INTELLIGENT RADIOLOGIST ASSISTANT

An optimized web application designed specifically for classifying tears in knee MRI images.

HIGH LEVEL DESIGN

Domain: Healthcare

Technologies: Deep Computer Vision

Creator: Arup Sankar Roy

Date: 20-05-2024

Email:asrl20221078.cse.uit@gmail.com

DOCUMENT CONTROL

Date issued	Version	Description	Author
February 29,2024	1.1	First Draft	Arup Sankar Roy
April 04,2024	1.2	Architecture building and Experimenting.	Arup Sankar Roy
May 07, 2024	1.3	Document Writing and other stuff	Arup Sankar Roy

Contents

Abstract

The "Intelligent Radiologist Assistant" web application is designed to assist radiologists and healthcare professionals in the accurate and efficient diagnosis of knee injuries through the analysis of MRI images. This application leverages advanced deep learning models, specifically AlexNet and ResNet18, to classify MRI images from different planes (axial, coronal, and sagittal). By providing detailed analysis and visualizations, the application helps in identifying the presence and type of tears in the knee, which can be critical for timely and effective treatment planning. The user-friendly interface, coupled with the ability to generate comprehensive reports, makes it an invaluable tool for enhancing diagnostic accuracy, reducing the workload on radiologists, and ultimately improving patient outcomes.

Introduction

What is High-Level Design Document?

The goal of this HLD or a high-level design document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- **Present all of design aspects and define them in detail**
- **Describe all user interfaces being implemented**
- **Describe the hardware and software interfaces**
- **Describe the performance requirements**
- **Include design features and architecture of the project**
- **List and describe the non-functional attributes such as security, reliability, maintainability, portability, reusability, application compatibility. resource utilization, serviceability**

Scope

The HLD documentation presents the structure of the system, such as database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly technical terms which should be understandable to the administrators of the system.

General Description

Definitions

Term	Description
IRA	Intelligent Radiologist Assistant
Database	Collection of the Information
IDE	Integrated Development Environment
UI	User Interface
CMD	Command prompt

Product Description

The IRA is a web application that automates the interpretation of knee MRI scans. It classifies MRI images, provides detailed visual reports, and prioritizes high-risk patients. This tool enhances diagnostic accuracy and efficiency with an easy-to-use interface for clinicians.

Problem Statement

Magnetic Resonance Imaging (MRI) of the knee is the preferred method for diagnosing knee injuries. However, interpreting knee MRI scans is time-intensive and subject to diagnostic errors and variability. An automated system for interpreting knee MRIs could prioritize high-risk patients and assist clinicians in making accurate diagnoses. Deep learning methods, which can automatically learn layers of features, are well-suited for modeling the complex relationships between medical images and their interpretations.

Proposed solution

The "Intelligent Radiologist Assistant" web application provides an automated solution for interpreting knee MRI scans using deep learning models, specifically AlexNet and ResNet18. It reduces diagnostic time and errors by automatically classifying tears from MRI image planes like axial, coronal, and sagittal. The application offers detailed reports with visualizations prioritizing high-risk patients. Its user-friendly interface ensures accessibility for clinicians, enhancing diagnostic accuracy and efficiency, and ultimately improving patient care.

Further improvements

- **Model Refinement:** Continuously refine and update the deep learning models to Improve accuracy and robustness.
- **User Interface**: Enhance the user interface for improved navigation and user experience.
- **Multi-Language Support**: Expand language support to cater to a wider audience.
- **Mobile Application**: Develop a mobile app version for convenient access.
- **Integration**: Integrate with hospital systems for seamless data exchange.
- **Advanced Reporting**: Include more sophisticated analysis and export options in reports.
- **Continuous Learning**: Implement a feedback loop for continuous model improvement.
- **Security**: Strengthen data security measures to protect patient information.
- **Collaboration Tools**: Add features for collaborative case review and sharing insights.

Data requirements

To use the "Intelligent Radiologist Assistant" web application, the following data is required:

MRI Image Stacks:

- **Axial Plane:** A .npy file containing a stack of images captured from the axial view of the knee.
- **Coronal Plane:** A .npy file containing a stack of images captured from the coronal view of the knee.
- **Sagittal Plane:** A .npy file containing a stack of images captured from the sagittal view of the knee.

Blank CSV File:

- A CSV file to be populated with the analysis results.

File Format

- **Accepted Formats:** .npy files for MRI image stacks.
- **CSV File**: Must be blank and ready to receive analysis data.

Upload Guidelines

- Ensure that all MRI stacks are clear and free of artifacts.
- The MRI scans should be of consistent quality and orientation for accurate processing.
- The .npy files should be properly formatted and contain the entire image stack for each plane.

These data requirements ensure that the IRA can effectively analyze and generate accurate reports for knee MRI scans.

Tools used

Python programming language and different frameworks & database such as NumPy, Pandas, Matplotlib,Seaborn,Plotly, Flask, PyTorch,HTML,CSS,Java Script and a few other libraries & MySQL as a database were used to build the whole project.

- **Flask:** Used for creating web pages and handling HTTP requests.
- **HTML/CSS and JavaScript:** Used for designing and styling and client-side interactivity and dynamic content. for the web interface.
- **PyTorch:** Utilized for building and training machine learning models.
- **NumPy:** Used for numerical computations and data manipulation.
- **Pandas:** Employed for data analysis and manipulation.
- **Matplotlib,Seaborn & Plotly:** Used for creating static, interactive, and animated visualizations in Python.
- **MySQL:** Employed as the database management system for storing and retrieving data.

Hardware Requirements

- **Processor: Any modern multi-core CPU.**
- **Memory (RAM): Minimum 4GB RAM.**
- **Storage: Adequate storage space, preferably SSD.**
- **GPU (Optional): Accelerates deep learning tasks.**
- **Network Connection: Stable internet connection.**

Constraints

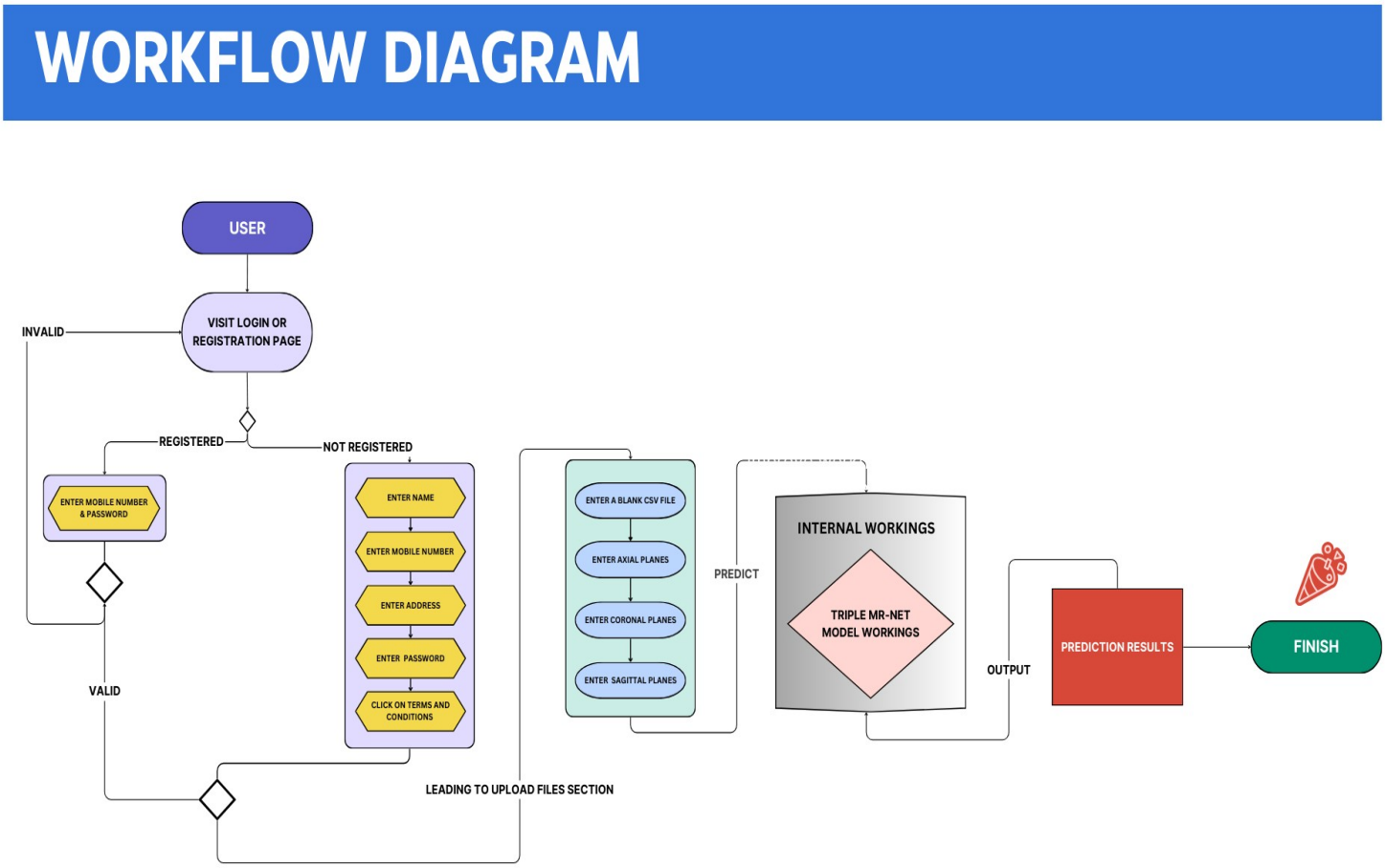
- **Limited computational resources may impact the speed and scalability of model training and inference.**
- **Data privacy regulations must be adhered to when handling patient medical data.**
- **Availability of MRI datasets for model training and validation may be limited.**

Assumptions

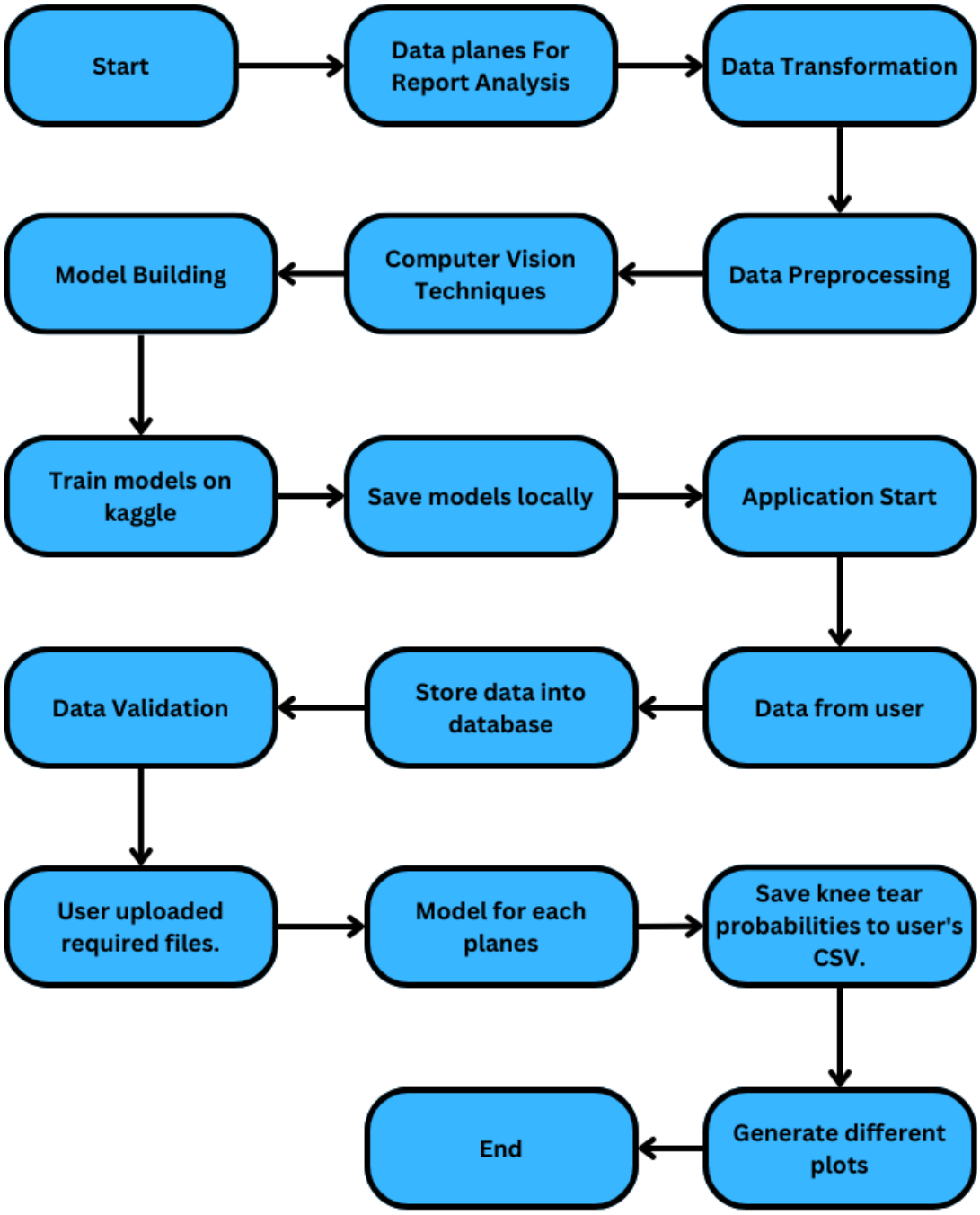
- **Users have basic knowledge of web browsing and interacting with web applications.**
- **The web application will be accessed primarily by healthcare professionals and radiologists.**
- **The accuracy of the diagnostic predictions generated by the models is assumed to be reliable for clinical decision-making.**

Design Details

- **Diagram for How Your Web App Works:**



- **Diagram for Overall Process from Raw Data to Prediction:**



Error Handling

Errors should be encountered, an explanation will be displayed in CMD and an error will be defined as anything that falls outside the normal intended usage.

Performance

The performance of the web application is critical for efficient MRI image analysis. It relies on fast model training and inference, scalable infrastructure, quick response times, optimal resource utilization, and robust error handling. Ensuring efficient utilization of computational resources, minimizing latency, and implementing effective error handling mechanisms are essential for delivering a smooth user experience and accurate diagnostic results.

Reusability

The code written and the components used should have the ability to be reused with no problems.

Application Compatibility

The IRA web application is designed to be compatible across various platforms and devices. It is accessible via modern web browsers, ensuring usability on both desktop and mobile devices. The backend, built with Flask, integrates seamlessly with various hospital information systems and databases, including MySQL.Compatibility with existing healthcare infrastructure is prioritized to ensure smooth deployment and integration into clinical workflows.

KPIs (Key Performance Indicators)

- **Accuracy: Achieve over 90% classification accuracy.**
- **Inference Time: Maintain under 5 seconds per scan.**
- **Report Generation: Generate reports within 1 minute of upload.**
- **Processing Efficiency: Ensure efficient data handling with minimal errors.**

Conclusion

The IRA web application successfully leverages advanced deep learning techniques to automate the interpretation of knee MRI scans. By providing accurate and timely diagnostic assistance, it enhances the efficiency and accuracy of radiologists, ultimately improving patient care. The application's robust design ensures compatibility with various systems, while its user-friendly interface facilitates easy adoption by healthcare professionals. With continuous improvements and scalability, the application is well-positioned to become an invaluable tool in the medical imaging field, reducing diagnostic errors and streamlining the workflow in clinical settings.