

# Milestone 1 - Gp3

LAHMAMSI MOHAMED AMINE  
LIU SHIZHENG  
ROUE ROBINSON

## Integrated Medical Imaging Platform (IMIP) – Specifications

### 1. Purpose

The **Integrated Medical Imaging Platform (IMIP)** is a modular, research-oriented software ecosystem designed to support experimentation, education, and innovation in the field of **AI-assisted medical imaging**.

It provides a secure and ethical environment for the **development, evaluation, and demonstration of medical imaging models**, including computer vision, segmentation, classification, and anomaly detection systems.

While **not intended for clinical use**, the platform bridges the gap between **academic deep learning models** and **medical data management workflows**, simulating real-world medical software architecture for teaching and prototyping purposes.

### 2. Platform Overview

The IMIP is built as a **multi-module system** combining:

- **AI Model Services** (e.g., tumor detection, organ segmentation)
- **Image Management & Annotation Tools**
- **User-facing Research Interface**
- **Compliance and Ethics Monitoring Layer**

The platform supports **MRI, CT, and X-ray** modalities and integrates **deep learning models** through a unified backend API.

Each model is encapsulated as a self-contained service with clear data, visualization, and compliance interfaces.

### 3. Intended Users

- **Medical imaging researchers** testing model pipelines or visualization tools
- **Biomedical engineering students** studying AI integration into healthcare workflows
- **AI developers** experimenting with model deployment and interpretability
- **Professors and lecturers** conducting laboratory demonstrations in medical AI courses

User capabilities:

- Familiarity with imaging data (DICOM, PNG, JPEG).
- Basic understanding of machine learning workflows.
- No direct medical or diagnostic responsibility.

### 4. System Architecture (High-Level Overview)

Layer	Description	Key Technologies
<b>Frontend Medical Dashboard</b>	Web-based interface for uploading scans, viewing predictions, and accessing model results.	React / TypeScript / Tailwind
<b>Backend Orchestrator</b>	Manages image ingestion, validation, inference routing, and results storage.	FastAPI / Flask / PostgreSQL
<b>Model Services Layer</b>	Hosts independent AI models (e.g., tumor detection, segmentation).	PyTorch / TensorFlow
<b>Visualization Engine</b>	Generates overlays, heatmaps, and confidence visualizations.	OpenCV / Matplotlib
<b>Data Management Layer</b>	Handles metadata, anonymization, and storage.	DICOM toolkit / SQLAlchemy
<b>Ethics &amp; Compliance Layer</b>	Displays disclaimers, enforces anonymization, and logs activity.	Policy framework
<b>Monitoring &amp; Analytics</b>	Collects performance, latency, and research metrics.	Prometheus / Grafana

## 5. Module – Brain Tumor Detection Subsystem

### Purpose

This subsystem demonstrates the integration of a **Convolutional Neural Network (CNN)** for **brain tumor classification from MRI images**.

It represents a proof-of-concept for AI-assisted radiology workflows, emphasizing model interpretability and ethical design.

### Software System Overview

Users upload MRI images through the main IMIP interface.

The image is forwarded to the **Brain Tumor Detection Service**, which preprocesses the data, runs model inference, and returns:

- A binary classification (“Tumor” / “No Tumor”)
- A confidence score
- An optional visual explanation (Grad-CAM or heatmap)

### DL Role in the Module

Component	Description	Role
User Interface	Scan upload & results visualization	Interaction
Backend Controller	Orchestrates inference pipeline	Routing
Preprocessing Module	Image resizing & normalization	Data preparation
DL Inference Engine	Keras CNN	Prediction
Visualization Layer	Heatmap overlay generation	Explainability
Ethics Layer	Non-diagnostic disclaimer enforcement	Compliance

# Software System Requirements

## 1. User Requirements

**UR1:** The user must be able to access the platform through a standard web browser (Chrome, Edge, Firefox).

**UR2:** The user must be able to upload anonymized medical images (MRI, CT, or X-Ray) via the interface.

**UR3:** The user must be able to select the desired AI analysis module (e.g., Brain Tumor Detection, Lung Nodule Segmentation).

**UR4:** The user must receive model predictions and visual explanations within 3m (end-to-end, including network latency).

**UR5:** The user must not be required to install any software locally.

**UR6:** The user must be presented with an ethical disclaimer before performing any inference.

## 2. System Requirements

**SR1:** The platform must host all DL model services (e.g., Brain Tumor Detection, Pneumonia Detection) on a backend server environment.

**SR2:** Each model service must expose a standardized `/predict` endpoint that receives image data and returns predictions in JSON format.

**SR3:** The platform backend must route requests to the appropriate model service based on user module selection.

**SR4:** The backend must process uploaded images, run model inference, and return results within 2 seconds (backend-only time budget).

**SR5:** The browser must only handle visualization, user interaction, and display of responses — no inference computation.

**SR6:** The system must delete uploaded medical files immediately after the response is returned.

### 3. Functional Requirements

**FR1:** The system must allow users to upload medical images through drag-and-drop or file selection.

**FR2:** The backend must validate and anonymize uploaded images before inference.

**FR3:** The preprocessing module must resize, normalize, and format input images according to the selected model's requirements.

**FR4:** The backend must perform inference and return structured results including:

- predicted label(s) (e.g., *Tumor*, *No Tumor*)
- model confidence score
- optional visual explanation (e.g., heatmap overlay)

**FR5:** The platform must display results and visualizations within 1 second after the backend response is received.

**FR6:** The system must present an error message if inference fails or total latency exceeds 15 seconds.

### 4. External Interfaces

#### EI1: REST API Gateway

- **Endpoints:** `/predict`, `/heatmap`, `/meta`
- **Method:** POST (for inference) / GET (for metadata)
- **Input:** Multipart-form encoded file (DICOM, PNG, JPEG)
- **Output:** JSON (e.g., `{"label": "Tumor", "confidence": 0.92}`)

#### EI2: Front-End Medical Dashboard

- **Technology:** React / HTML / CSS / JavaScript
- **Behavior:**
  1. User uploads scan and selects analysis module
  2. System displays loading indicator

### 3. Results and visual explanations rendered upon response

#### EI3: Backend-to-Model Interface

- **Technology:** PyTorch / TensorFlow
- **Behavior:**
  - Model loaded at server startup
  - Requests handled asynchronously through inference manager

#### EI4: Ethics and Compliance Interface

- **Purpose:** Display disclaimers and consent forms prior to inference
- **Technology:** Web-based modal or popup acknowledgment

## 5. Constraints

**C1:** All AI inference must execute on backend servers (Python environment: Flask, FastAPI, or equivalent).

**C2:** Client-side inference via TensorFlow.js or Web ML is strictly prohibited.

**C3:** Uploaded files must be automatically deleted after inference to ensure data privacy.

**C4:** All data used for testing or demonstration must be anonymized and non-diagnostic.

**C5:** The system must operate on student-grade hardware or free-tier cloud environments (e.g., Render, Railway, HuggingFace Spaces).

**C6:** No online model retraining, fine-tuning, or parameter modification must be accessible to users.

## 6. Quality Attributes (Non-Functional Requirements)

Attribute	Requirement
<b>Performance</b>	Model inference $\leq$ 1 minute, full round-trip $\leq$ 2 minutes
<b>Usability</b>	Prediction accessible in $\leq$ 3 clicks
<b>Security</b>	Data deletion immediately after session
<b>Portability</b>	Backend compatible with Linux & Windows
<b>Transparency</b>	All models must provide confidence and visual explanation
<b>Ethical Compliance</b>	Mandatory disclaimers and research-only flags

# High-Level Description of the Model

The objective of this project is a binary classification to analyze an MRI scan and unambiguously determine if a brain tumor is present ("Yes") or absent ("No").

The chosen model is a custom sequential Convolutional Neural Network (CNN), (<https://www.kaggle.com/code/bobweng/brain-tumor>) built with Keras. This architecture was intentionally chosen over other complex alternatives because it is specifically designed to address this precise binary question.

## Alternative Models Considered and Rejected

### 1. The "Multi-Class" Approach

- **Description:** These models are designed for a different, more complex problem: multi-class classification. They do not just say "yes" or "no" but attempt to distinguish *between multiple types* of tumors (e.g., glioma, meningioma, pituitary) in addition to the "healthy" class.
- **Justification for Rejection:** Using a multi-class model for a binary problem is a fundamental design mismatch.
  - **Inefficiency:** It forces the model to expend resources learning to make distinctions (for example, "glioma vs. meningioma") that are unnecessary for our objective.
  - **Unnecessary Complexity:** The use of a softmax activation function (typical of these models) is inappropriate where a simple sigmoid is sufficient. Our model is focused: it only answers the question asked.

### 2. The "Transfer Learning" Approach

- **Description:** This method involves importing a very large model (millions of parameters) pre-trained on millions of real-world images (ImageNet: cats, dogs, cars, etc.).
- **Justification for Rejection:** This is an unsuitable tool for this job for three reasons:
  - **Domain Mismatch:** The features a model learns from ImageNet (colors, fur, leaf textures) have little relevance to analyzing the anatomical structures in an MRI. Our model, trained *from scratch*, will learn features *specific* to MRI scans.
  - **High Risk of Overfitting:** Using such a large model on a relatively small medical dataset (like our 4900 images) is problematic. The model is so large it is prone to "memorizing" the training images instead of learning to generalize. Our lighter model is better suited to our data's scale.
  - **Lack of Interpretability:** EfficientNet acts as a "black box." In the medical field, it is important to be able to interpret *why* a model makes a decision. Our simpler architecture is more transparent.

## **Justification for the Chosen Model (Keras Sequential Binary - Brain-Tumor Detection (CNN))**

The architecture is designed for a single purpose. The final Dense(1, activation='sigmoid') layer is a standard and correct mathematical method for binary classification. The model is neither overly simple (it has 3 convolutional levels to learn complex features) nor overly complex.

By not using transfer learning, we require the model to learn a visual "vocabulary" that is highly relevant to MRIs. It is sized to perform well on our dataset without simply memorizing it.

This is a proven architecture. It offers a good balance between capability (thanks to its 3 convolution blocks) and interpretability, which is important for validating a diagnostic-aid tool.

## **Model Card - Brain Tumor**

### **Model Details**

- **Model Name:** Brain-Tumor Detection (CNN)
- **Author:** Bob Weng (Kaggle user)
- **Model Date:** May 2025
- **Model Type:** Convolutional Neural Network (CNN). This is a sequential architecture built with Keras (TensorFlow), using Convolution (Conv2D), Pooling (MaxPooling2D), Relu activation, Flatten, and Dense layers.
- **Application Domain:** Medical imaging, specifically for brain tumor detection (binary or multi-class classification depending on the images), BUT not adapted for real use on patients.
- **License:** This Notebook has been released under the Apache 2.0 open source license.

### **Intended Use**

- **Recommended Use:**
  - "Pre-screening tool or aid for detecting brain tumors from medical images."
  - Serve as an educational example for Deep Learning applied to the medical field.
- **Main Intended Users:** Students, data scientists, and health researchers exploring image classification techniques.
- **Non-Recommended Use:**
  - This model must **NOT** be used for actual medical diagnosis, clinical decision-making, or patient counseling.
  - It must **not** be used as the sole or unique clinical diagnostic tool.
  - It does **not** replace a medical examination, a radiologist's analysis, or qualified human intervention.

- It must **not** be used for organs other than the brain without rigorous adaptation and validation.

## Factors

- **Demographic and Phenotypic Factors:** Patient age and sex are relevant factors that could influence performance. As the current dataset does not appear to include this information, it is impossible to evaluate disaggregated performance across these groups.
- **Instrumentation (Equipment):** The performance of a computer vision model is often sensitive to the hardware used. Factors include:
  - The scanner type.
  - The magnetic field strength.
  - The acquisition sequences.
- **Environmental Conditions:** Image quality, the presence of artifacts (motion blur, noise), and different viewing angles are key factors. The model was trained with data augmentation (rotation, zoom) to attempt to mitigate these variations, but an evaluation on "difficult cases" (e.g., noisy images) has not been provided.

## Metrics

- **Training Metrics:** The model uses the Loss and Accuracy on the training and validation sets. These are used to monitor model convergence and diagnose overfitting.
- **Recommended Analysis:** Given the medical context and the identified ethical risks, accuracy and Loss alone is insufficient. The evaluation must be based on the Confusion Matrix and the following metrics:
  - **False Negative Rate (FN):** The most important metric. It measures the proportion of actual tumors that the model wrongly classified as "no tumor". The goal is to minimize this rate.
  - **False Positive Rate (FP):** Measures the proportion of healthy cases wrongly classified as "tumor".
  - **ROC Curve:** An ROC (Receiver Operating Characteristic) curve is necessary to visualize the trade-off between the False Positive Rate and Sensitivity (Recall) at different decision thresholds.

## Evaluation data

- Evaluation was performed on a partition (15%) of the public "Brain Tumor MRI Dataset". This partition was set aside and not used during training or validation.
- This dataset was chosen because it provides real MRI images of brains with actual tumor cases. It was also chosen because it provides multiple images from different possible angles.
- We do not have information regarding the demographic diversity (age, sex) of the patients in the dataset, which may have an impact.
- Adjustments are applied to the images (rotation, zoom, flipping, resizing, pixel normalization) to improve image quality for the model.

## Training Data

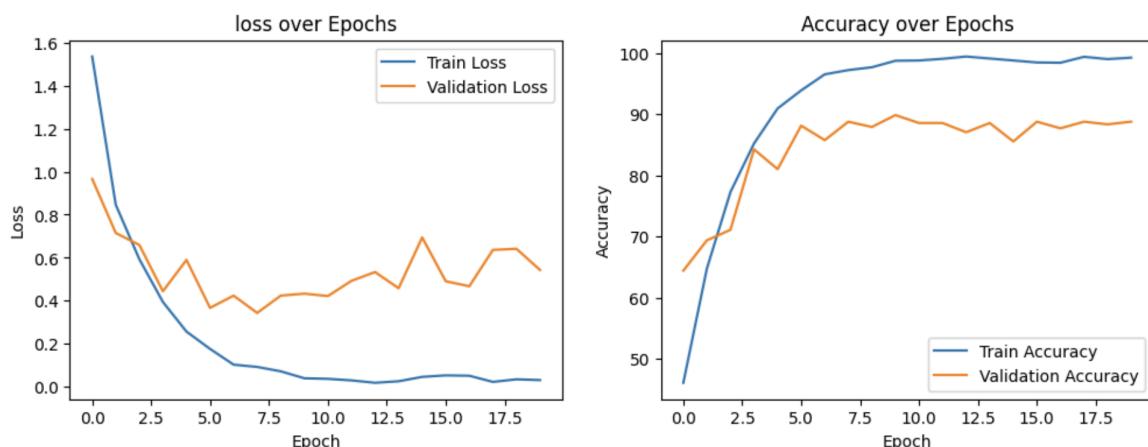
- Just like the evaluation data, the training data comes from the "Brain Tumor MRI Dataset".
- This set contains 4900 MRI images for training.
- The data is distributed into two classes: approximately 1600 "tumor" images and 3200 "no tumor" images.
- The data is divided into training (train\_set) and validation (val\_set) sets for model training and tuning.
- Adjustments are applied to the images (rotation, zoom, flipping, resizing, pixel normalization) to improve image quality for the model.

## Ethical Considerations

- The model is trained on medical data (MRI images) which is inherently sensitive. Although the public dataset is anonymized, the use of such data requires special attention to privacy.
- Applications of this type of model directly affect human health and life, which is why this model must not be used for actual medical diagnosis or replace consultation with a doctor.
- **Risks and Harms:**
  - **False Negatives:** The most critical risk is classifying a tumor image as "no tumor," which could delay a vital diagnosis.
  - **False Positives:** Classifying "no tumor" as a tumor could lead to unnecessary anxiety and superfluous invasive exams.

## Quantitative Analyses

We can note that the accuracy is very good. However, there is slight overfitting, as the accuracy during training is much better than during validation.



## Caveats and Recommendations

- **Non-Clinical Model:** Due to the observed overfitting and the lack of testing on external data (from other hospitals or scanners), this model is a proof of concept. It must under no circumstances be used as a standalone diagnostic tool.
- **Impact of Imbalance:** The training dataset is imbalanced (3200 "no tumor" vs 1600 "tumor"). A 90% accuracy can therefore be misleading if the model is less performant on the "tumor" class, which is in the minority but is more critical.

## Dataset Card: Brain Tumor Detection Dataset (Binary Classification)

### 1. Dataset Description

- **Homepage:** <https://www.kaggle.com/datasets/akrashnoor/brain-tumor>

This dataset is curated to support the development and evaluation of machine learning and deep learning models for binary brain tumor detection. It includes high-quality brain MRI images labeled into two primary categories: Tumor (which consolidates glioma, meningioma, and pituitary tumors) and No Tumor.

#### 1.1. Dataset Summary

The dataset contains grayscale brain MRI images organized for a binary classification task (Tumor vs. No Tumor). While the original source provides multi-class labels, this card focuses on its direct application for binary detection. All images are preprocessed and standardized (e.g., uniform size, grayscale format, normalized intensity) to ensure consistency, serving as an excellent resource for research and education in AI-driven medical imaging.

#### 1.2. Languages

The dataset is image-based and does not contain textual language data.

### 2. Dataset Structure

#### 2.1. Data Fields

Each data point consists of an image and its corresponding binary label.

- **Image** (Grayscale MRI) – Input
- **Label (Tumor or No Tumor)** – Output

#### 2.2. Data Splits

There are 4200 pictures that represent 'Tumor' and 2000 that represent 'No tumor'.

70% Training 15% Validation 15%Testing

### 3. Dataset Creation

#### 3.1. Curation Rationale

The dataset was curated to provide a standardized resource for researchers to experiment with, benchmark, and compare models for brain MRI image classification.

#### 3.2. Source Data

Images were collected from public medical imaging datasets and anonymized clinical sources. All data are preprocessed to a uniform size, grayscale format, and have normalized intensity.

#### 3.3. Annotations

The data provides classification labels. No segmentation masks or bounding box annotations are included.

#### 3.4. Personal and Sensitive Information

The dataset contains no personally identifiable or sensitive patient information. All images are anonymized and intended solely for research and educational purposes.

### 4. Considerations for Using the Data

- **Potential Bias:** MRI data sources may have inherent variations in acquisition conditions (scanner type, patient demographics, image quality), which could introduce bias into the model.
- **Clinical Use:** This dataset should **not** be used for primary clinical decision-making or diagnosis without extensive, independent validation.
- **Class Imbalance:** A key consideration for this binary task is the potential for significant class imbalance. The "Tumor" class is an aggregation of three distinct tumor types, while the "No Tumor" class is a single category. This likely results in the "Tumor" class being over-represented.

### 5. Additional Information

- **Dataset Curators:** Public contributors and medical image curators
- **Licensing:** Research and educational use only (non-commercial)