**Brain Tumor Detection Using Deep Learning**

**1. Introduction**

Brain tumor detection is a crucial medical task that helps in diagnosing and treating brain abnormalities. This project uses **Deep Learning** techniques, specifically a **Convolutional Neural Network (CNN)**, to classify MRI images as either having a Tumor or not. The model is developed using **TensorFlow** and deployed using **Stream lit** to provide an interactive user interface.

**2. Understanding Deep Learning**

Deep Learning is a subset of Machine Learning that uses neural networks to process data and recognize patterns. **CNNs (Convolutional Neural Networks)** are widely used for image classification tasks, making them ideal for detecting brain tumors in MRI scans.

**3. How CNN Works**

CNNs work by passing images through multiple layers:

* **Convolutional Layers**: Extract features from images.
* **Pooling Layers**: Reduce dimensionality and computational cost.
* **Fully Connected Layers**: Make final classification decisions.
* **Activation Functions**: Ensure non-linearity for better learning.

**4. Technologies Used**

This project is built using the following technologies:

* **TensorFlow/Keras**: For building and training the CNN model.
* **Flask**: Initially used for API-based deployment.
* **Stream lit**: Used for the final web-based deployment.
* **Python**: Core programming language.
* **HTML, CSS**: Used for designing the frontend in the initial phase.
* **GitHub**: Version control and project sharing.

**5. Project Requirements**

To run this project, you need:

* Python 3.7+
* Required libraries: Install using

pip install -r requirements.txt

* Hardware: A system with a decent GPU is recommended for training.

**6. Implementation Steps**

1. **Dataset Preparation**: Preprocess MRI images and split them into training and testing sets.
2. **Model Training**: Use TensorFlow/Keras to build and train a CNN model.
3. **Flask API Deployment**: Initially deployed using Flask to test backend predictions.
4. **Stream lit Integration**: Implemented Stream lit for a better UI experience.
5. **Final Deployment**: Optimized the model and UI for real-time predictions.

**7. Issues Faced & Solutions**

* **File upload not displaying** → Fixed by ensuring st.image(uploaded\_file) updates correctly.
* **CSS responsiveness issues** → Adjusted width of elements for better UI experience.
* **Flask API not returning correct predictions** → Ensured proper image preprocessing before model input.
* **Deployment challenges** → Shifted from Flask to Stream lit for a more interactive user interface.

**8. How to Run the Project**

1. Clone the GitHub repository:

git clone https://github.com/yourusername/brain-tumor-detection.git

1. Navigate to the project folder and install dependencies:
2. cd brain-tumor-detection

pip install -r requirements.txt

1. Run the Stream lit application:

Stream lit run app.py

1. Upload an MRI image to see the classification results.

**9. GitHub Repository & Contribution Guide**

This project is open-source, and contributions are welcome.

* **GitHub Repo:**
* **How to Contribute**: Fork the repo, make changes, and submit a pull request.

**10. Conclusion**

This project provides an effective way to detect brain tumors using Deep Learning. It combines the power of **CNNs** with **Stream lit** for an interactive and user-friendly deployment. Future improvements include expanding the dataset and refining the model for higher accuracy.

**Thank you for exploring this project! 🚀**