Brain Tumor Detection

Name : Arsh
Student ID : 22021190
Superviser : Dr. Manoj Diwakar

Acknowledgement

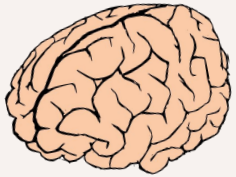
I would like to express my sincere gratitude to Dr. Manoj Diwakar for their invaluable guidance and unwavering support throughout the course of this project. Their expertise and feedback have been instrumental in shaping the direction of my work.

Table of contents

*

01

Problem Statement



One of the most crucial tasks of neurologists and radiologists is early brain tumor detection. However, manually detecting and segmenting brain tumors from Magnetic Resonance Imaging (MRI) scans is challenging, and prone to errors. That is why there is a need of an automated brain tumor detection system for early diagnosis of the disease.

*

02

Introduction

But Wait, What is Brain Tumor?

About the Disease

A brain tumor is an abnormal growth of cells in the brain that can be benign or malignant, disrupting normal brain function. It can cause symptoms such as headaches, seizures, and cognitive changes.

Why Detection is necessary?

Timely identification enables medical professionals to implement appropriate therapies, potentially preventing the tumor from reaching an advanced and more challenging stage.

Tech Stack Used

OpenCV

Image processing

TensorFlow

ML framework for deep learning

Keras Util

CNN Layers

Scikit-learn

Splitting Dataset

NumPy

Array Manipulations

PIL

Image operations

Pathlib

Working with File paths

Flask

Web Application

Tailwind

UI interface

Building the Model

1. Data Preprocessing

Initially I constructed some custom functions for handling the dataset, and then organise them into appropriate categories. Subsequent step involves using OpenCV for resizing and PIL for standardized input size of 64x64 pixels of the images in the dataset.

2. Converting Lists to Array

With the dataset structured, I converted the lists of images into NumPy arrays for compatibility with deep learning frameworks, the labels indicating 'tumor' & 'non-tumor' are also converted into np arrays. At this point, our data is ready for model building.

3. Dataset Splitting and Normalization

For evaluation we split our dataset into training and testing sets (80%-20 %) using scikit-learn. Prior to feeding the data into the neural network, we normalize the pixel values using the normalize function from Keras.it is a crucial preprocessing step that ensures consistent and effective learning during the training phase.

4. Building CNN

The model architecture consists of convolutional layers with ReLU activation functions, max-pooling layers, and fully connected layers. The model is compiled using categorical cross-entropy loss and the Adam optimizer. Training is executed over 10 epochs, and the resulting model is saved in a .h5 format.

Creating the Web Application

Web Server Setup

Using Flask which is (a light-weight python framework, I established the backbone of the application. Crucially ,I created routes for handling prediction and static page requests. Such as ‘model‘ , ‘model/predict‘ and ‘/ .’

Prediction Logic

Within the '/predict' route, I integrated the pre-trained deep learning model. This backend logic processes user inputs, prepares them for inference, and generates predictions regarding brain tumor presence. This modular approach ensures adaptability for future model updates.

Front - End Design

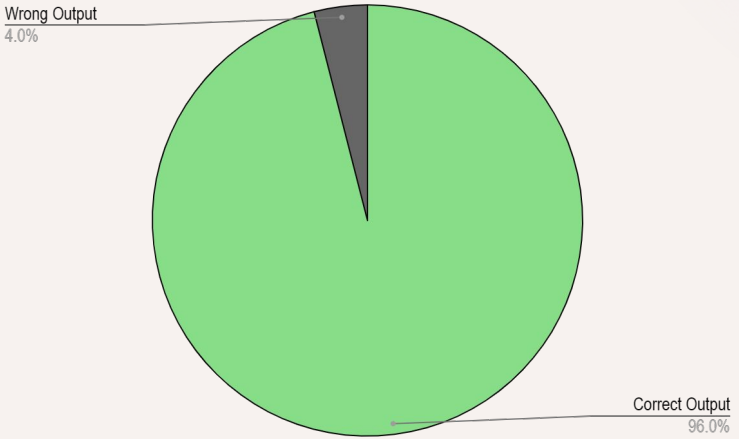
I used GenAI Models, such as idegram.ai, to generate clean graphics and cartoons, enhancing the app's visual appeal. Additionally, I utilised Tailwind CSS to craft a modern and clean user interface.

Results

Model Accuracy : 97%

Supports Low-End Devices

Integrated with Fully Working and easily Hostable Web Application.



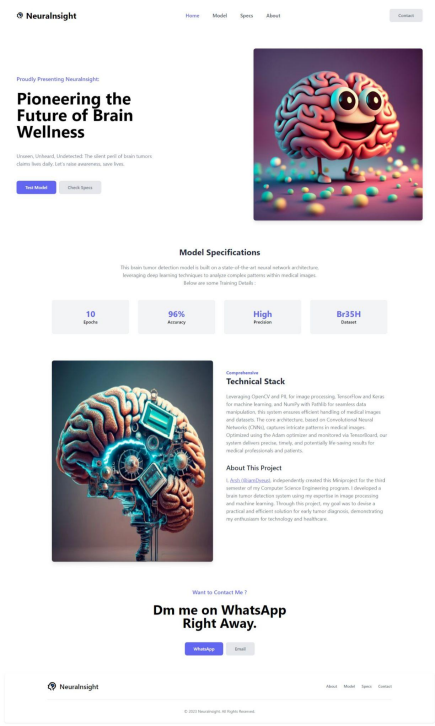
Summary

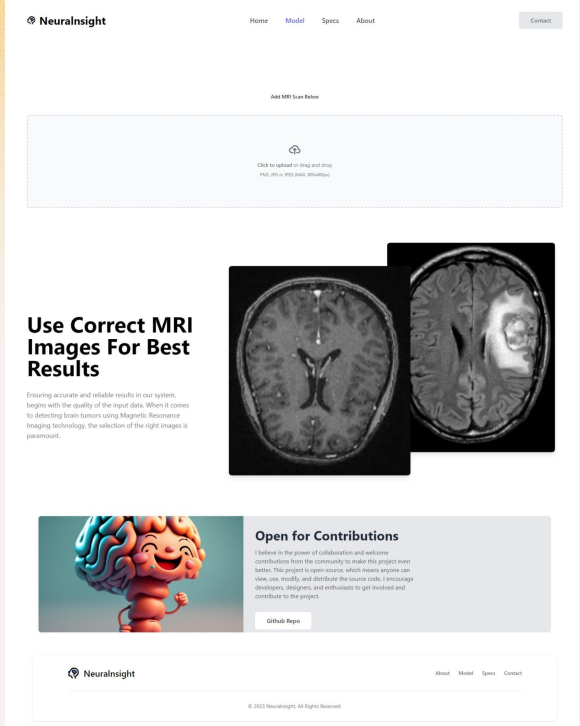
Project Highlights

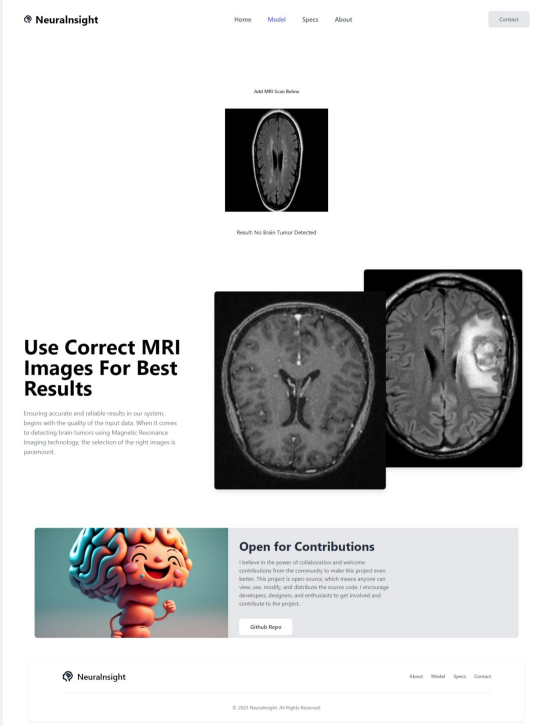
Mini Project - Summary

- Machine Learning Focus: The project centers on leveraging advanced machine learning techniques for early brain tumor detection.
- Deep Learning Model: At the heart of the initiative is a highly trained deep learning model, specifically implemented with convolutional neural networks (CNNs).
- TensorFlow and Keras: The model is constructed using TensorFlow and Keras, leading frameworks for developing and training deep learning models.
- Precision and Sensitivity: The model emphasizes precision and sensitivity in analyzing medical images, crucial for accurate brain tumor detection
- Distinguishing Features: Utilizing its deep learning capabilities, the model excels in distinguishing between brain tumor and non-tumor images.
- Critical Medical Contribution: The ultimate goal is to contribute significantly to early diagnosis in critical medical conditions, particularly those related to brain tumors.

Project Snippets







Thanks!

Do you have any questions?

removed for privacy

github.com/iamDyeus

Project Open Sourced on Github
github.com/iamDyeus/NeuralInsight