Tumor Tracking Web Application – Workshop Project Proposal

Project Overview

This project aims to streamline and improve clinical workflows by providing an automated system for analyzing tumor progression across multiple MRI scans. Currently, medical personnel must manually compare large volumes of imaging data, which is time-consuming, prone to error, and difficult given the complexity and variability of MRI studies. The system will generate an automated tumor-tracking report that summarizes detected changes across time, helping clinicians make more informed and precise decisions.

Problem Definition

Doctors and radiologists are required to interpret MRI scans from various time points to determine tumor trajectory. This process involves significant cognitive load and can lead to overlooked trends or delayed treatment decisions. A streamlined system for tumor progression analysis would reduce that burden and offer standardized output that supports clinical decision-making.

Objectives

- Build a full-stack web application that accepts MRI scans and generates automated tumor trajectory reports.
- Provide segmentation, multi-scan comparison, and structured PDF output.
- Support clinician workflows with patient history, user authentication, and secure storage.
- Implement privacy controls to ensure all medical data remains secure and compliant.

Key Features

Core Features

- **Tumor Segmentation:** Using nnU-Net to identify tumors within MRI scans.
- **Multi-Scan Tumor Tracking:** Automatic comparison across multiple MRI studies.
- **Automated PDF Reporting:** Summaries of tumor volume, morphological changes, and progression.
- **Integration of RAG-based patient data summaries**
- **Authentication System:** Doctors access only their assigned patient data.

Nice-to-Have Features

- Enhanced UI and dashboard components.
- Annotation tools for clinicians.
- 3D MRI visualization.
- Privacy system for regulations handling.

Target Audience

- **Primary users:** Doctors, especially oncologists and radiologists.
- **Secondary users:** Radiation therapy personnel and medical imaging staff.

Example User Stories

- **As an Oncologist**, I want to upload a patient's current MRI scan and have it automatically aligned with their scan from 3 months ago, so that I can visually verify if the tumor size has increased or decreased without manually scrolling through slices.
- **As a Radiologist**, I want the system to calculate the exact volumetric change (in cubic centimeters) of the tumor between two time points, so that I can report objective "Response to Treatment" metrics (e.g., RECIST criteria) rather than subjective estimates.
- **As a Clinician**, I want to see a summarized textual report that integrates the MRI findings with the patient's recent general health data (age, symptoms, comorbidities), so that I have a holistic view of the patient's status before making a treatment decision.
- **As a Medical Administrator**, I want to ensure that only assigned doctors can view specific patient records via role-based authentication, so that we remain compliant with hospital privacy regulations.

Existing Approaches and Our Differentiation

Existing Products

Companies such as Aidoc, Viz.ai, Arterys (Tempus), RadAI, and Qure.ai provide AI-driven radiology tools but focus on triage or single-scan interpretation rather than longitudinal tumor tracking.

Research tools like BraTS models, 3D Slicer, and academic segmentation papers offer strong segmentation but lack robust, automated multi-scan comparison and practical clinical integration.

Our Competitive Advantage

- **Longitudinal Analysis:** Automated tumor tracking across multiple MRI scans.
- **Standardized Reporting:** PDF summaries designed for clinical workflows.

- **Hospital-Focused Architecture:** Built directly from pain points reported by Ichilov Hospital personnel.
- **Secure Deployment:** Privacy-first design to meet regulatory definitions.

High-Level Solution Architecture

This system utilizes a **Dual-Stream Processing Architecture** to handle the distinct nature of image data versus textual patient data.

High-Level Input/Output Flow: `[Input: MRI Scans + Clinical Notes] --> [Processing Core] --> [Output: PDF Report + Interactive Dashboard]`

1. The Image Stream (Tumor Detection):

- **Input:** Longitudinal MRI sequences (DICOM/NIfTI). Fetching data from local data sources by client (data don't leave the hospital)
- **Processing:** The system uses **Deep Learning (for example: nnU-Net)** for segmentation. It identifies the tumor in 3D space and performs geometric comparisons between Timepoint A and Timepoint B.
- **Customization:** The model adapts to the specific anatomical features of the patient's scan to generate a precise mask.

2. The Text Stream (Patient Context via RAG):

- **Input:** Unstructured text (Medical history, pathology reports, physician notes) and structured data (Age, Gender).
- **Technique (RAG vs. Fine-Tuning):** We utilize **Retrieval-Augmented Generation (RAG)** rather than fine-tuning an LLM.
  - **Why RAG?** Patient data is dynamic and highly specific. Fine-tuning a model on every patient is computationally expensive and risks "hallucination." RAG allows us to retrieve exact facts (e.g., "Patient had surgery on date X") from the specific patient's documents to prompt a frozen LLM, ensuring accuracy and data isolation.
- **Processing:** Clinical notes are chunked and stored in a vector database. When generating the report, the system queries this database to summarize the patient's general health status alongside the imaging results.

Data Sources and Training Strategy

To build a robust solution, we will access data from the following sources:

- **Ichilov tumor dataset:** Data set provided by Ichilov for scientific research.
- **nnU-Net (Segmentation Core):** Self-adapting Framework for U-Net-Based Medical Image Segmentation (2018).
- **BraTS Dataset:** Updated every year for the BraTS challenge.

System Architecture and Technologies

Backend

- **Languages & Frameworks:** Python, FastAPI
- **ML Frameworks:** PyTorch, nnU-Net. (consider other technologies such as MONAI, and TorchIO)
- **Imaging Libraries:** optional technologies are Pydicom, SimpleITK, and NiBabel
- **Task Processing:** Celery with Redis
- **Database:** PostgreSQL
- **Reporting:** PDF generation service

Frontend

- React.js

Infrastructure

- **Cloud:** AWS (EC2, S3, RDS, VPC, ECR, CloudWatch, KMS)
- **Containerization:** Docker, Docker Compose
- **CI/CD:** GitHub Actions
- **Security:** Private VPC, encryption in transit and at rest, audit logs (read and write), 2FA, role-based control access.

Architecture Style

- Asynchronous Service-Oriented Architecture: inference service, API gateway, frontend, worker service, database, storage.

Constraints and Dependencies

- MRI training data is limited and expensive.
- GPU resources are required for training and inference.
- Strict privacy constraints due to handling medical data (PHI).
- All patient data must remain fully secure and private.

Challenges and Risks

Technical Challenges

- Matching and tracking the same tumor across different MRI scans.
- Ensuring robustness despite variations in imaging protocols.
- Efficiently processing large 3D volumetric datasets.

Privacy Risks

- Handling PHI requires strict compliance with security standards.
- Preventing data and meta-data leakage is critical (e.g Re-Identification, PHI leakage in DICOM files).
- Broken access control could cause a Horizontal Privilege Escalation.
- Data Manipulation - An attacker could alter an MRI scan (injecting invisible "noise") to trick the model into missing a tumor (“Adversarial Attack”) or modify the final report numbers to mislead the doctor.

Design Challenges

- Presenting complex medical data in a simple, interpretable interface.

Algorithmic Challenges

- Accurate segmentation across timepoints.
- Image registration across heterogeneous scans.

Expected Impact

- Improved consistency in tumor monitoring.
- Reduced manual workload for radiologists.
- Early detection of meaningful tumor changes.

Future Scope

- Integration with full patient medical records.
- Statistical dashboards for multi-patient analytics.
- Multi-modal data (CT, pathology).
- AI-driven treatment recommendation support.
- Potential commercialization or open-source release.

Relevant Paper References

To ensure our methodology is grounded in state-of-the-art research, we are referencing the following key papers:

1. **nnU-Net (Segmentation Core):** Self-adapting Framework for U-Net-Based Medical Image Segmentation (2018).
2. **The 2024 Brain Tumor Segmentation (BraTS) Challenge:** Glioma Segmentation on Post-treatment MRI
3. **Automated longitudinal treatment response assessment of brain tumors:** A systematic review (2025)
4. **Retrieval-Augmented Generation (RAG) in Healthcare:** A Comprehensive Review (2025)
5. **Artificial Intelligence-Enabled Medical Devices** (website)