\_matrix\_display = ConfusionMatrixDisplay(confusion\_matrix) confusion\_matrix\_display.plot()

plt.title("Confusion Matrix of Brain Tumor Detection") plt.show()

classification = classification\_report(y\_test,y\_predict,labels=[0,1])

print("Classification Report: ") print(classification)

print()

scores = model.evaluate(x\_test, y\_test, verbose=1) print("Accuracy: %.2f%%" % (scores[1]\*100)) print()

print(model.summary())

model.save('BrainTumorDetectionModelRMSProp.h5')

##### modelSGD.py:

import os import cv2

from PIL import Image import numpy

from sklearn.model\_selection import train\_test\_split

from tensorflow import keras

from keras.models import Sequential

from keras.layers import Conv2D, MaxPooling2D, BatchNormalization, Dropout,

Flatten, Dense

import matplotlib.pyplot as plt

from sklearn.metrics import confusion\_matrix, ConfusionMatrixDisplay, classification\_report

tumor\_yes\_no\_images= [] yes\_no\_flag = []

c=0

for (root, directory, file) in os.walk(r'./brain\_tumor\_no\_yes\_images\_data'): c=c+1

if(c>1):

for f in file:

if '.jpg' in f:

path = os.path.join(root, f) img = cv2.imread(path)

img = Image.fromarray(img,'RGB') img = img.resize((128,128))

img = numpy.array(img) tumor\_yes\_no\_images.append(img) if(c==2):

yes\_no\_flag.append(0) else:

yes\_no\_flag.append(1)

tumor\_yes\_no\_images = numpy.array(tumor\_yes\_no\_images) yes\_no\_flag = numpy.array(yes\_no\_flag)

x\_train, x\_test, y\_train, y\_test = train\_test\_split(tumor\_yes\_no\_images, yes\_no\_flag, test\_size=0.2, random\_state=0)

model = Sequential()

model.add(Conv2D(32, kernel\_size=(2, 2), input\_shape=(128, 128, 3)))

model.add(Conv2D(32, kernel\_size=(2, 2), kernel\_initializer='he\_uniform', activation ='relu', padding = 'Same'))

model.add(BatchNormalization()) model.add(MaxPooling2D(pool\_size=(2, 2))) model.add(Dropout(0.25))

model.add(Conv2D(64, kernel\_size = (2,2), kernel\_initializer='he\_uniform', activation ='relu', padding = 'Same'))

model.add(Conv2D(64, kernel\_size = (2,2), kernel\_initializer='he\_uniform', activation ='relu', padding = 'Same'))

model.add(BatchNormalization()) model.add(MaxPooling2D(pool\_size=(2,2), strides=(2,2))) model.add(Dropout(0.25))

model.add(Flatten()) model.add(Dense(128, activation='relu')) model.add(Dropout(0.5)) model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary\_crossentropy', optimizer='sgd',metrics = ['accuracy'])

model.fit(x\_train, y\_train, batch\_size=32, verbose =1, epochs=50, validation\_data

= (x\_test,y\_test), shuffle = False)

y\_predict = (model.predict(x\_test)>0.5).astype("int32")

confusion\_matrix = confusion\_matrix(y\_test,y\_predict,labels=[0,1])

print("Confusion Matrix : ") print(confusion\_matrix) print()

confusion\_matrix\_display = ConfusionMatrixDisplay(confusion\_matrix) confusion\_matrix\_display.plot()

plt.title("Confusion Matrix of Brain Tumor Detection") plt.show()

classification = classification\_report(y\_test,y\_predict,labels=[0,1])

print("Classification Report: ") print(classification)

print()

scores = model.evaluate(x\_test, y\_test, verbose=1) print("Accuracy: %.2f%%" % (scores[1]\*100)) print()

print(model.summary())

model.save('BrainTumorDetectionModelSgd.h5')

##### main.py:

from flask import Flask, render\_template, request from keras.models import load\_model

import os

from werkzeug.utils import secure\_filename import cv2

from PIL import Image import numpy

app = Flask( name )

model =load\_model('BrainTumorDetectionModel.h5')

@app.route('/', methods=['GET']) def webPage():

return render\_template('webPage.html')

@app.route('/predict', methods=['POST']) def upload():

if request.method == 'POST': f = request.files['file']

basepath = os.path.dirname( file )

file\_path = os.path.join(basepath, 'inputImagesData', secure\_filename(f.filename))

f.save(file\_path)

image = cv2.imread(file\_path)

image = Image.fromarray(image, 'RGB') image = image.resize((128, 128)) image=numpy.array(image)

input\_img = numpy.expand\_dims(image, axis=0) print(model.predict(input\_img),"result");

result = (model.predict(input\_img)>0.5).astype("int32") if result==0:

return "No, Brain Tumor is not Present (Detected with an Accuracy of 98.83%)."

elif result==1:

return "Yes, Brain Tumor is Present (Detected with an Accuracy of 98.83%)."

return None

if name == ' main ': app.run(debug=True)

##### webpage.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 Detection</title>

<link href="{{ url\_for('static', filename='styles/bootstrap.min.css') }}" rel="stylesheet">

<script src="{{ url\_for('static', filename='bootstrapJQuery/jquery.min.js')

}}"></script>

<script src="{{ url\_for('static', filename='bootstrapJQuery/bootstrap.min.js')

}}"></script>

<link href="{{ url\_for('static', filename='styles/test.css') }}" rel="stylesheet">

<script src="{{ url\_for('static', filename='bootstrapJQuery/main.js') }}" type="text/javascript"></script>

</head>

<body>

<nav class="navbar navbar-light bg-light">

<div class="container">

<a class="navbar-brand" href="#"><h2></h2></a>

<a class="navbar-brand" href="#"><h1>Brain Tumor Detection</h1></a>

<a class="navbar-brand" href="#"><h2></h2></a>

</div>

<button id="btn-home" href="#">Home</button>

</nav>

<center>

<br></br>

<div id="imgup"><h2> Please upload an Brain MRI Scan Image to Detect Brain Tumor:</h2></div>

<div id="imgupdone"><h2>Uploaded Brain MRI Scan Image to Detect Brain Tumor :</h2></div>

<form id="imageInputForm" method="post" enctype="multipart/form- data">

<input type="file" name="file" id="inputImage" accept=".png, .jpg,

.jpeg">

</form>

<div class="image-section" style="display:none;">

<img id="imageView" class="img-responsive" src="#" style="width:300px;height:300px;"/><br><br>

<div>

<button type="button" class="btn btn-success btn-lg " id="btn- CheckResult">check Results</button>

</div>

</div>

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

<h3 id="output">

<span> </span>

</h3>

</center>

</body>

</html>

##### main.js:

$(document).ready(function () { function readInputImage(image) {

if (image.files && image.files[0]) { var reader = new FileReader(); reader.onload = function (img) {

$('#imageView').attr( 'src', img.target.result );

}

displayImage = image.files[0] reader.readAsDataURL(displayImage);

}else {

$('.image-section').hide();

$('#imgupdone').hide();

$('.loader').hide();

$('#output').hide();

$('#imgup').show();

$('#upload-file')[0].reset();

}

}

$('#imgupdone').hide();

$('.image-section').hide();

$('.loader').hide();

$('#output').hide();

$("#inputImage").change(function () {

$('.image-section').show();

$('#imgup').hide();

$('#imgupdone').show();

$('#btn-CheckResult').show();

$('#output').text('');

$('#output').hide(); readInputImage(this);

});

$('#btn-home').click(function () {

$('.image-section').hide();

$('#imgupdone').hide();

$('#imgup').show();

$('.loader').hide();

$('#output').hide();

$('#imageInputForm')[0].reset();

});

$('#btn-CheckResult').click(function () {

var uploadedImage = new FormData($('#imageInputForm')[0]);

$(this).hide();

$('.loader').show();

$.ajax({

type: 'POST',

url: '/predict',

data: uploadedImage, processData: false, contentType: false, cache: false,

success: function (data) {

$('.loader').hide();

$('#output').fadeIn(600);

$('#output').text('Final Result :- ' + data);

},

});

});

});

**CHAPTER-7**

# OUTPUT SCREENS

### Output Screens

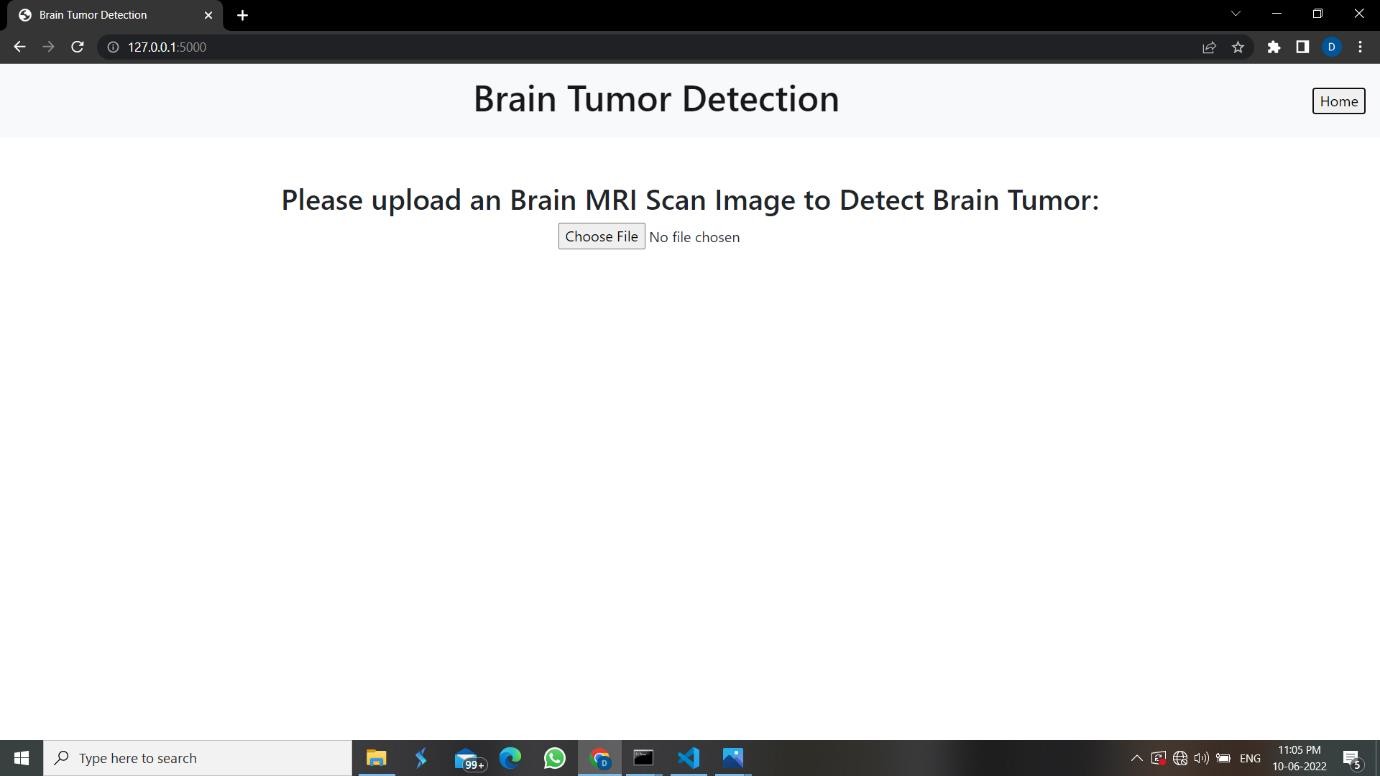


Figure 7.1 User Interface

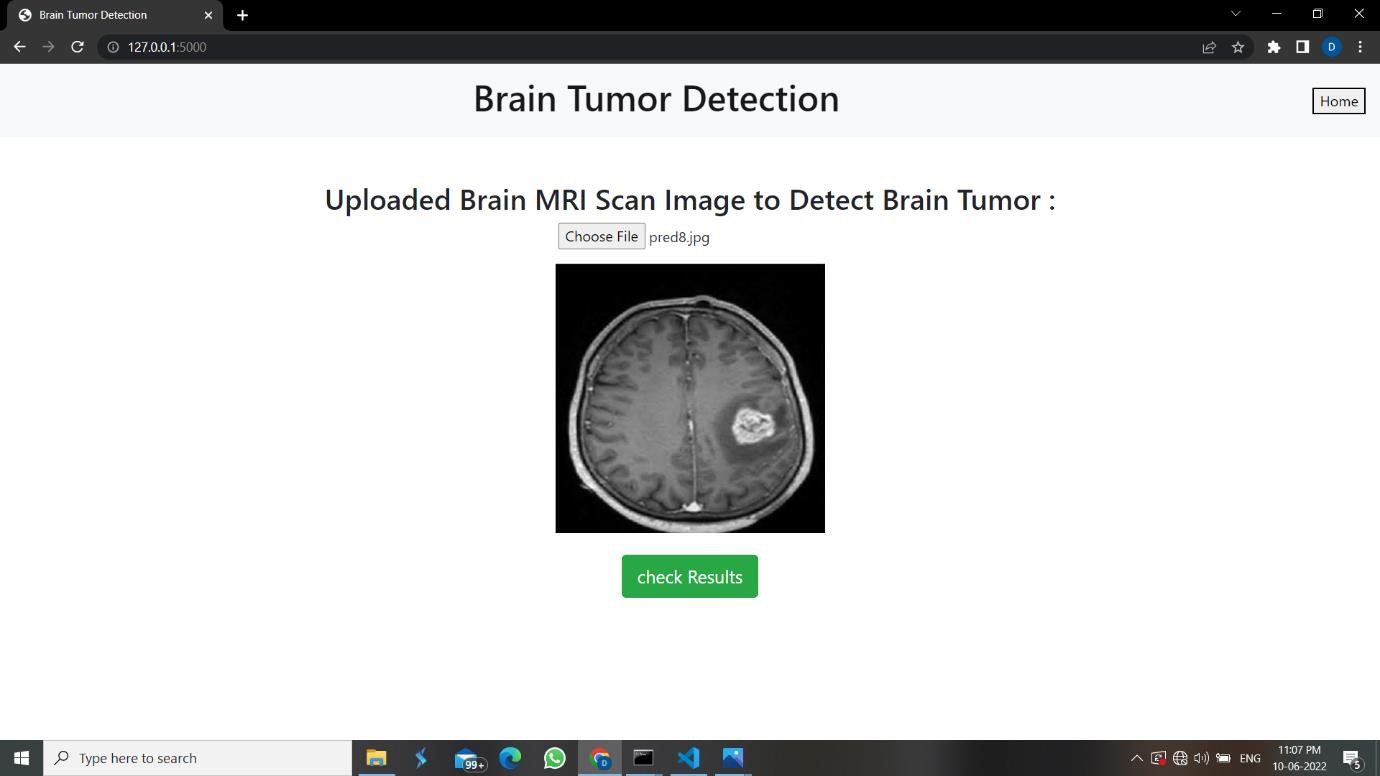


Figure 7.2 MRI Scan with brain tumor uploaded

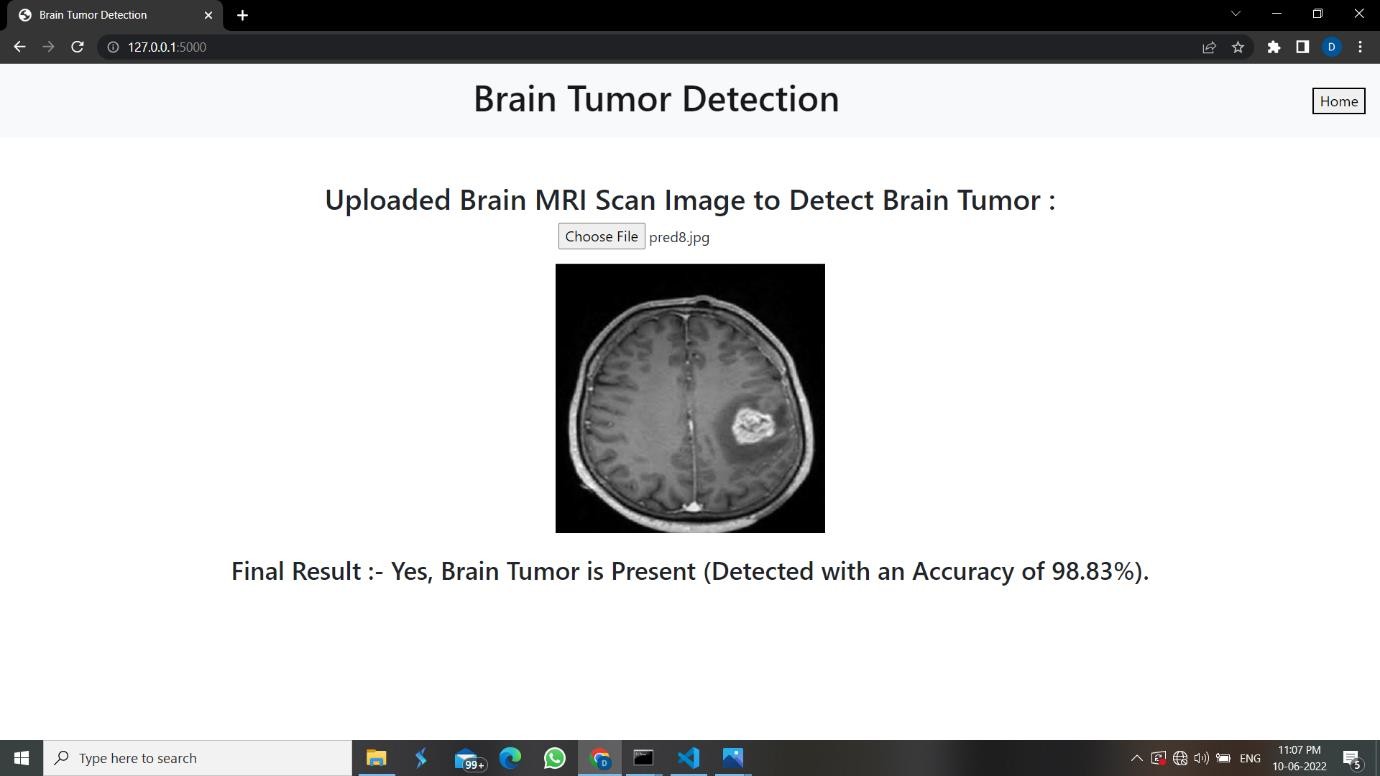


Figure 7.3 MRI Scan with brain tumor result

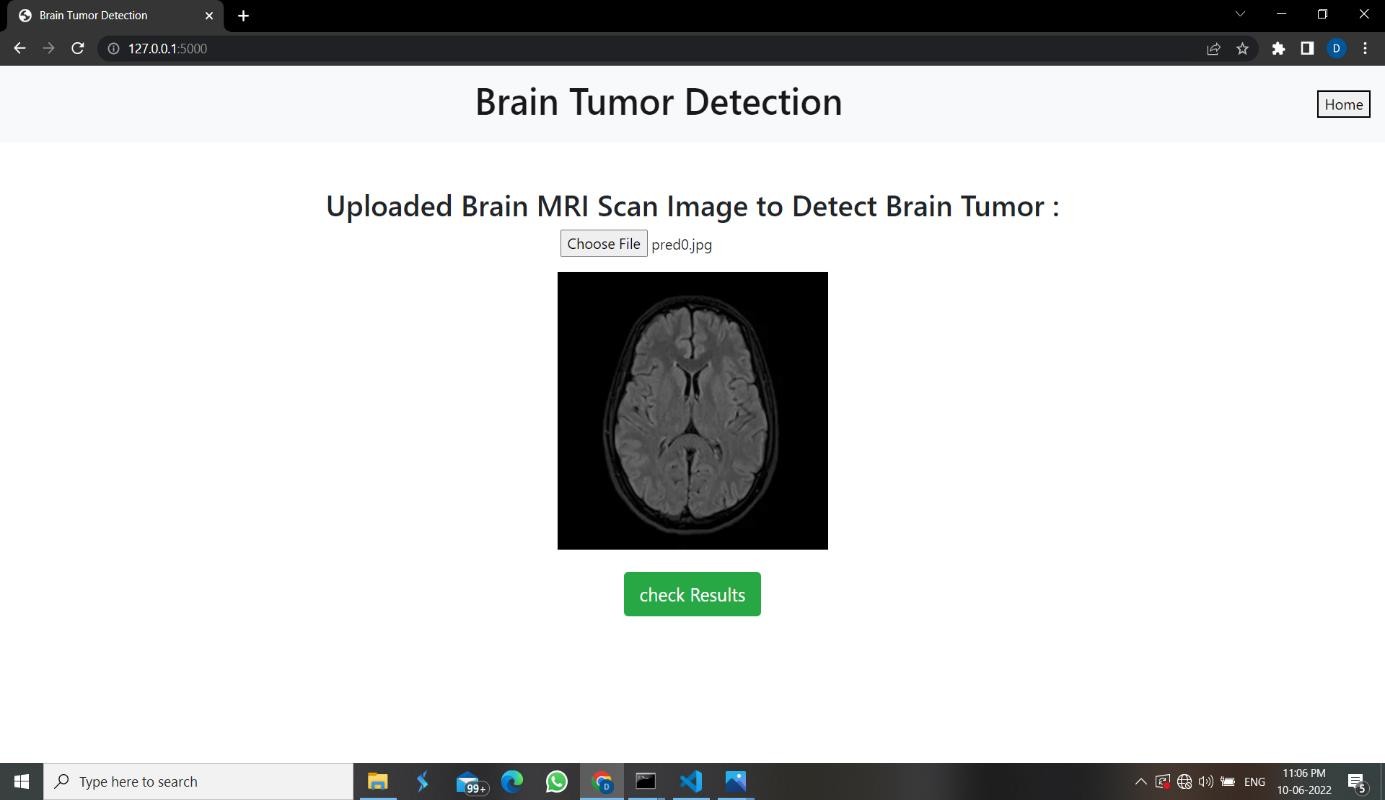


Figure 7.4 MRI Scan with no brain tumor uploaded

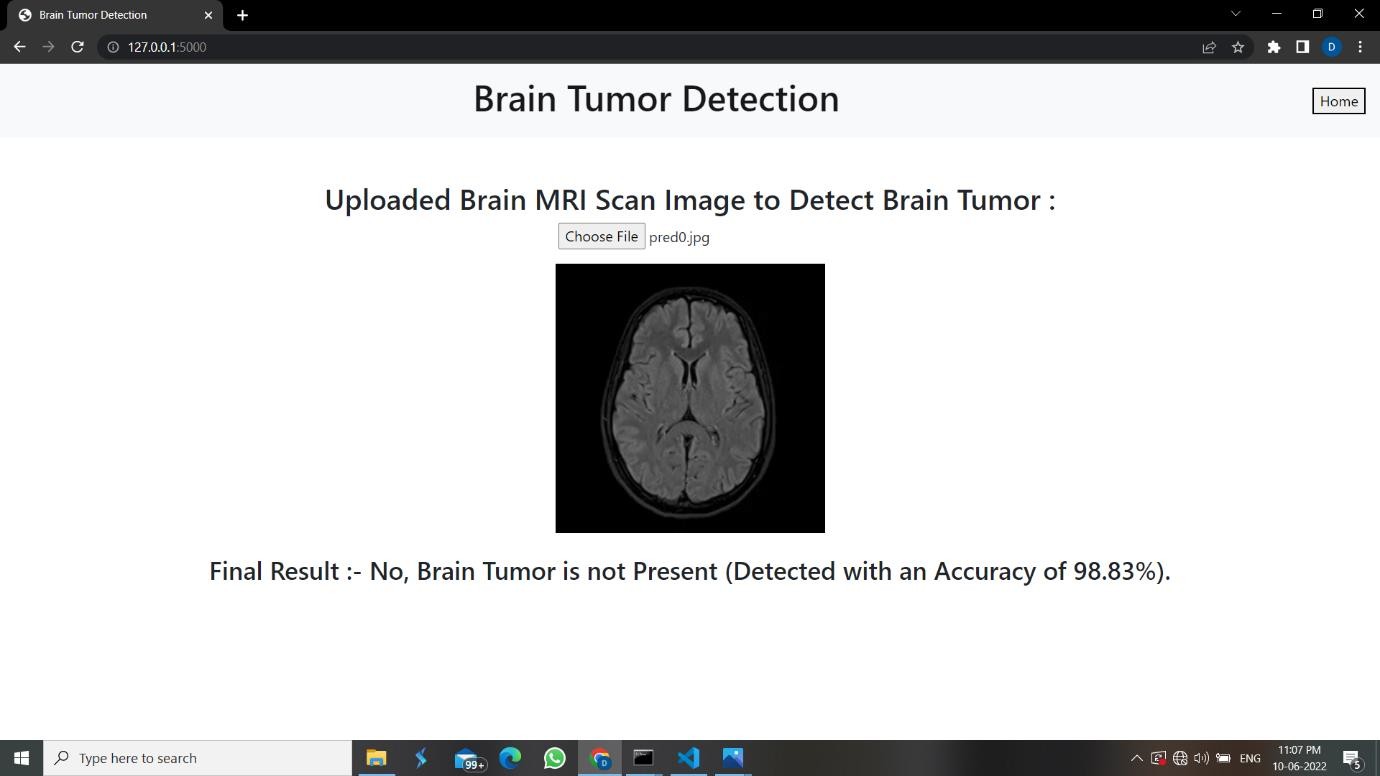


Figure 7.5 MRI Scan with no brain tumor result

### Model Training Output

We used three different optimizers in this project, i.e., ADAM, RMS Prop, and SGD. Among the three optimizers, ADAM produces the best accuracy results.

Model with Adam optimizer training result, confusion matrix, accuracy, summary classification reports:

#### Adam optimizer training results

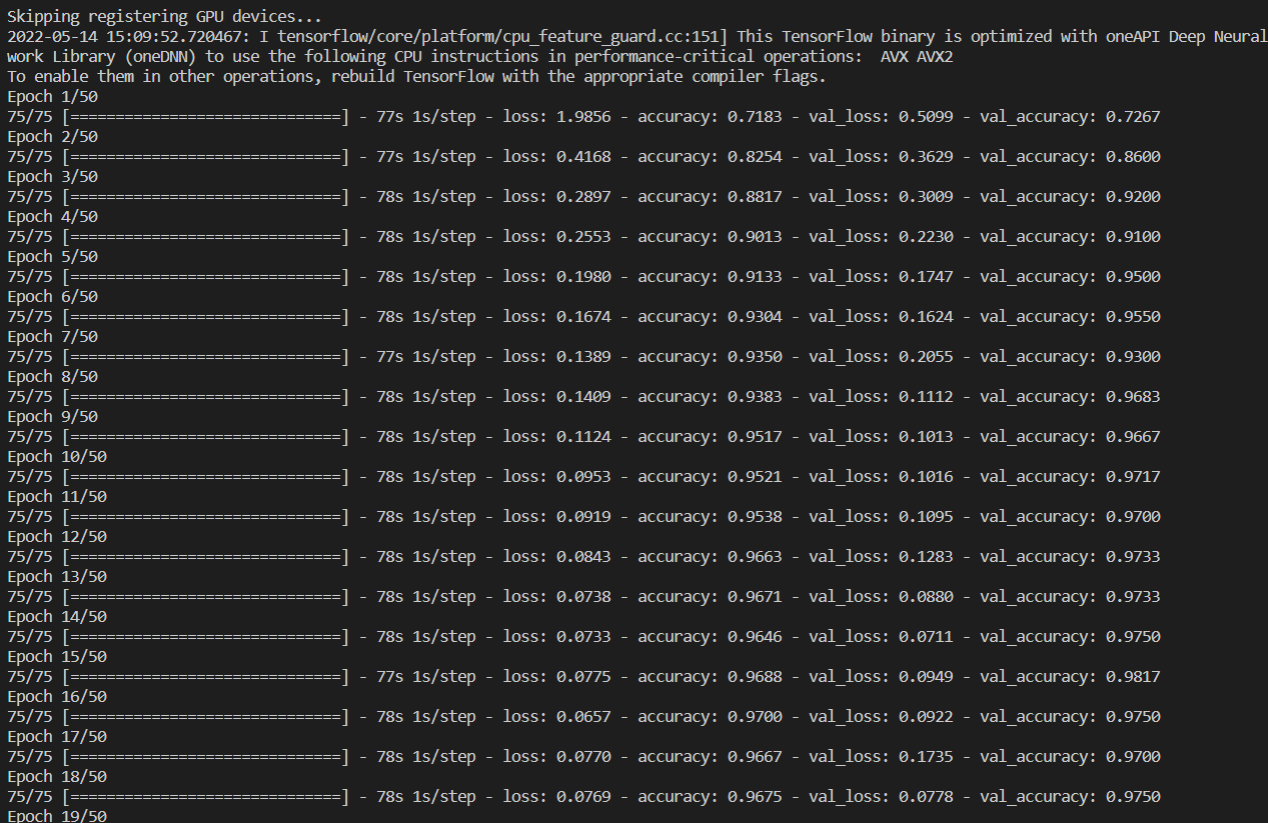


Figure 7.6 Adam optimizer model Epoch Result

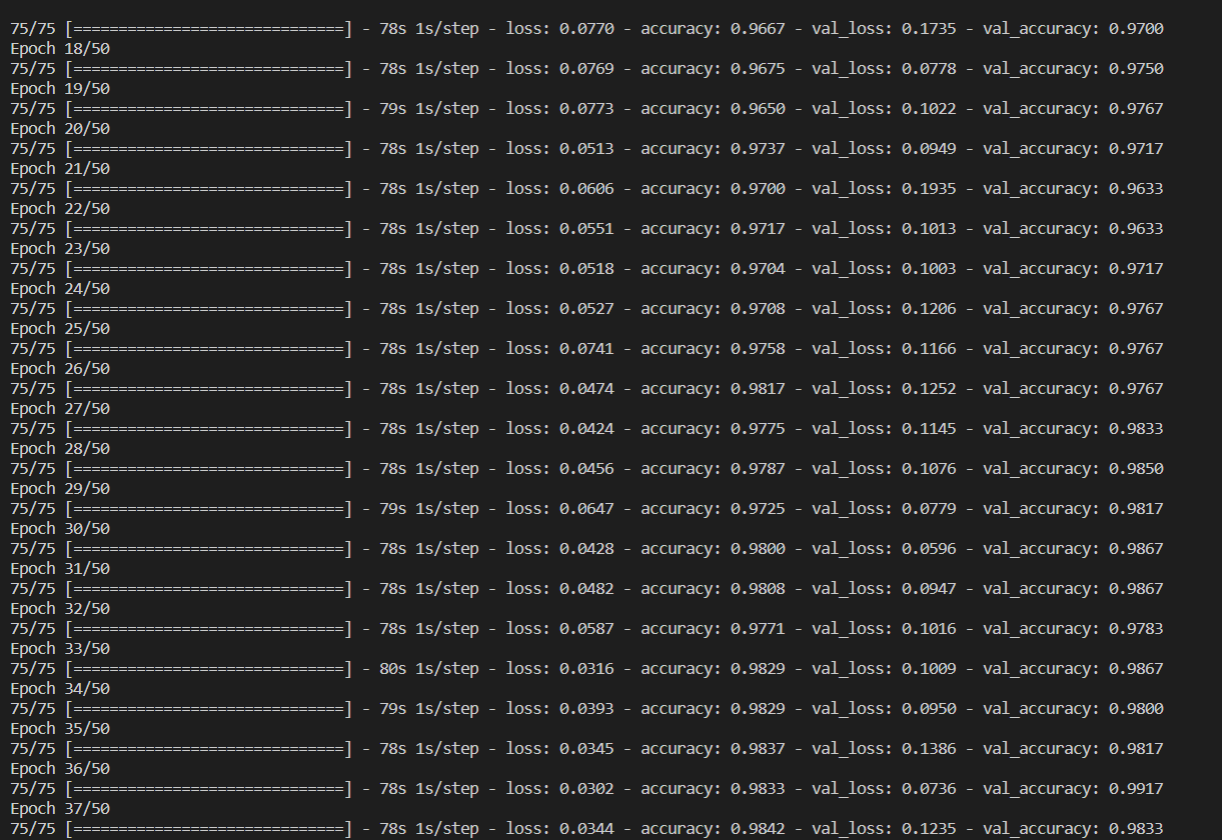


Figure 7.7 Adam optimizer model Epoch Result

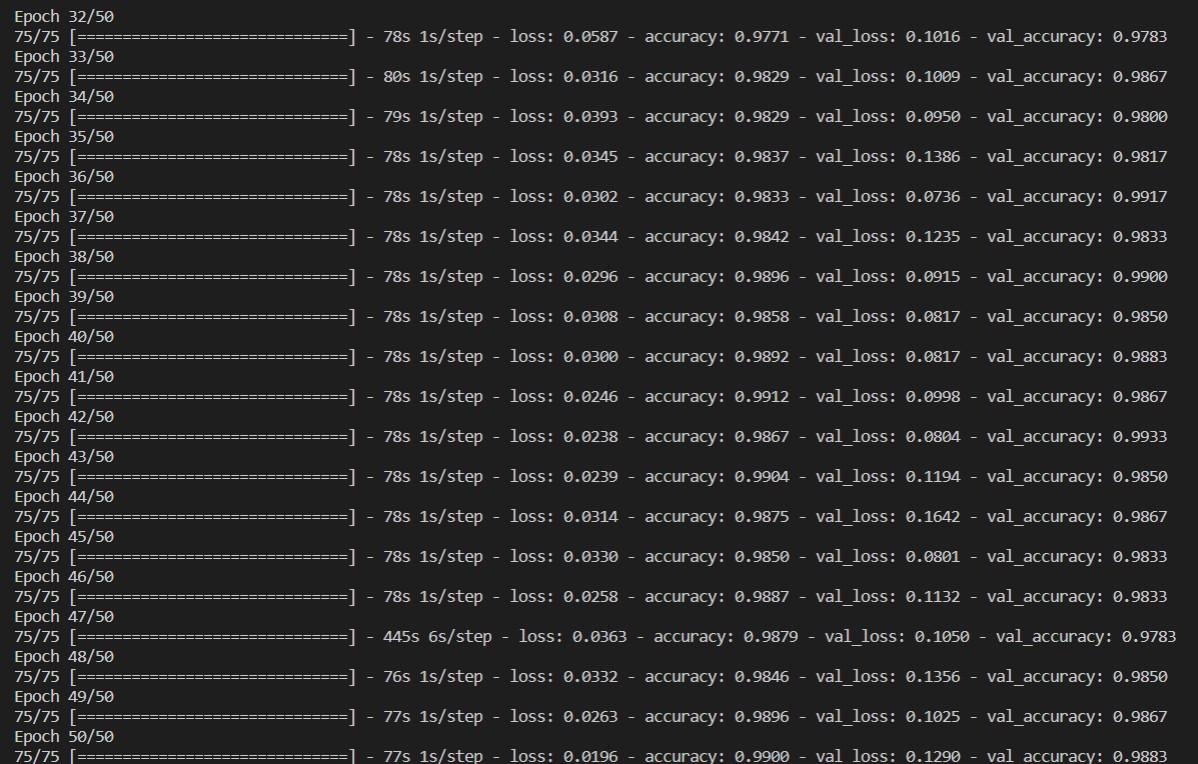


Figure 7.8 Adam optimizer model Epoch Result

#### Adam optimizer Model summary:

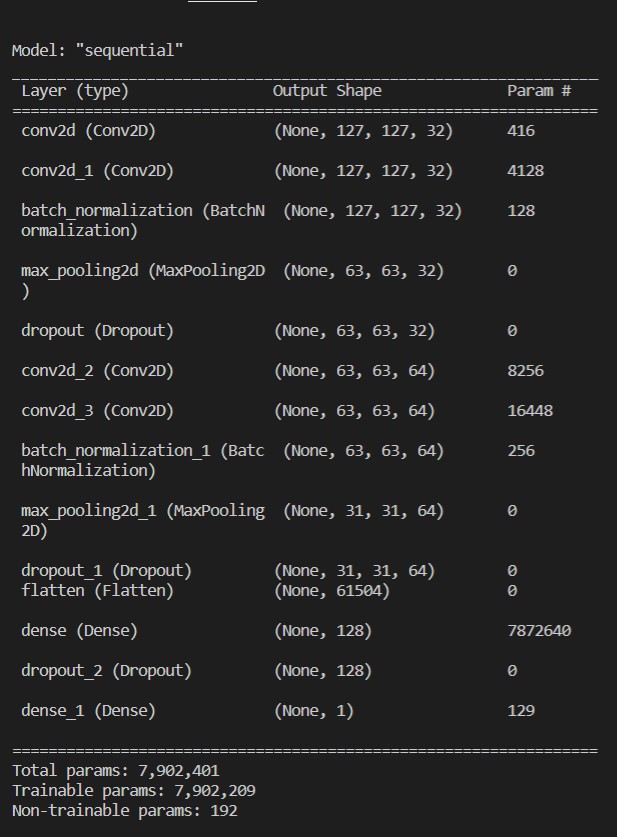


Figure 7.9 Adam optimizer model summary

#### Adam optimizer Model Confusion matrix is shown in the below figure

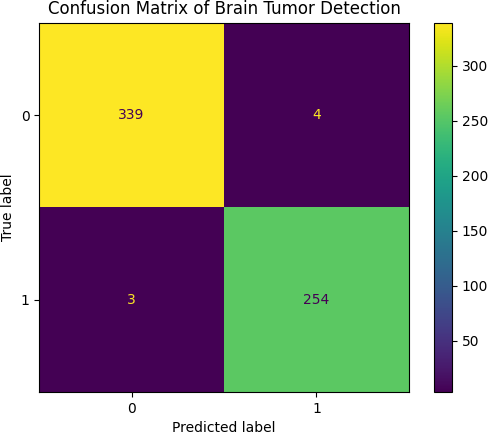


Figure 7.10 Adam optimizer Confusion matrix

#### Adam optimizer model accuracy is shown the below in Figure:

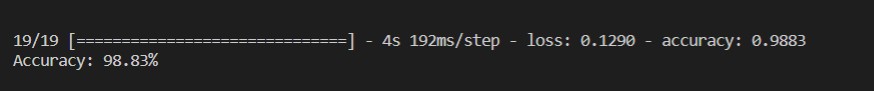


Figure 7.11 Adam optimizer model accuracy

#### Adam optimizer Classification Report is shown the below Figure:

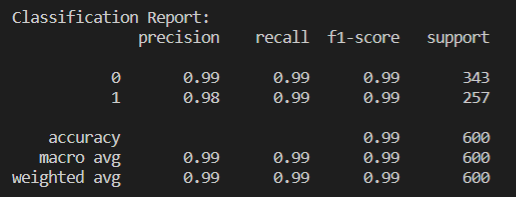


Figure 7.12 Adam optimizer Classification Report

* The results indicate that the Model with Adam optimizer displays accuracy of 98.8% because unlike maintaining a single learning rate via training in SGD and other metrics, it adjusts the learning rate for each network weight independently, resulting in high performance.
* Model with RMSProp optimizer training result, confusion matrix, accuracy and classification reports:

#### RMSProp optimizer training results :

Figure 7.13 RMSProp optimizer model Epoch Result

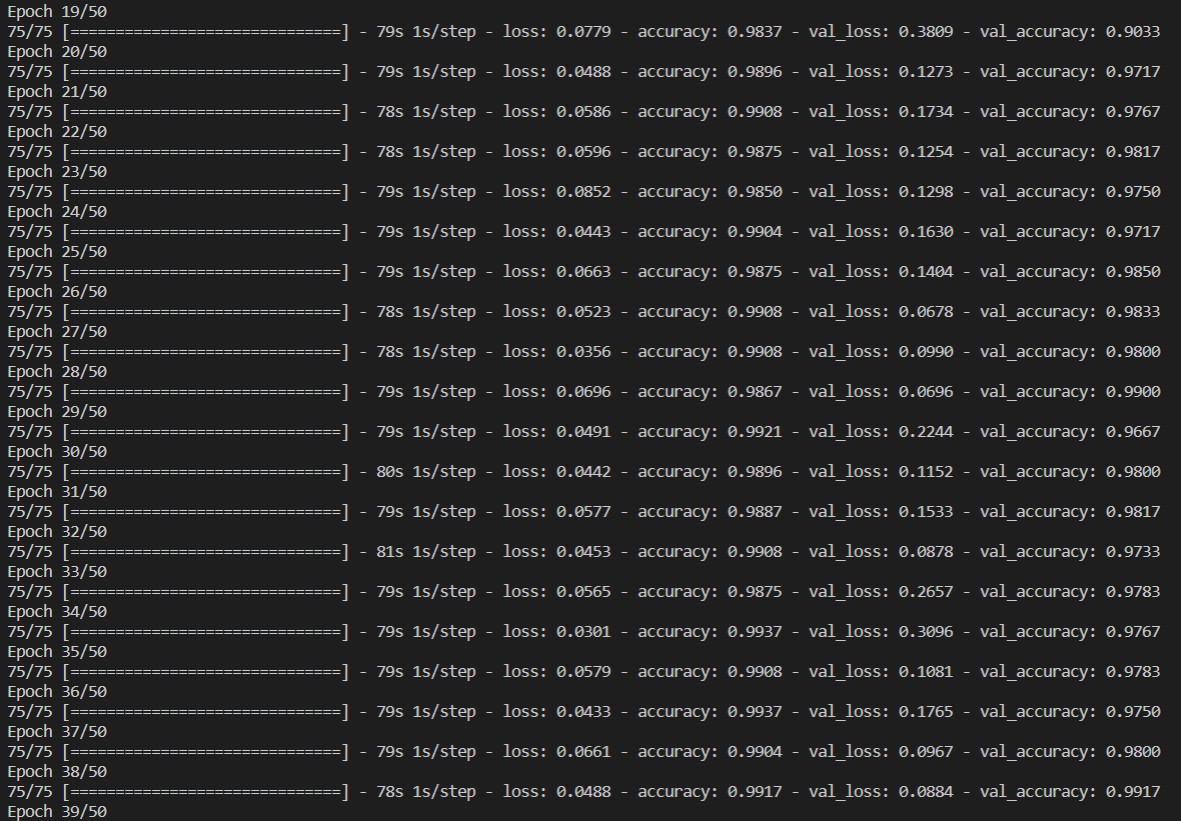


Figure 7.14 RMSProp optimizer model Epoch Result

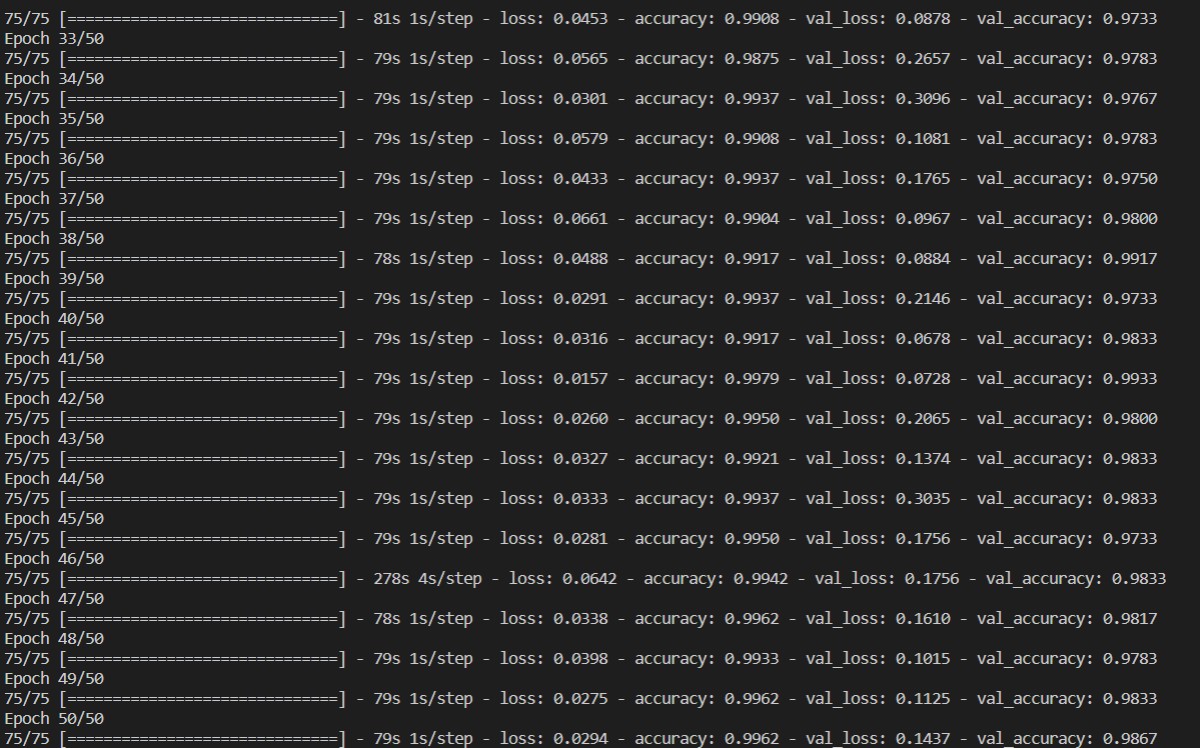


Figure 7.15 RMSProp optimizer model Epoch Result

#### RMSProp optimizer Model summary:

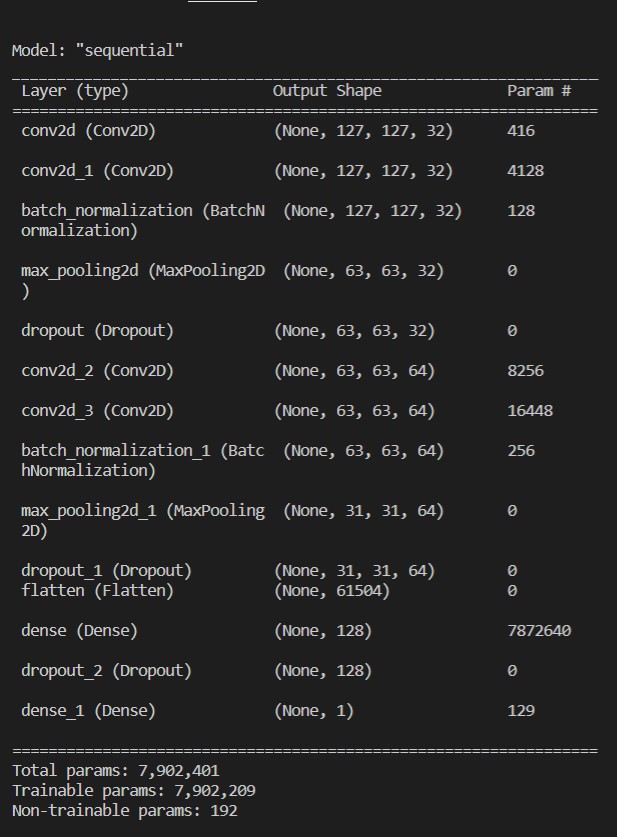


Figure 7.16 RMSProp optimizer model summary

#### RMSProp optimizer model Confusion matrix is shown in the below Figure:

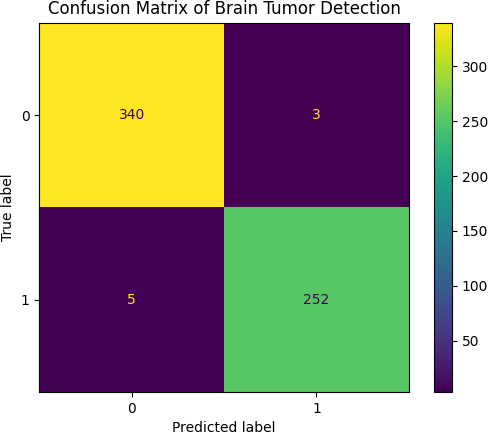


Figure 7.17 RMSProp optimizer model Confusion matrix

#### RMSProp optimizer model accuracy is shown in the below Figure:

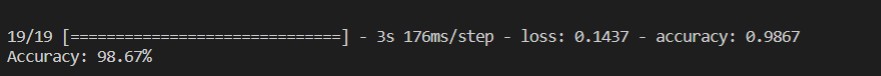


Figure 7.18 RMSProp optimizer model accuracy

#### RMSProp optimizer model Classification Report as shown the below in Figure:

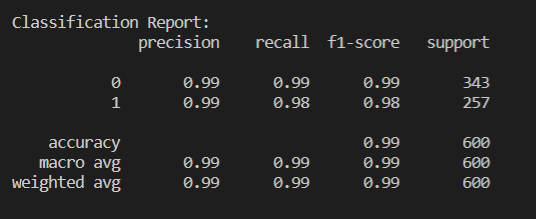


Figure 7.19 RMSProp optimizer model Classification Report

* The results indicate that the Model with RMSProp optimizer displays accuracy of 98.6%. It uses the sign of the gradient adapting the step size individually for each weight, unlike defining a single learning rate and other metrics delivering good performance.
* Model with SGD training result, confusion matrix, accuracy and classification reports:

#### SGD optimizer model training Results:

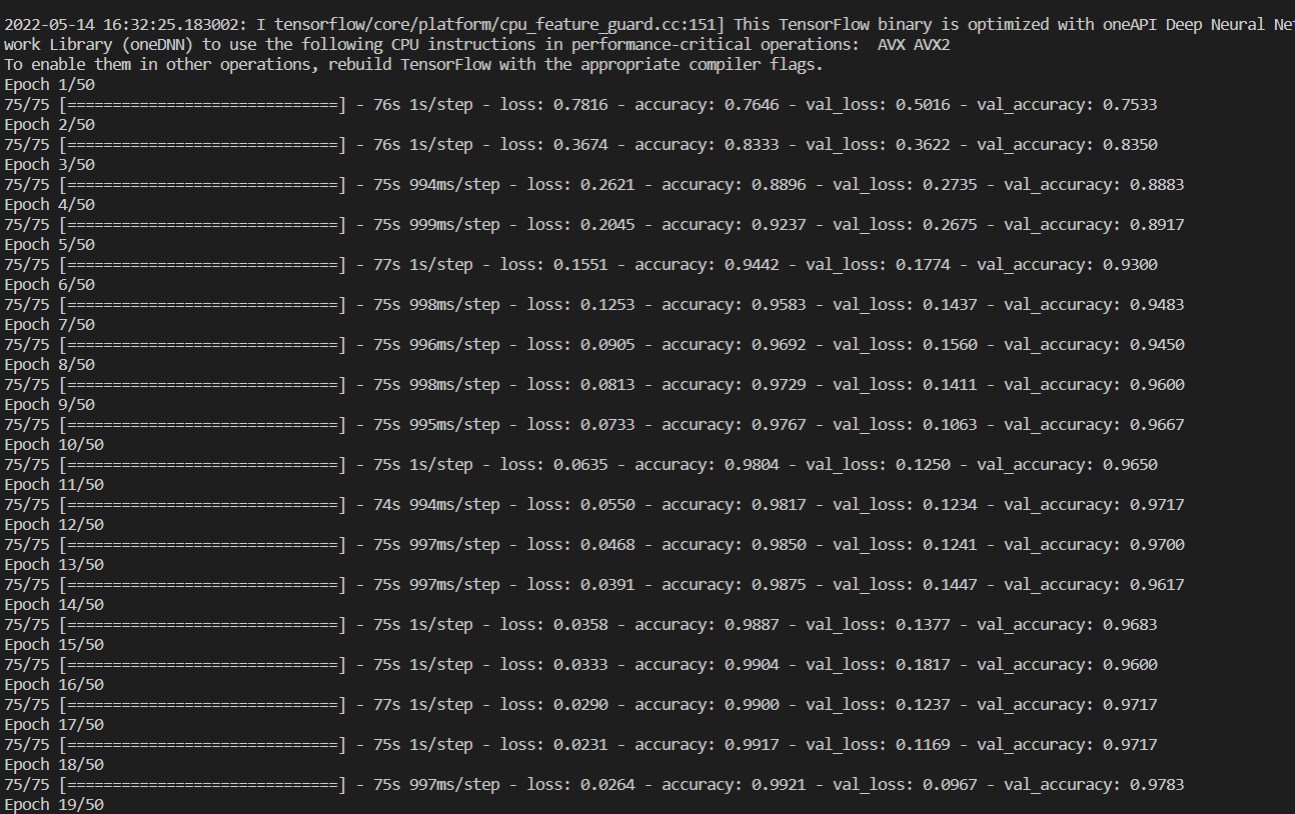


Figure 7.20 SGD optimizer model Epoch Result

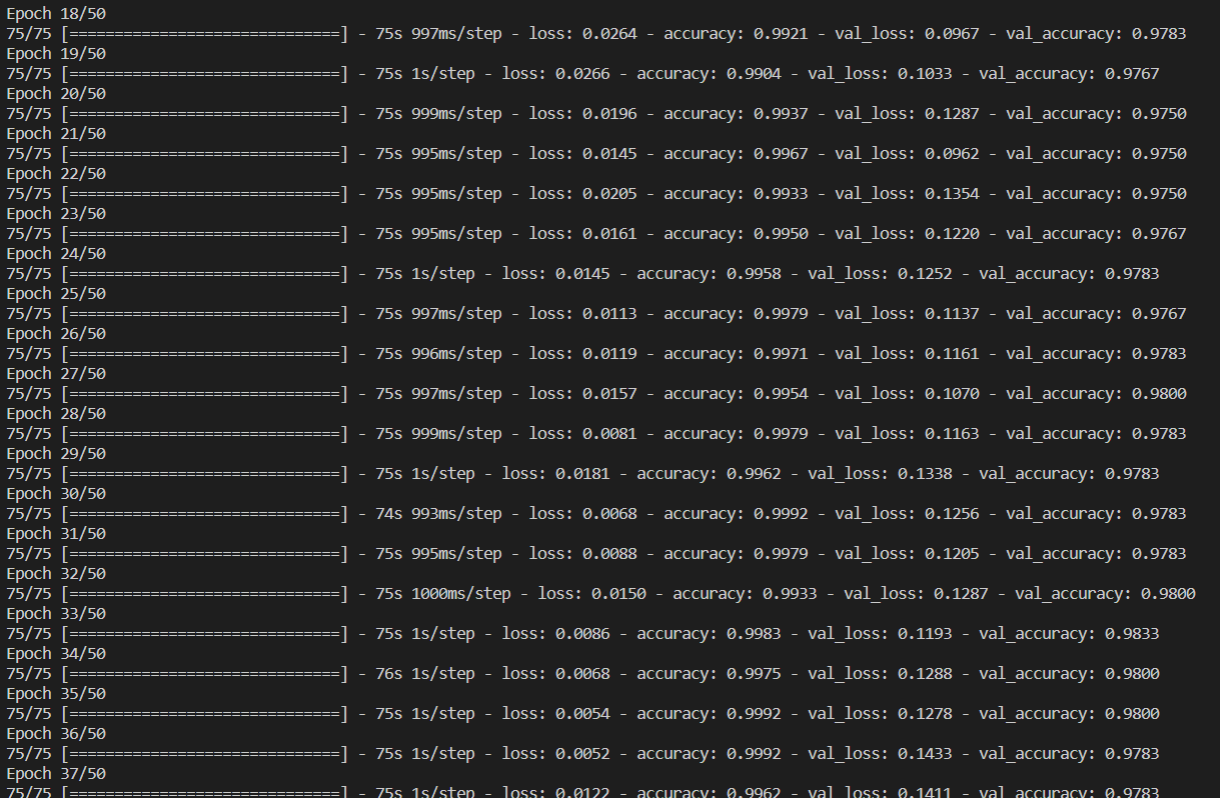


Figure 7.21 SGD optimizer model Epoch Result

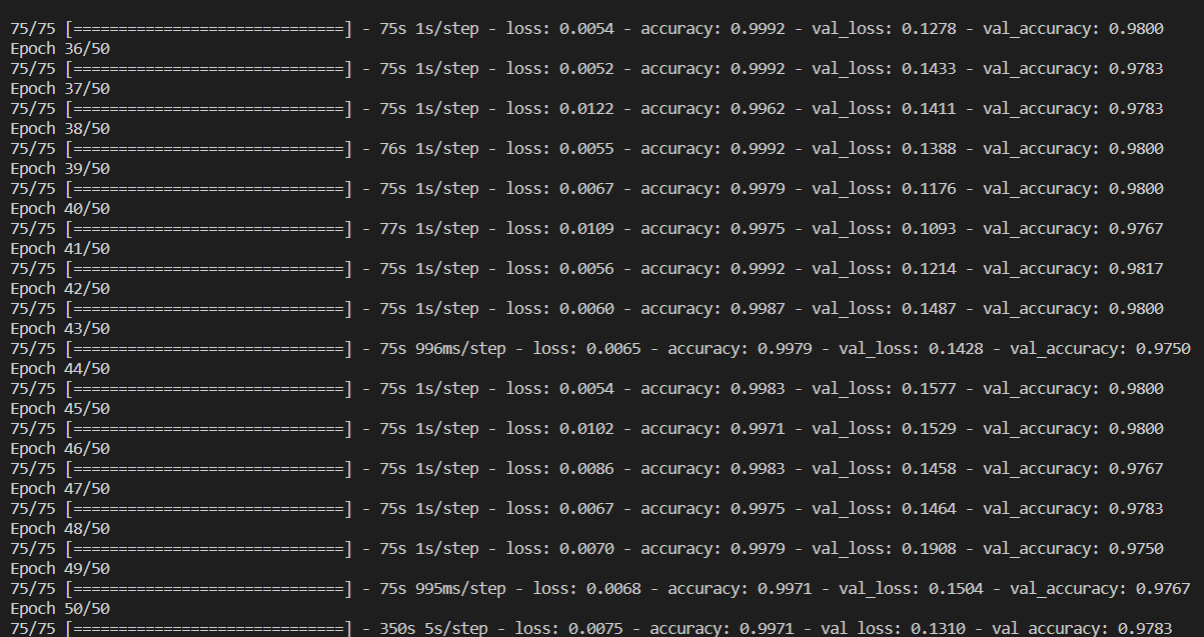


Figure 7.22 SGD optimizer model Epoch Result

#### SGD optimizer Model summary:

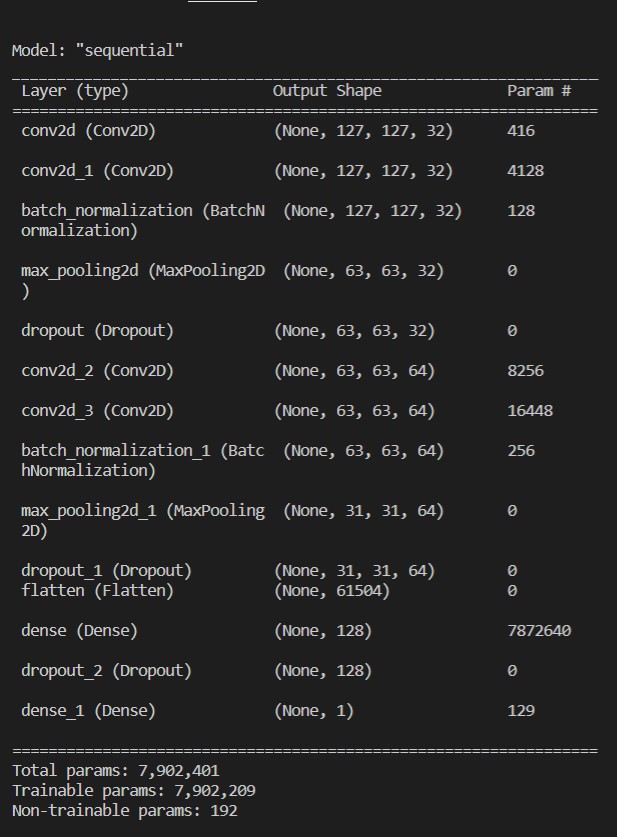


Figure 7.23 RMSProp optimizer model summary

#### SGD optimizer model Confusion matrix is shown in the below Figure:

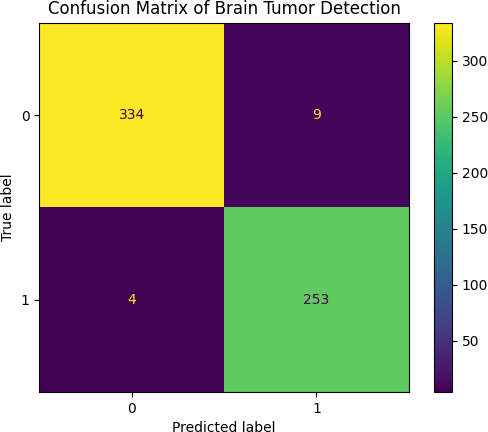


Figure 7.24 SGD optimizer model Confusion matrix

#### SGD optimizer model accuracy as shown the below Figure:



Figure 7.25 SGD optimizer model accuracy

#### SGD optimizer model Classification Report as shown in the below Figure:

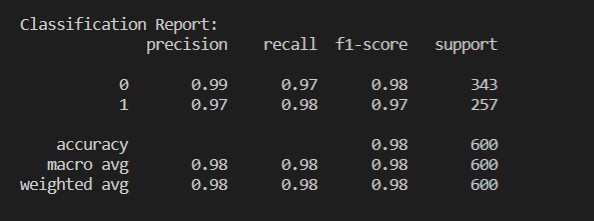


Figure 7.26 SGD optimizer model Classification Report

The results indicate that the Model with RMSProp optimizer displays an accuracy of 97.8% because the model parameters are altered after the computation of loss on each training example, because model parameters are often changed, loss functions have a lot of variance and fluctuation at different intensities, and other measures perform well.

#### Results of models with three different optimizers are shown in table 1:

Table 1: summary of a model with different optimizers

|  |  |  |  |
| --- | --- | --- | --- |
| Optimizers | Accuracy | Precision | F1 Score |
| Adam | 98.8% | 0.99 | 0.99 |
| RMS Prop | 98.6% | 0.99 | 0.99 |
| SGD | 97.8% | 0.98 | 0.98 |

**CHAPTER-8**

# TESTING

### Introduction to Testing

Testing is a process, which reveals errors in the program. It is the major quality measure employed during software development. During testing, the program is executed with a set of conditions known as test cases and the output is evaluated to determine whether the program is performing as expected.

Software testing is the process of testing the functionality and correctness of software by running it. Process of executing a program with the intent of finding an error.

A good test case is one that has a high probability of finding an as yet undiscovered error. A successful test is one that uncovers an as yet undiscovered error. Software testing is usually performed for two reasons.

* + - Defect detection
    - Reliability estimation

### Types of Testing:

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

##### Unit Testing:

Unit Testing is done on individual models as they are completed and become executable. It is confined only to the designer’s requirements.

Each module can be tested using the following two strategies:

##### Black Box Testing:

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program. This testing has been uses to find errors in the following categories:

* 1. Incorrect or missing functions
  2. Interface errors
  3. Errors in data structure or external database access
  4. Performance errors
  5. Initialization and termination errors.

In this testing only the output is checked for correctness. The logical flow of the data is not checked.

##### White Box testing:

In this the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases. It has been uses to generate the test cases in the following cases:

1. Guarantee that all independent paths have been executed.
2. Execute all logical decisions on their true and false sides.
3. Execute all loops at their boundaries and within their operational bounds
4. Execute internal data structures to ensure their valid

##### Integrating Testing:

Integration testing ensures that software and subsystems work together as a whole. It tests the interface of all the modules to make sure that the modules behave properly when integrated together.

##### System Testing:

Involves in-house testing of the entire system before delivery to the user. Its aim is to satisfy the user the system meets all requirements of the client's

specifications.

##### Acceptance Testing:

It is a pre-delivery testing in which entire system is tested at client's site on real world data to find errors.

##### Validation:

The system has been tested and implemented successfully and thus ensured that all the requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input corresponding error messages are displayed.

##### Compiling Test:

It was a good idea to do our stress testing early on, because it gave us time to fix some of the unexpected deadlocks and stability problems that only occurred when components were exposed to very high transaction volumes.

##### Execution Test:

This program was successfully loaded and executed. Because of good programming there were no execution errors.

##### Test Case 1:

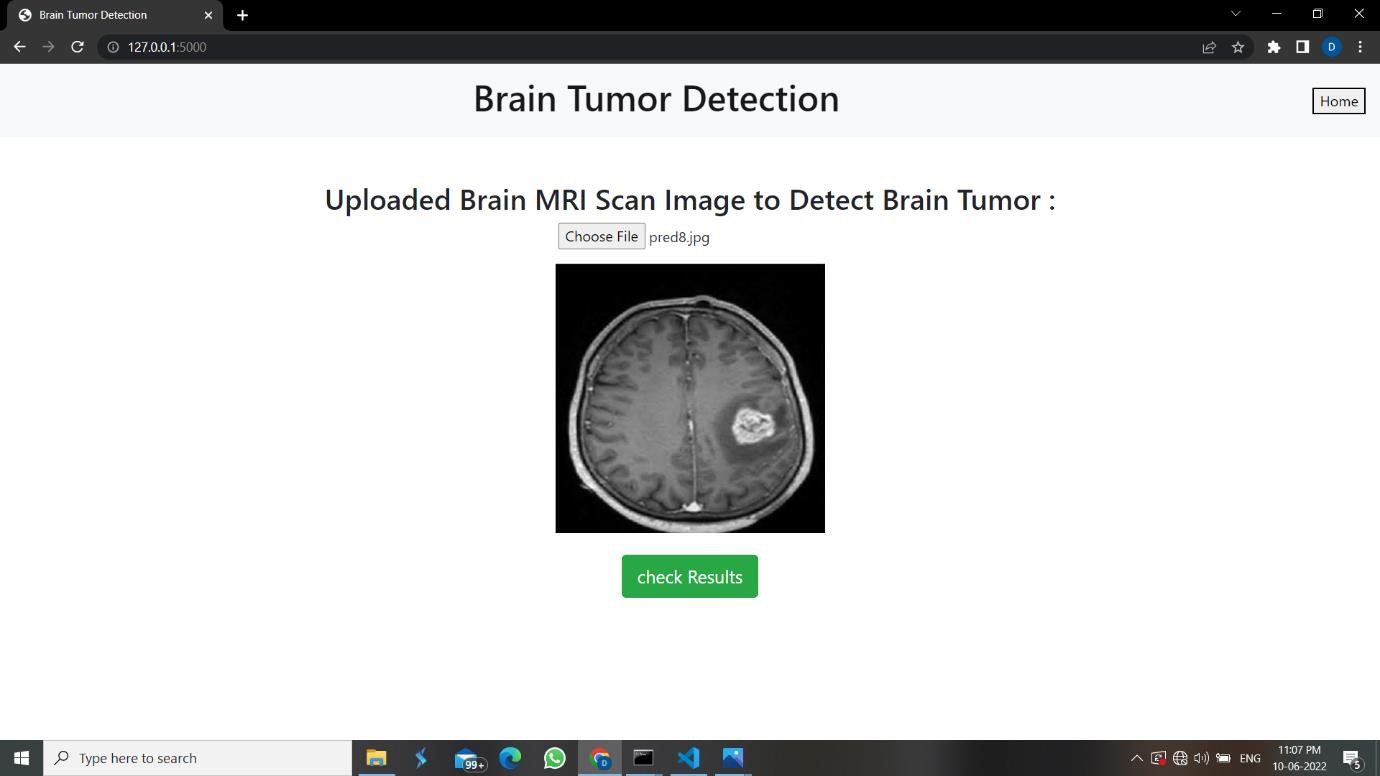
When an image of MRI scan with Brain tumor is uploaded

Figure 8.1 Image of Brain MRI Scan with tumor is uploaded

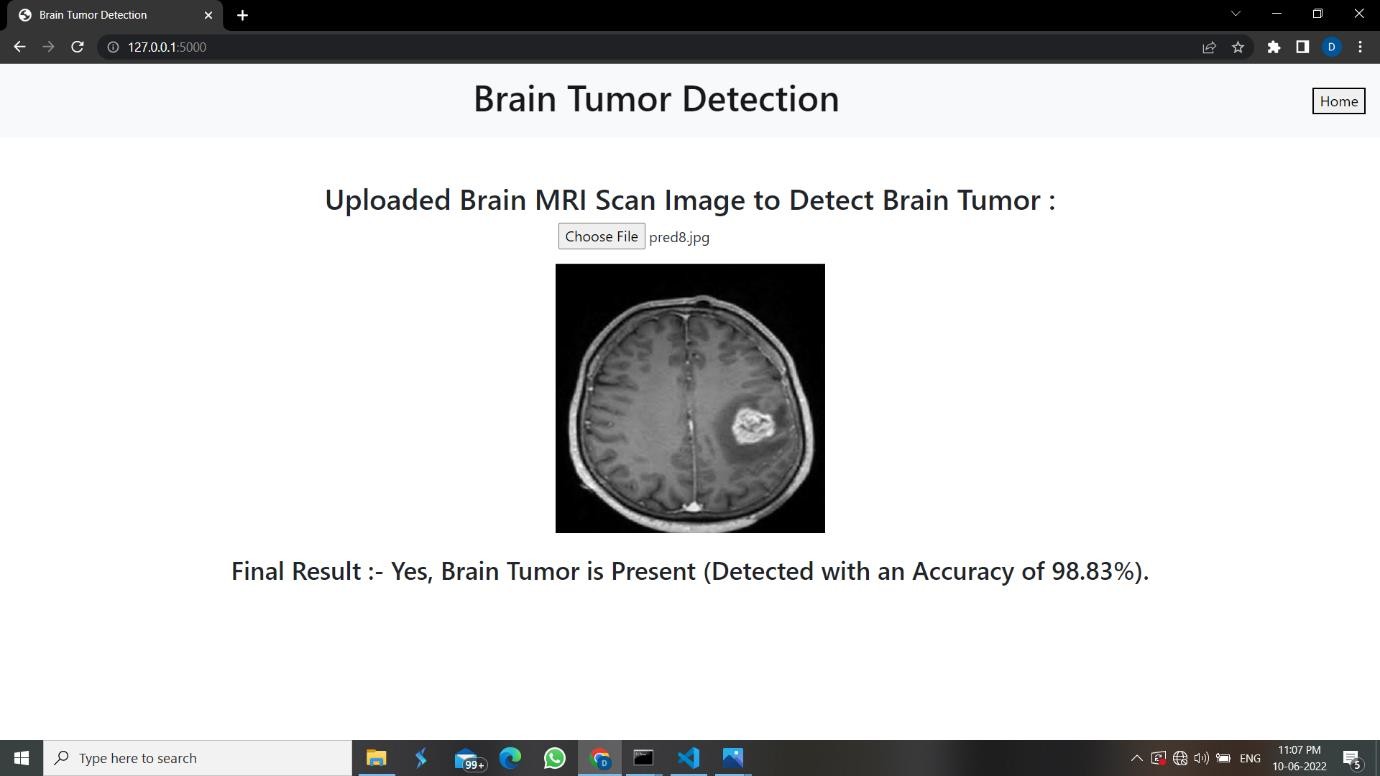


Figure 8.2 Result when Image MRI Image of Brain MRI Scan with tumor is uploaded

##### Test Case 2:

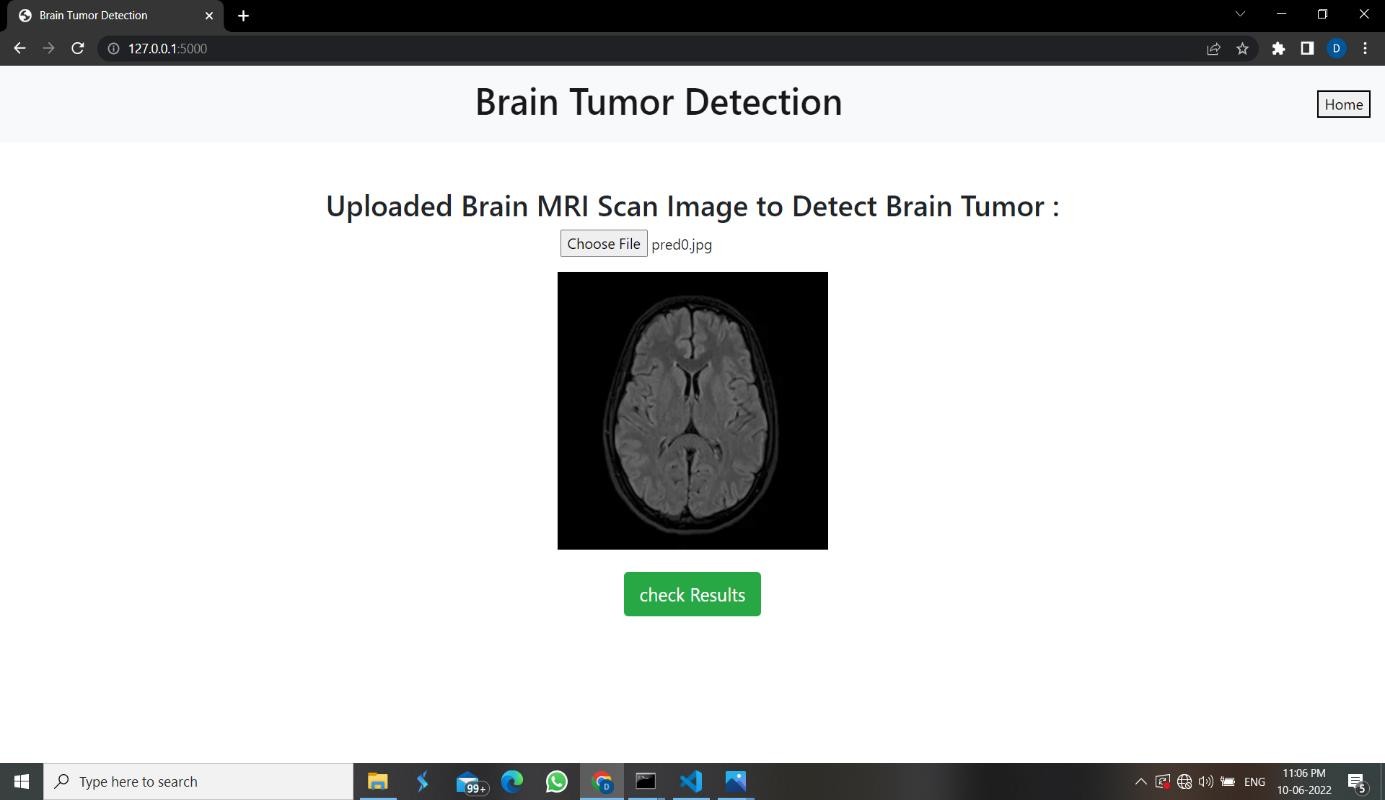
When an image of MRI scan without Brain tumor is uploaded

Figure 8.3 Image of Brain MRI Scan with no tumor is uploaded

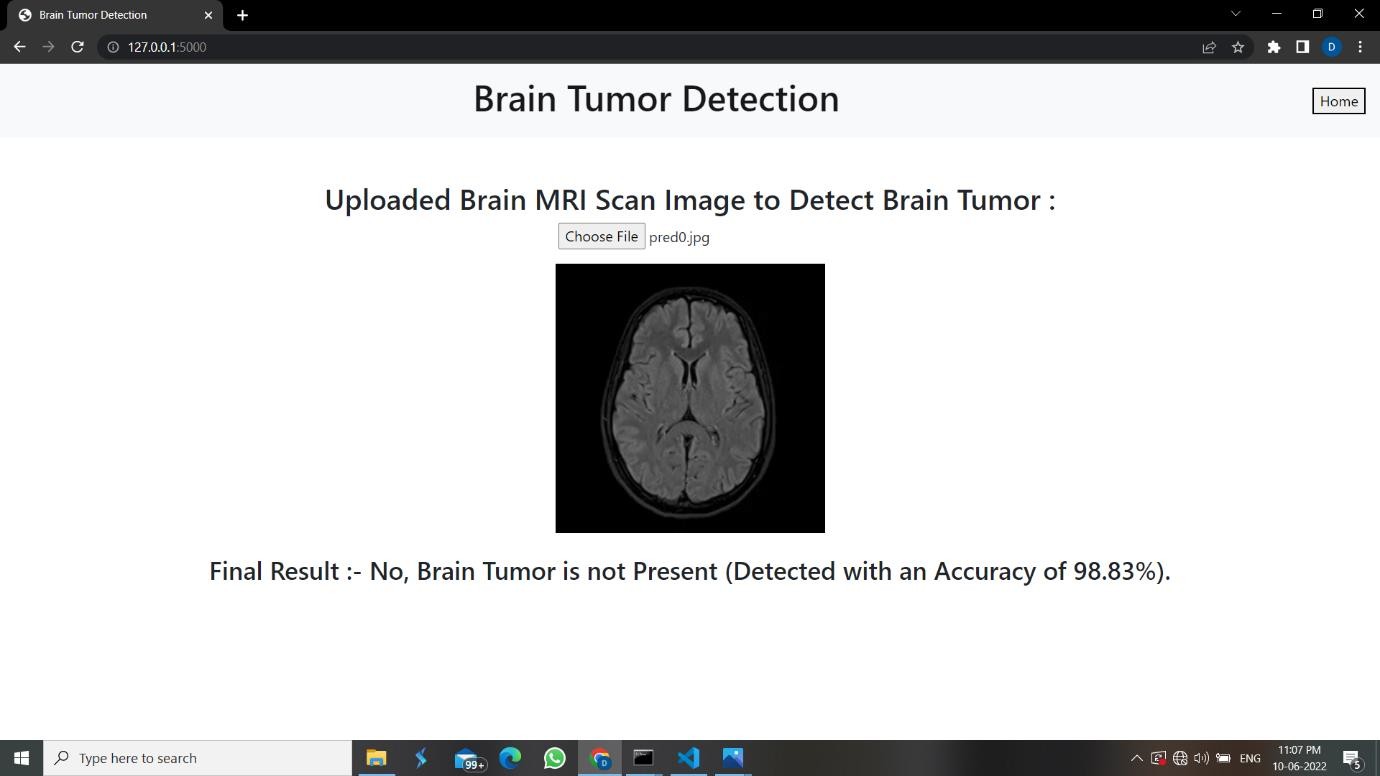


Figure 8.4 Result when Image MRI Image of Brain MRI Scan with no tumor is uploaded

**CHAPTER-9**

# CONCLUSION AND FUTURE ENCHANCEMENT

## Conclusion

Brain tumor detection plays a vital role in the medical field. we detect brain tumors by using deep learning with CNN model and evaluated different optimizers with the help of MRI scans. Results signify that Adam optimizer outperforms compared to other optimizers by the percentage of 98.8% accuracy and also with other metrics.

We created a GUI-based user interface in which we can upload MRI scans and get the results whether tumor is present or not.

## Future scope

Adding on to our work, localization of Brain tumor and the percentage calculation of tumor can be done for better way to analyze the tumor. Web application can be created. so that, it would be more helpful for user.

**CHAPTER-10**

# REFERENCES

### References

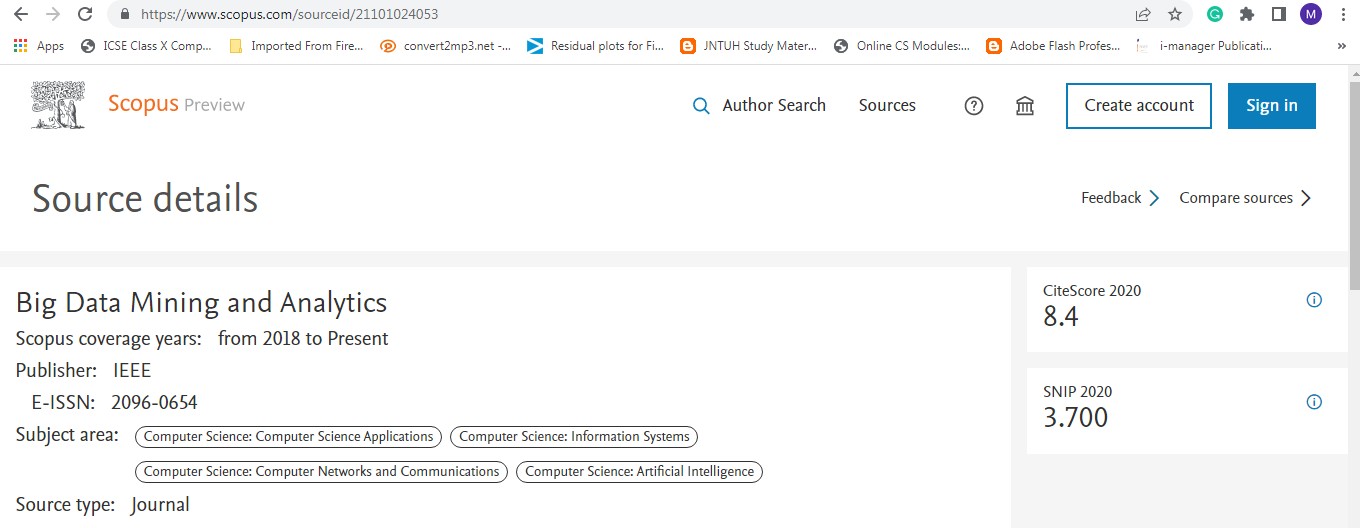
1. <https://en.wikipedia.org/wiki/Deep_learning>
2. <https://www.reachiteasily.com/2021/02/convolutional-neural-network.html>
3. Devkota, B. & Alsadoon, Abeer & Prasad, P.W.C. & Singh, A.K. & Elchouemi, A. (2018). Image Segmentation for Early Stage Brain Tumor Detection using Mathematical Morphological Reconstruction. Procedia Computer Science. 125. 115-123. 10.1016/j.procs.2017.12.017.
4. Minz, Astina, and Chandrakant Mahobiya. "MR Image Classification Using Adaboost for Brain Tumor Type." 2017 IEEE 7th International Advance Computing Conference (IACC) (2017): 701-705.
5. P.S. Mukambika, K Uma Rani, "Segmentation and Classification of MRI Brain Tumor," International Research Journal of Engineering and Technology (IRJET), Vol.4, Issue 7, 2017, pp. 683 – 688, ISSN: 2395-0056.
6. Shika gitte, Dr. BH Chandra Shekhar, " Brain tumor detection using CNN" International journal of creative research thoughts (IJCRT) vol 9, issue 6 June,2021, ISSN:2320-2882.
7. Anjaly Antony, Minla K.S, "Brain Tumor Detection From MRI Images using CNN" International Journal Of Creative Research Thoughts (IJCRT), Volume 10, Issued on 4th April 2022, Issn 2320-2882.
8. Ashwini Patne, Laxmibai, Pooja DH, Joyti patil, "Brain Tumor Detection Using Convolutional Neural Network" International Research Journal of Innovative Research In Engineering and Multidisciplinary Physical Sciences (IJIRMPS), Volume 9, Issued on 4, 2021, ISSN: 2349-7300.
9. Pranav Shetty,Suraj Singh,Rasvi Jambhulkar,Kajal Sheth and Deepali Ujlambkar "Detection of brain tumor using CNN and ML" International Journal of Applied Research Vol , Issue 2021,ISSN:308-313.
10. Dr.K.Kranthi Kumar, Mr.V.Raju, V.Abhinaya, P.Anusha, K.Akaansha, B.Soniya "Detection of Tumor Cells in Brain using CNN", I nternational Research Journal of Engineering and Technology(IRJET), Volume 7 , Issued on 5th may 2020, ISSN:2395-0056.
11. Shreyash Gupta, Shreyas.S, Lalita VP "Segmentation approach for Brain Tumor Detection" International Journal Of Advance Research, Ideas And

Innovations In Technology (IJARIIT),Volume-7,Issued on 4,2021,ISSN:2454-132X.

1. Khalifa Shantta, Otman Basir, "Brain Tumor Detection and Segmentation: A Survey" International Journal Of Technology And Engineering (IRA), Volume-10, Issued on 4-2020,ISSN-2455-4480
2. <https://www.mygreatlearning.com/blog/introduction-to-image-pre-processing/>
3. [https://www.tutorialspoint.com/flask/index.htmhttps://www.tutorialspoint.](https://www.tutorialspoint.com/flask/index.htm) [com/flask/index.htm](https://www.tutorialspoint.com/flask/index.htm)
4. <https://docs.opencv.org/4.x/d0/de3/tutorial_py_intro.html>
5. <https://www.geeksforgeeks.org/os-module-python-examples/>
6. <https://www.tensorflow.org/learn>
7. <https://pillow.readthedocs.io/en/stable/>
8. <https://pypi.org/project/Werkzeug/>
9. <https://www.kaggle.com/ahmedhamada0/brain-tumor-detection>

### Paper Publication

Submission Confirmation



Scopus Proof

**Automated detection of Brain Tumor using Deep Learning technique and assessing the performance using different optimizers**

**Abstract**

A brain tumour is an abnormal growth in the brain, some of which have the potential to develop into cancer. Brain tumours are usually diagnosed using magnetic resonance imaging (MRI) scans. From The MRI Images, information about the abnormal tissue growth in the brain is found. One of the most challenging issues in tumor segmentation is variation in brain tumor size, shape, and form. The tumor area is retrieved from all volume slices using the segmentation approach. Automated defect identification in medical imaging has emerged as anemerging topic in numerous medical diagnostic applications, one of which is determining whether or not a brain tumor exists. In this research, we propose a Deep learning algorithm such as convolution neural networks (CNN) to detect a tumor in the brain with different optimizers. The results of applying the suggested approach toMRI scans from the Kaggle dataset determine whether or not a brain tumor is present. The accuracy of 98.8% using the Adam optimizer was observed best among the three optimizers considered.

**Keywords:**

***Brain tumor, Medical Imaging, Deep Learning, Convolutional Neural Network, Optimizers.***

##### Introduction

More than 190,000 patients worldwide arediagnosed with primary or metastatic brain (secondary) tumors each year. A brain tumour is a collection of irregular brain cells that forms a mass.Tumors of the brain exist in a wide range of forms and sizes.

There are two types of cancer cells: benign (noncancerous) and malignant (cancerous) (malignant). Brain tumours can develop in the brain and spread to other parts of the body (primary brain tumours) (metastatic brain tumors). Ignoring thefact that the genesis of brain tumours is unknown, there are numerous features among patients who get them. It can impact anyone, whether they are young or old.

To lower the chances of death, the tumour region must first be identified. Computed Tomography (CT scan) and Magnetic Resonance Imaging (MRI) are used to monitor patients physically. Because MRI images reveal the structure, size, and location of the tumour in the brain, they will make it so much easier to diagnose the tumour and plan the surgical procedure to remove it. In a variety of medical cancer diagnosis and therapy, machine learning-based automatic defect diagnosis in medical imaging became a hot research topic. Its significance in MRI brain tumour detection is crucial because it identifies irregular tissues that must be examined when chemotherapy is planned.

Automatic digital and electronic sickness detection and treatment based on medical vision analysis, according to a recent survey, might be a powerful device since it would minimize radiologist work while also increasing quality. Secondly, if computer algorithms can generate accurate and consistent tumour representation metrics, these automated assessments mightconsiderably benefit in the treatment of brain tumours by eliminating the burden of manual tumour depiction from clinicians. Machine Learning (ML) is a kind of AI that allows computers to learn for itself, communicate, learn some new data from the previous data stored, and evolve and gradually improve.

For health care specialists, manually detecting a brain tumour is a time-consuming and tedious task. As a result, for this project, modern computer techniques based on Artificial Intelligence are applied. Artificial intelligence (AI) refers to the simulation of human intelligence processes by computers, specifically computer systems. Deep Learning is an branch of machine learning that deals with artificial neural networks modelled after the structure and brain function.

In our study, we used the Convolutional Neural Network architecture to detect and classify brain tumours with demonstrated accuracy. For image classification, image processing, face identification, and other applications, convolutional neural networks evaluate densely connected data. It's a specialised 3D structure with specialised NN for analysing the RGB layers of a picture. Also, we assess the performance of CNN with different

optimizers such as Adam, Root Mean Square, and Stochastic Gradient Descent. We created a GUI- based user interface in which we can upload MRI scans and get the results whether there is a tumor.

The rest of the research is structured as follows. The literature that supports the hypothesis

is presented in the next section. Section 3 explains the research approach and methodologiesemployed. The findings are presented in Section 4, followed by a discussion of the findings. The final portion discusses the conclusion and future directions

##### Related Work

In the present day, the field of Medical Imaging diagnosis using AI has gained a lot ofimportance, and it is the need of the hour to overwhelm the burden of health professionals. Most researchers have proposed various approaches to medical diagnosis earlier, and some of them are listed below.

B. Devkota et al. [1] suggested a computer-aided detection (CAD) method for detecting aberrant tissues using Morphological procedures. Among the several segmentation procedures available, morphological opening and closure operations are recommended because they need less processing time while extracting tumor regions with minor defects.

Astina Minz et al. [2] used the AdaBoost gadget mastering method to build an effective automated classification solution for brain images. The suggested system is divided into three parts. Pre- processing removed noise from the datasets and transformed photos to grayscale. The pre-processed picture includes median filtering and thresholding segmentation.

Mukambika et al. [3] proposed a method for determining whether or not a tumour is present at a later stage. The Level set technique, discontinuous wavelength transformations (DWT), and K-method segmentation algorithms are all part of their planned work, which is a comparative evaluation oftumour identification approaches utilising MR images. The feature extraction process is then followed by SVM classification.

Shikha Gitte et-al. [4] described various techniques to classify brain tumors, therapies improving patient's quality of life

Anjaly Antony et al. [5] present the implementation of a CNN for diagnosing brain tumors from MRI images. If the brain tumor is predicted, the position and the size of the tumor can be identified.

Ashwini patne et al. [6] present the process of detecting a tumor in the brain with image processing techniques.

Pranav Shetty et al. [7] present the segmentation technique of MRI based on CNN for spatial and structural changeability of brain tumors

Dr. K. Kranthi Kumar et al. [8] predict the magnitude of brain tumors using a Convolution Neural Network algorithm that provides us with reliable results.

Shreyash Gupta et al. [9] describe the importance of artificial intelligence in detecting tumor regions using computer vision and image segmentation.

Khalifa shantta et al. [10] reviews the state-of-the- art brain tumor detection techniques and highlights their pros and cons.

##### Methods

We used Kaggle dataset [11], in this data set we have 3045 images as data collection.

The CNN architecture for Brain tumor Detection and Classification is used in this work.

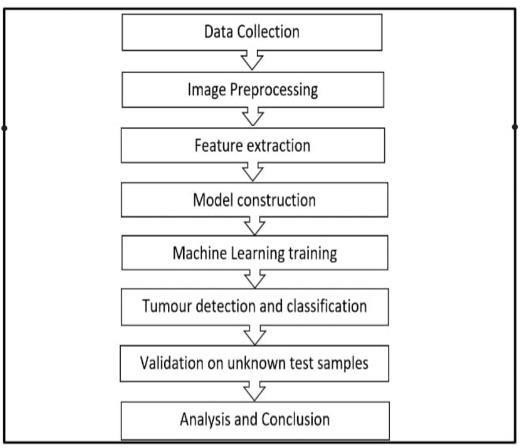
* 1. **Convolutional Neural Network**

Creative models depend on hand-crafted features and prejudicial models associated with classical learning techniques such as support vector machine (SVM), Random Forest (RF), and Convolutional Neural Networks are the two different types of classifier models (CNN). In contrast to generative modelling approaches, discriminative models (CNN) are utilised in brain tumour segmentation because they rely onextracting a large variety of low-level image qualities and explicitly modelling the link between these features and the label of a single voxel.

Convolutional neural networks are used to evaluate densely related data in picture classification, image processing, face recognition, and other applications. It's a three-dimensional framework that examines RGB layers in an image applying particular neural networks. It examines one image at a time, recognizing and collecting crucial parts and utilizing them to characterize the image, unlike other algorithms. Convolutional Neural Networks (ConvNets) learn mid- and high-level outputs or abstractions from receiving training data.

* 1. **Workflow**

The work is an image processing technique for brain tumor identification and localization. Pre- processing, edge detection, and segmentation are among the phases of the method. The pre- processing stage involves converting the original image to greyscale and, if necessary, removing noise. After that, image enhancement methods are used to improve the appearance. Edge detection is then performed using Sobel and Canny algorithms. Finally, in the MRI images, segmentation is used to emphasize the tumor using morphological operations toward the afflicted region. The workflow diagram is shown below in Fig.1.



**Fig.1 Work Flow Diagram**

* 1. **Image Pre-Processing**

Rescaling, Binary Thresholding, noise reduction and morphological operations like as erosion, dilation, and contour shaping these are part of our pre-processing (edge-based methodology).In the very first pre-processing phase, the image's memory size was reduced by scaling the pixels'grey values between 0-255. Because the brain isnot separated as a tumour in this scenario, we have used Gaussian blur filter to remove the noise. It is considered as produce better results than theMedian filter.

* 1. **Segmentation**

Brain tumour segmentation is a method for differentiating tumour tissues (Region of Interest - ROI) from functional brain tissues and concrete brain tumours using MRI images or other new process. Its process includes measuring the similarity measure between the objects and grouping the items with the greatest similarity or calculating the dissimilarity measure between both the objects and isolating the most dissimilar objectsin the region to group similar kinds of subjectsinside a snapshot. The 2 categories of segmentationalgorithms are bi- clusters (2 sub-parts) and multi-

clustered segmentation algorithms (more than two sub-parts). Other segmentation methods including Edge Detection, Region Growing, Watershed, Clustering via FCM, Spatial Clustering, Split and Merge segmentation, and Neural Network via MLP (ANN+DWT).

The tumour area is detected from the brain image using the Binary Thresholding strategy (through the Region Growing method). Based on the threshold settings, this approach converts a greyscale image to a binary image.

* 1. **Feature Extraction**

To extract quantifiable characteristics of modifications / deviations in resolution that aren't readily visible, mathematical analytical feature extraction approaches have been applied. Entropy, RMS, softness, skewness, symmetry, kurtosis, mean, texture, variance, centre of gravity, central tendency, IDM (reverse moment), correlation, energy, homogeneity, dissimilarity, contrast, shadow, prominence, Excessiveness, boundaries, areas, and other traits are instances of these attributes.

Feature Extraction is all about identifying abnormalities. Because we'll be employing a classifier to categorize the photographs and these qualities will be utilized to train the classifier, we'll need to extract specific features from the photos. I decided to exclude GLCM from the equation (texture-based feature). Gray-level co-occurrence matrices are constructed using a pixel iteration rate (GLCM) similar to the probability density function. GLCM is a statistical texture analysis method that considers the spatial relationships between pixels.

Finally, the model has been trained and evaluated using the dataset.

* 1. **Optimizers**

Optimizers are techniques or approaches that adjust the characteristics of your neural network, such as weights and learning rate, to decrease losses.

* + 1. **Adam Optimizer**:

Adaptive moment estimate is the source of the word adam. To update network weights during training, this optimization approach further develops stochastic gradient descent (SGD). Unlike SGD, Adam optimizer changes the learning rate foreach network weight individually rather than maintaining a single learning rate throughout the training.

Adma Deep Learning Optimizer

--- (1)

The adam optimizer's operation is represented by equation (1) above. The decay rates of the average of the gradients are represented by B1 and B2.

* + 1. **RMS Prop (Root Mean Square) Optimizer:**

The RMS prop is a popular optimizer among deep learning enthusiasts. The work of RMS prop is a natural continuation of that of RPPROP. RPPROP solves the problem of fluctuating gradients. The problem with the slopes is that some were little and others may be rather significant. As a result, using a single learning rate might not be the best choice. RPPROP adjusts the step size for each weight based on the gradient sign, and the RMS Prop optimizer is represented by equation (2) below.

RMS Prop(Root Mean Square) Deep Learning Optimizer

------(2)

RMS Prop(Root Mean Square) Deep Learning Optimizerwhere the forgetting factor is called gamma. The equation(3) below is used to update the weights.

-----(3)

* + 1. **Stochastic Gradient Descent:**

Gradient descent is a variant of stochastic gradient descent where the learning rate is decreased as the error gets smaller. The model tries to update its parameters more often. The model parameters are updated once each time a training example's loss is computed. Since the SGD algorithm updates the model parameters 1000 times per cycle, it will update the model more frequently than the Gradient Descent algorithm.

**θ=θ−α**⋅∇**J(θ;x(i);y(i))** (4)

where {x(i) ,y(i)} are the training examples**.** Loss functions have a lot of unpredictability and fluctuation at different intensities because model parameters are always changing.

##### Results and Discussion

We used CNN with several optimizers to identify the tumor in the brain, and we compared the suggested approaches to some existing models. According to the findings, the new framework outperforms and improves the prior techniques. The best results of the created approach are

obtained by comparing the findings to existing work in the literature. Our suggested approach has predictive value in identifying tumors in patients with brain tumors. Doctors and clinics will benefit from the suggested paradigm.

A confusion matrix is a matrix that is being used to judge the effectiveness of classification techniques on a number of experimental data. They could only be determined just after true values of the testing data are known. The matrix itself is simple to understand, but the terms involved can be confusing. The metrics used for the performance analysis from the confusion matrix is as follows:

Accuracy: It gives you the overall accuracy of the model

Accuracy= (TP+TN)/(TP+TN+FP+FN)

Precision: Accuracy of positive predictions. Precision = TP / (TP + FP)

Recall: Correctly detected positives as a percentage. Recall = TP/(TP+FN)

F1 score: The F1 score is a weighted harmonicmean of accuracy and recall, with 1.0 being the highest and

0.0 being the worst.

(Recall \* Precision) / (Recall + Precision) = F1 Score

Support: The number of actual instances of the class in the supplied dataset is referred to as support.

Macro averaging: To calculate the macro-averaged F1 score, the arithmetic mean (also known as the unweighted mean) of all per-class F1 scores is employed (or macro F1 score).

Weighted Average: The weighted-average F1 score is calculated by adding all per-class F1 scores together while taking into account each class's support.

We used three different optimizers in this project, i.e., ADAM, RMS Prop, and SGD. Among the three optimizers, ADAM produces the best accuracy results.

* 1. Model with Adam optimizer accuracy, confusion matrix, and classification reports:

Adam optimizer Confusion matrix is shown the below Fig.2.

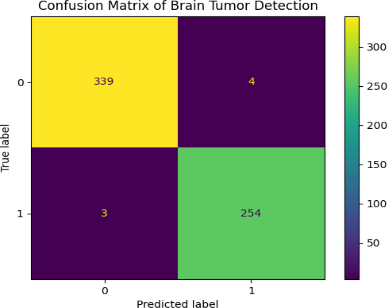
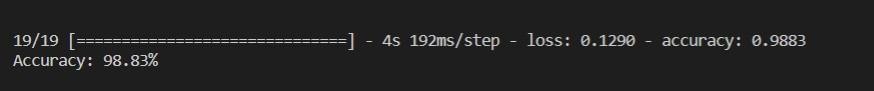


Fig.2 Adam optimizer Confusion matrix Adam optimizer model accuracy is shown the



below in Fig.3:

Fig.3 Adam optimizer model accuracy

Adam optimizer Classification Report is shown the below Fig.4.

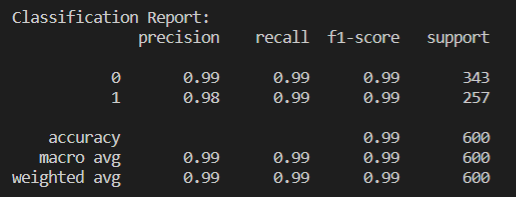


Fig.4 Adam optimizer Classification Report

The Model with Adam optimizer has a 98.8 percent accuracy because, rather than maintaining a single learning rate via SGD and other measures, it changes the learning rate for each network weight individually, resulting in excellent performance.

* 1. Model with RMSProp optimizer accuracy, confusion matrix, and classification reports:

RMSProp optimizer model Confusion matrix is shown in the below Fig.5

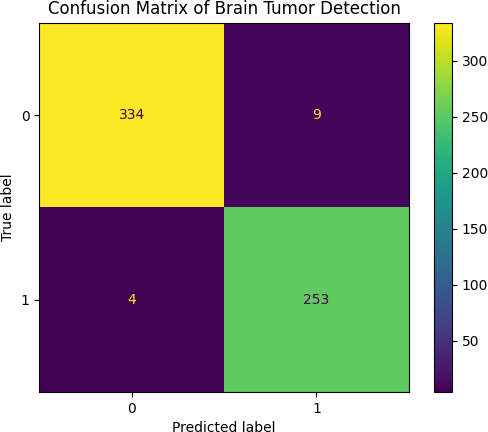


Fig.5 RMSProp optimizer Confusion matrix

RMSProp optimizer model accuracy is shown in the below Fig.6

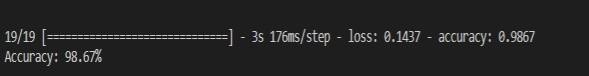


Fig.6 RMSProp optimizer model accuracy

RMSProp optimizer model Classification Report is shown the below in Fig.7

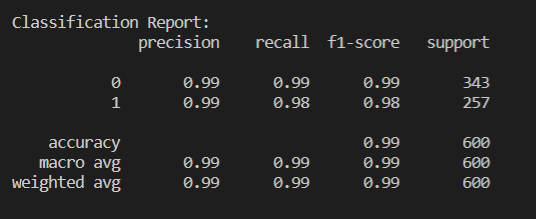


Fig.7 RMSProp optimizer model Classification Report

The findings show that the Model with RMSProp optimizer has a 98.6 percent accuracy. Rather than setting a single learning rate and other performance indicators, it leverages the sign of the gradient to adjust the step size for each weight independently.

* 1. Model with SGD optimizer accuracy, confusion matrix, and classification reports:

SGD optimizer model Confusion matrix is shown in the below Fig.8

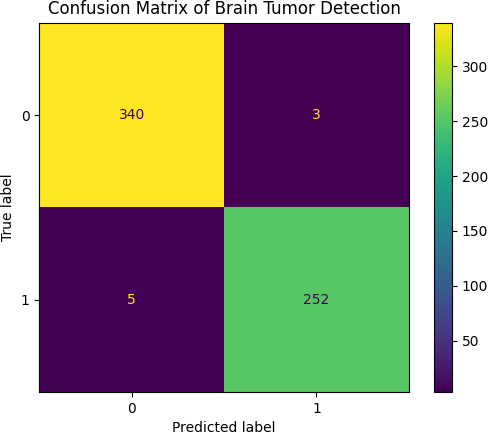


Fig.8 SGD optimizer model Confusion matrix

SGD optimizer model accuracy is shown the below Fig.9

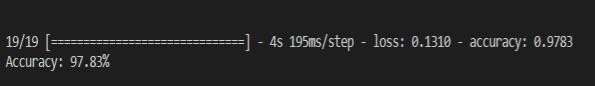


Fig.9 SGD optimizer model accuracy

SGD optimizer model Classification Report is shown in the below Fig.10

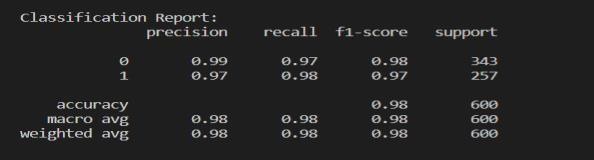


Fig.10 SGD optimizer model Classification Report

Because model parameters are regularly modified after loss calculation on each training sample, loss functions have a lot of volatility and fluctuation at different intensities, and other metrics perform well, the Model with RMSProp optimizer has a 97.8% accuracy.

Results of models with three different optimizers are shown in table 1:

Table 1: summary of a model with different optimizers

|  |  |  |  |
| --- | --- | --- | --- |
| Optimizers | Accuracy | Precision | F1  Score |
| Adam | 98.8% | 0.99 | 0.99 |
| RMS Prop | 98.6% | 0.99 | 0.99 |
| SGD | 97.8% | 0.98 | 0.98 |

##### Conclusion

The identification of brain tumours is critical in the medical sector. In our study, we used deep learning using a CNN model to identify brain cancers and MRI images to assess alternative optimizers. The results show that the Adam optimizer surpasses other optimizers by a percentage of 98.8 percent accuracy and other criteria.

In this paper, we detected a brain tumor in the brain with the help of CNN. We have tested different optimizers. We created a GUI-based user interface in which we can upload MRI scans and get the results whether there is a tumor.

This work can be extended in the future, and we can find the tumor percentage in the brain, find the size of the tumor in the brain, and find which stage of tumor the patient is by testing the MRI scans.

##### References

1. 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.

1. Minz, Astina, and Chandrakant Mahobiya. "MR Image Classification Using Adaboost for Brain Tumor Type." 2017 IEEE 7th International Advance Computing Conference (IACC) (2017): 701-705.
2. P.S. Mukambika, K Uma Rani, "Segmentation and Classification of MRI Brain Tumor," International Research Journal of Engineering and Technology (IRJET), Vol.4, Issue 7, 2017, pp. 683 – 688, ISSN: 2395-0056.
3. Shika gitte, Dr. BH Chandra Shekhar, " Brain tumor detection using CNN" International journal of creative research thoughts (IJCRT) vol 9, issue 6 June,2021, ISSN:2320-2882.
4. Anjaly Antony, Minla K.S, "Brain Tumor Detection From MRI Images using CNN" International Journal Of Creative Research Thoughts (IJCRT), Volume 10, Issued on 4th April 2022, Issn 2320-

2882.

1. Ashwini Patne, Laxmibai, Pooja DH, Joyti patil, "Brain Tumor Detection Using Convolutional Neural Network" International Research Journal of Innovative Research In Engineering and Multidisciplinary Physical Sciences (IJIRMPS), Volume 9, Issued on 4, 2021, ISSN: 2349-7300.
2. Pranav Shetty,Suraj Singh,Rasvi Jambhulkar,Kajal Sheth and Deepali Ujlambkar "Detection of brain tumorusing CNN and ML" International Journal of Applied Research Vol , Issue 2021,ISSN:308-313.
3. Dr.K.Kranthi Kumar, Mr.V.Raju, V.Abhinaya, P.Anusha, K.Akaansha, B.Soniya "Detection of Tumor Cells in Brain using CNN", I nternational Research Journal of Engineering and Technology(IRJET), Volume 7 , Issuedon 5th may 2020, ISSN:2395-0056.
4. Shreyash Gupta, Shreyas.S, Lalita VP "Segmentation approach for Brain Tumor Detection" International Journal Of Advance Research, Ideas And

Innovations In Technology (IJARIIT), Volume-7, Issued on 4, 2021,ISSN:2454- 132X.

1. Khalifa Shantta, Otman Basir, "Brain Tumor Detection and Segmentation: A Survey" International Journal Of Technology And Engineering (IRA), Volume-10, Issued on 4-2020, ISSN- 2455-4480
2. [https://www.kaggle.com/ahmedhamada0/](https://www.kaggle.com/ahmedhamada0/brain-tumor-detection) [brain-tumor-detection](https://www.kaggle.com/ahmedhamada0/brain-tumor-detection)