Iteration 2 Report: Brain Tumor Detection

Team Members: Sri Divija Enturi, Reha Jambavadekar, Jessica Pham

Course: Essentials of Data Science

1. Project Kickoff

1.1 Specific Goals and Expected Outcomes

The goal of this project is to design and implement an end-to-end system for **brain tumor detection and management** using MRI scans. It integrates machine learning, database design, and web development.

Goals:

- Detect and classify brain tumors from MRI scans using a pre-trained CNN model.
- Store MRI scans, patient metadata, and classification results in a database.
- Implement anonymization and role-based access for secure data handling.
- Automate ETL pipelines for image ingestion, preprocessing, and metadata extraction.
- Generate SQL reports to analyze tumor types, patient demographics, and classification accuracy.
- Build a web interface to upload scans, visualize results, and query insights.

Expected outcomes: A functioning prototype integrating model inference, database storage, and data visualization with secure and efficient workflows.

1.2 Project Scope

The scope focuses on three main modules:

- **Tumor Detection (Reha):** Train or adapt a pre-trained CNN for tumor classification and automate batch image processing.
- **Database Management (Divija):** Design schema, manage data storage, ensure anonymization and governance, and build ETL pipelines.
- **Integration & Web Interface (Jessica):** Develop the front-end in Flask, integrate model and database, and prepare the final demo.

1.3 Key Deliverables by Phase

Phase	Deliverables
Phase 1	Dataset selection and environment setup
Phase 2	Database schema design and ER diagram
Phase 3	Pre-trained CNN integration for classification
Phase 4	Automated ETL pipeline implementation
Phase 5	Web interface integration and testing
Phase 6	Analytical SQL reports and final presentation

1.4 Milestones and Deadlines

Milestone	Target Date	Responsible Member
Dataset finalization	Week 3	Reha, Divija
Schema design	Week 4	Divija
Model integration	Week 5	Reha
ETL workflow	Week 6	Divija
Web interface setup	Week 7	Jessica
SQL reporting	Week 8	Divija, Jessica
Final integration and testing	Week 9	Team

1.5 Team Capabilities and Gaps

The team possesses skills in ML, SQL, and web integration. A small gap exists in cloud deployment and visualization, which will be addressed using online tutorials and reference materials.

1.6 Dataset Availability

Dataset: Brain MRI Classification Dataset (GitHub)
It contains de-identified MRI images of glioma, meningioma, pituitary, and no-tumor cases. The dataset is publicly available and open-licensed.

2. Team Discussions

2.1 Core Skills and Contributions

Team Member	Core Skills	Contributions
Reha Jambavadekar	Python, TensorFlow, Image Processing	Tumor detection and classification, automated batch scripts
Jessica Pham	Web Development, Flask, HTML/CSS	Web interface, system integration, demo preparation
Sri Divija Enturi	SQL, Data Engineering, Security	Database schema, ETL pipelines, data governance

2.2 Missing Skills and Challenges

Challenges include model deployment and managing large MRI files. These will be mitigated through research and peer consultation.

2.3 Tools and Technologies

Familiar Tools: Python, SQL, GitHub, VS Code, Jupyter Notebook
To Learn: TensorFlow deployment, SQLAlchemy integration, Flask web APIs

2.4 Programming Languages and Platforms

- **Languages:** Python, SQL, HTML, CSS
- **Frameworks:** TensorFlow, Flask, Pandas, OpenCV
- **Database:** PostgreSQL or SQLite
- **Version Control:** GitHub

3. Skills and Tools Assessment

3.1 External Resources

Guidance will be taken from Kaggle forums, Stack Overflow, and open GitHub repositories. Faculty feedback will guide database normalization and security practices.

3.2 Tools, Frameworks, and Libraries

Category	Tools / Libraries
Model Development	TensorFlow, Keras, OpenCV
Data Management	PostgreSQL, SQLAlchemy
ETL Pipeline	Python, Pandas
Visualization	Matplotlib, Plotly
Web Interface	Flask, HTML/CSS
Version Control	Git and GitHub

3.3 Ensuring Team Proficiency

Short internal sessions will be conducted to share learnings and resolve technical issues. Each member will test and validate their module using a small sample dataset before integration.

3.4 Task Assignment and Clarity

Tasks have been divided based on individual strengths, ensuring clear accountability and efficient workflow.

4. Initial Setup

4.1 Development Environment

- Python 3.10+, TensorFlow, and OpenCV installed locally.
- PostgreSQL database configured for MRI metadata.
- Flask web environment for interface and integration.
- VS Code and Jupyter Notebook for development.

4.2 Version Control Configuration

A shared GitHub repository is set up with access for all members. Each task is committed through branches, merged after testing, and documented in the README file.