Iteration 4 Report: Brain Tumor Detection Project Progress and Plans

Sri Divija Enturi, Reha Jambavadekar, Jessica Pham
Essentials of Data Science

November 2025

Project Links

GitHub Repository Link: `https://github.com/jp74ham/Brain_Tumor_Detection`
Dataset Link: `https://github.com/SartajBhuvaji/Brain-Tumor-Classification-DataSet`

1 Dataset Description

4 The team is utilizing the **Brain Tumor Classification Dataset** for this project. This is a publicly available, open-licensed dataset containing de-identified MRI images classified into four distinct, multi-class categories: **glioma, meningioma, pituitary**, and **no-tumor** cases. The dataset is structured as a collection of image files alongside associated metadata.

Relevance and Suitability: This dataset is uniquely suitable because it directly addresses the project's core machine learning objective: to **detect and classify brain tumors from MRI scans** using deep learning. Its pre-categorized structure is ideal for transfer learning with a Convolutional Neural Network (CNN). Furthermore, the variety of tumor types (glioma, meningioma, pituitary) provides a realistic and challenging classification task, enabling the generation of meaningful analytical reports and demonstrating the system's ability to handle secure, de-identified patient data.

2 Tools and Methodologies

The project integrates a comprehensive and justified technology stack across the three main modules.

For **Tumor Detection (Modeling)**, the team will use **TensorFlow** and **Keras** for transfer learning. The primary focus is on two pre-trained CNN models: Xception and VGG-16.

- Xception is the preferred choice due to its **state-of-the-art performance in multi-class medical image classification, leveraging depthwise separable convolutions** to efficiently extract fine-grained, subtle features crucial for tumor detection. This choice maximizes potential diagnostic accuracy.
- VGG-16 will be implemented as a **robust benchmark** model for comparison. It is chosen for its simple, uniform architecture and proven generalization capability, serving as a reliable alternative that minimizes the risk of overfitting on the specialized MRI dataset.

OpenCV will be used for essential image preprocessing (e.g., resizing, normalization).

For **Data Engineering**, the system relies on **PostgreSQL** or **SQLite** for database management, integrated via SQLAlchemy. This ensures secure and structured storage of MRI metadata, classification results, and patient anonymization. The automated **ETL pipeline** will be built using **Python** and the Pandas library to handle image ingestion, preprocessing orchestration, and metadata extraction, ensuring data quality and governance.

The **Web Interface** will be built using the lightweight and flexible Flask framework, along with HTML/CSS, enabling the integration of the model inference API and database queries for a functioning prototype that allows users to upload scans and view results.

3 Preliminary Timeline

The following is a detailed, weekly timeline covering key tasks and deliverables for the remainder of the semester:

Phase	**Key Task / Milestone**	**Target Week**	**Deliverable / Outcome**
Data Prep	Dataset finalization and cleaning	Week 3	Finalized, normalized dataset ready for ingestion.
Data Prep / Backend	**Database Schema Design**	Week 4	Completed ER Diagram and initial SQL for table creation.
Model Dev	Pre-trained CNN Integration and Training (Xception/VGG-16)	Week 4	Initial trained model with benchmarked performance metrics.

Phase	Key Task / Milestone	Target Week	Deliverable / Outcome
ETL Pipeline	Automated ETL Pipeline Implementation	Week 5	Python scripts to ingest, preprocess, and load data/metadata into the database.
Integration / UI	Web Interface Setup and Model API Integration	Week 6	Functional Flask application with working image upload and model inference display.
Evaluation	Analytical SQL Reports Generation	Week 6	Finalized SQL queries to report on tumor statistics and model accuracy.
Finalization	Final Integration, Testing, and Optimization	Week 7	System-wide stress tests, bug fixes, and performance tuning.
Reporting	Final Project Submission and Presentation Preparation	Week 7	Completed final report, presentation slides, and demo ready.

---

4 Team Member Contributions

The team maintains clearly defined roles based on individual core competencies, ensuring efficient workflow and accountability.

- **Sri Divija Enturi** (Role: Data Engineering, Backend): Divija's expertise lies in **SQL, Data Engineering, and Security**. Her primary contributions include the **Database schema design**, building the robust ETL pipelines, and implementing **data governance** mechanisms for anonymization.
- **Reha Jambavadekar** (Role: Modeling, Scripting): Reha specializes in **Python, TensorFlow, and Image Processing**. Her responsibility is

focused on the core **Tumor detection and classification** module, including adapting the pre-trained CNNs (Xception/VGG-16) and developing **automated batch scripts** for image processing.

- **Jessica Pham** (Role: Integration, Frontend): Jessica's expertise covers **Web Development, Flask, and HTML/CSS**. Her role centers on developing the user-facing **Web interface**, performing system **integration** between the model and database, and preparing the final demonstration.

Collaboration and Evolution: The team utilizes a shared **GitHub repository** for version control. Collaboration is maintained through regular check-ins and short internal sessions for knowledge transfer. Roles will organically evolve during the final integration and testing phases (Week 9), transitioning into a shared "Team" responsibility for documentation and bug resolution.

5 Progress and Next Steps

5.1 Progress Made to Date

Progress has been substantial, establishing a strong foundation across all system modules:

- **Project Foundation:** All initial goals, scope, and team roles were finalized. The Development Environment (Python, TensorFlow, PostgreSQL) has been successfully configured.
- **Database Management:** The crucial Database schema design and ER diagram has been completed.

  Database Schema and ER Diagram Overview:

  The database has been designed to efficiently manage MRI metadata, tumor classification results, and user roles while maintaining strict data anonymization and security standards. The schema consists of five primary entities — **Patients**, **MRI_Scans**, **Tumor_Classification**, **Model_Performance**, and **User_Roles** — with one-to-many and one-to-one relationships ensuring consistency and referential integrity.

Database Schema Overview:

The designed database consists of five key entities. Each entity is structured to ensure data integrity, anonymization, and efficient linkage between MRI scans, classification results, and user roles.

- **Patients**
  - **Primary Key:** patient_id
  - **Attributes:** patient_name, age, gender, date_of_scan, hospital_name, anonymized_id
  - Stores anonymized patient details and scan metadata.
- **MRI_Scans**
  - **Primary Key:** scan_id **Foreign Key:** patient_id
  - **Attributes:** image_path, scan_type, uploaded_by, upload_timestamp
  - Holds image file paths and metadata for each MRI scan.
- **Tumor_Classification**
  - **Primary Key:** classification_id **Foreign Key:** scan_id
  - **Attributes:** tumor_type (glioma, meningioma, pituitary, no_tumor), model_version, confidence_score, classified_on
  - Stores classification outcomes generated by the CNN model for each MRI scan.
- **Model_Performance**
  - **Primary Key:** model_id
  - **Attributes:** model_name, accuracy, precision, recall, f1_score, trained_on
  - Tracks model evaluation metrics and version history.
- **User_Roles**
  - **Primary Key:** user_id
  - **Attributes:** username, role (Admin, Radiologist, Researcher), access_level, last_login
  - Defines system access privileges and maintains user authentication logs.

Entity Relationships:

- One **Patient** can have multiple **MRI_Scans**.
- Each **MRI_Scan** is linked to one **Tumor_Classification**.
- Multiple **Tumor_Classifications** can reference a single **Model_Performance** record.
- A **User_Role** can be associated with multiple MRI uploads or classification entries.

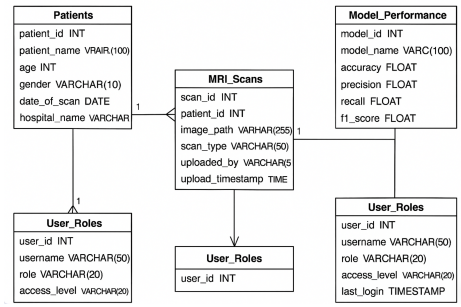

Figure 1: ER Diagram

- **Tumor Detection:** A suitable pre-trained CNN model set (Xception and VGG-16) has been identified. Initial research and testing on image preprocessing requirements using OpenCV have begun, preparing the module for the Week 5 training phase.
- **Integration & Web Interface:** The Flask web environment has been initialized, and basic routing is configured. The team has started learning SQLAlchemy integration and Flask web APIs to ensure smooth future module connection.

5.2 Ongoing Challenges and Next Steps

Ongoing Challenges: The team is actively mitigating a potential skill gap in advanced cloud deployment and optimization techniques necessary for production-level performance. Additionally, strategies for efficient handling of large MRI files within the ETL pipeline require further refinement.

Next Steps (Upcoming Week - Week 5): The immediate focus is on completing the core machine learning task and advancing the data pipeline:

1. Tumor Detection (Reha): Complete the Pre-trained CNN integration and initial training (Xception/VGG-16) and validate their inference capabilities (Week 5 milestone). 2. ETL Pipeline (Divija): Begin implementation of the core Automated ETL pipeline (Week 6 preparation), focusing on the ingestion of image files, preprocessing, and metadata extraction into the defined schema. 3. Integration (Jessica): Develop the initial secure role-based access control logic and build the first functional prototype of the scan upload interface.

The project plan remains aligned with the established timeline, with the current focus ensuring successful model training and robust data flow implementation.