Brain MRI Tumor Detection AI

Manual

A ResNet18-based classifier with a Gradio web interface for detecting brain tumors from MRI images.

---

typst ITCS-6155 Final Project MIT License

Group 11

Diego Lopez*

Issam Alzouby†

Jake Pinos‡

Purva Jagtap§

Liliana Coste¶

Contents

*dlopez18@charlotte.edu

†ialzouby@charlotte.edu

‡jpinos@charlotte.edu

§pjagtap1@charlotte.edu

¶lcoste@charlotte.edu

Overview

This application detects the presence of a brain tumor from an MRI image using a ResNet18-based CNN and provides a simple Gradio web UI. Predictions include the label (Tumor / No Tumor) and a confidence score. This manual explains how to install, run, and use the system, with clear walkthroughs for common tasks.

Deployment & Installation

- Ensure Python 3.10+ on Windows, macOS, or Linux. CPU works; a CUDA-enabled GPU is optional for faster training/inference.
- Required libraries are listed in `requirements.txt` (torch, torchvision, gradio, opencv-python, nibabel, SimpleITK, scikit-learn, numpy, pandas, pillow, matplotlib, tqdm, streamlit).

```
~$ git clone https://github.com/purva115/brain_tumor_predictor.
git
~$ pip install -r requirements.txt
```

Model file:

- A pretrained model is included at `models/brain_tumor_model.pth`.
- Dataset (if training) should be under `data/BrainMRI/{yes,no}`.

Main Features

- ResNet18-based classifier for MRI tumor detection.
- Gradio web app (`src/app.py`) for point-and-click inference.
- CLI backend inference (`python -m src.inference.test_inference`).
- Utilities for basic anomaly check and report generation in `src/inference/`.

Primary Walkthrough: Predict from the Web App

Steps to run the interface and get a prediction.

```
~$ python -m src.app
```

1. Your browser opens Gradio “AI Brain Tumor Detection”.
2. Click “Upload Brain MRI” and select a JPG/PNG image.
3. Click Submit to see Prediction and Confidence.

Screenshots to add:

- Show launching the app and the browser page URL. Capture the whole window including title.
- Show the upload control with a sample MRI selected. Ensure the file chooser is visible.
- Show the output with Prediction and Confidence. Make text large enough to be readable.

Additional Walkthrough 1: Backend Inference (no UI)

Use the backend pipeline to verify predictions from the console.

```
~$ python -m src.inference.test_inference
```

1. When prompted, provide the path to an MRI image.
2. The console prints the predicted label and confidence.
3. Use Ctrl+C to exit when done.

Screenshots to add:

- Show the command prompt running the module. Include the full command line.
- Show sample input image path entered. Make sure the path is readable.
- Show the resulting printed prediction and confidence in the terminal.

Additional Walkthrough 2: Retrain the Model

Prerequisites:

- Place data as `data/BrainMRI/yes` and `data/BrainMRI/no`.

Train:

```
~$ python -m src.training.train_model
```

Outputs:

- A trained model checkpoint is saved (update `models/brain_tumor_model.pth` as needed).
- After training, re-run the app: `python -m src.app`.

Screenshots to add:

- Show the dataset folders in your file explorer. Ensure both yes/no classes are visible.
- Show the training script running with progress/epoch logs. Capture at least one epoch.
- Show the updated `models/brain_tumor_model.pth` timestamp in the folder.

Data & Formats

- Input: single MRI image (JPG/PNG). Gradio passes a file path to the pipeline.
- Expected classes: `yes` (tumor) and `no` (no tumor) for training.
- Typical preprocessing: resizing, normalization, optional channel conversion.

Troubleshooting

- Ensure requirements are installed without errors. Try upgrading pip.
- Check that `models/brain_tumor_model.pth` exists.
- Verify the image is a valid JPG/PNG and not corrupted.
- Check console logs from `src/app.py` for tracebacks.

- Training and inference work on CPU. For GPU, install a CUDA-compatible PyTorch build.

System Requirements

- OS: Windows 10/11, macOS 12+, or Linux.
- Python: 3.10 or newer.
- Hardware: 8 GB RAM minimum (training benefits from 16 GB+). Optional NVIDIA GPU with CUDA.

How to Get Help

- Read `README.md` for quick commands and repository structure.
- Review scripts in `src/` for reference usage (e.g., `src/app.py`, `src/inference/inference_pipeline.py`).
- If issues persist, include error messages and steps to reproduce when asking for support.