Bilkent University
Department of Computer Engineering

Senior Design Project

T2407

Lesion Lens

Final Report

22001293, Berk Özkan, berk.ozkan@ug.bilkent.edu.tr
22101734, Mert Fidan, mert.fidan@ug.bilkent.edu.tr
21802354, Arda Baktır, arda.baktir@ug.bilkent.edu.tr
22101882, Emir Kerem Şahin, emirkerem@ug.bilkent.edu.tr
22101343, Muhammad Shayan Usman, shayan.usman@ug.bilkent.edu.tr

Supervisor: Uğur Güdükbay
Course Instructors: Mert Bıçakçı, Atakan Erdem

02.05.2025

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfilment of the requirements of the Senior Design Project course CS491/2.

Contents

Final Report

Lesion Lens

1. Introduction

1.1. Purpose of the System

LesionLens is a web-based application designed to assist doctors, neurologists, and radiologists in analyzing brain MRI scans, with a particular focus on detecting and tracking multiple sclerosis (MS) lesions. The system provides an interactive MRI viewer, allowing users to explore 2D slice views (axial, sagittal, and coronal) as well as a 3D reconstruction of MRI scans. Additionally, it supports historical tracking of lesion progression to aid in diagnosis and treatment planning.

The primary objective of LesionLens is to offer an efficient, accurate, and intuitive platform for MRI analysis while ensuring seamless integration with DICOM standards [1]. To achieve this, the system is built using a modular architecture, combining frontend visualization capabilities with a backend for secure data management and authentication.

1.2. Design Goals

LesionLens is designed to be **efficient, accurate,** and **user-friendly**, ensuring a seamless and easy to use experience for medical professionals analyzing brain MRI scans.

LesionLens should be **modular** and **scalable**:

- To enable seamless **integration** between the MRI viewer frontend and backend data management.
- To allow **future expansions**, including AI-based report generation and advanced analytics.
- To ensure **maintainability** and ease of updates as medical imaging technologies evolve.

LesionLens should provide **high performance** and **reliability**:

- To support **smooth 2D and 3D rendering** of large DICOM files using **VTK.js**.
- To optimize system efficiency, minimizing **loading times** for MRI scans.
- To handle **large datasets** without performance degradation.

LesionLens should ensure **security** and **data protection**:

- To implement **secure authentication** and **role-based access control** for patient confidentiality.
- To comply with **medical data standards** for handling sensitive patient information.
- To prevent **unauthorized access** to MRI scans and historical patient data.

LesionLens should **enhance diagnostic capabilities**:

- To provide **historical trend analysis** for tracking lesion progression over time.
- To integrate **AI-driven lesion detection** for more precise and automated analysis [2].
- To improve **decision-making efficiency** for medical professionals.

2. Requirements Details

2.1 Functional Requirements

2.1.1 MRI Data Upload

The system will enable doctors to upload multiple MRI collections for processing and analysis, providing the capability to manage extensive datasets efficiently. All uploaded MRI data will be securely stored in the database to ensure privacy and facilitate further analysis, complying with all relevant data protection regulations.

2.1.2 Viewing 2D MRI Slices

Doctors will be able to view individual MRI slices, presented as 2D images, through an intuitive web application interface. This functionality will allow for detailed examination and analysis of specific areas within the MRI scans.

2.1.3 Lesion Detection and Analysis

The system will incorporate an advanced machine learning (ML) model designed to detect and outline lesion-like objects in brain and spine MRI data. The model will accurately count the number of lesion-like objects identified and possess the capability to differentiate between closely clustered lesions, ensuring precise and reliable results.

2.1.4 3D Mapping and Lesion Measurement

The system will include a sophisticated tracking algorithm that generates 3D mappings of lesion-like objects by processing multi-angle MRI data. This functionality will enable comprehensive visualization and analysis. Furthermore, the system will calculate the area and volume of identified lesions, providing quantitative metrics to support clinical decision-making.

2.1.5 Monitoring Lesion Progression

To facilitate monitoring over time, the tracking algorithm will compare current and previous MRI scans, identifying potential lesion growth or the emergence of new developments. This feature will assist doctors in tracking disease progression and evaluating treatment efficacy.

2.1.6 3D Reconstruction Visualization

Doctors will have access to a 3D reconstruction of lesion-like objects through the web application, allowing for enhanced visualization and a better understanding of complex cases.

2.1.7 Sharing and Collaboration

The system will enable doctors to share their analysis, including both 2D images and 3D visualizations, with other authorized users for collaborative review and consultation.

2.1.8 Exporting Analysis Results

Analysis results can be exported in a standardized format, ensuring compatibility with offline use or integration into other tools and systems. This feature will support a seamless workflow for doctors who may require results outside of the web application environment.

2.1.9 Administrative Tools

For administrative needs, the system will provide a dedicated set of tools for admin users. These tools will include functionalities for user management, system monitoring, and the management of MRI collections. This ensures the system remains secure, well-maintained, and capable of adapting to the evolving requirements of its users.

2.2 Non-functional Requirements

2.2.1 Usability

Even though our application will run on the web, we have a very specific user base and specific desktops to run on—namely neurology doctors at Hacettepe Hospital (HH) and their designated computers. As such, our web application should be intuitive and easily adoptable for our users. To achieve these goals, our usability requirements are as follows: The application must allow multiple users to use the same account simultaneously. It should have an intuitive user interface tailored to the target audience (doctors at HH). Any login process should take at most four steps with valid credentials. All functionalities of the application must be accessible and easily visible from the main page. Furthermore, undo/back buttons should always be placed in the top left corner of the screen, while "exit," "escape," or "X" buttons must be positioned in the top right corner of the corresponding window. The application must handle all network requests asynchronously, enabling users to interact with the frontend continuously. Additionally, all in-progress network requests must be clearly visible to users.

2.2.2 Reliability

Reliability is the most critical requirement since doctors will use our application to make decisions and manage sensitive personal data. To ensure reliability, lesion-like detection must achieve an accuracy of 94% for both brain and spine analyses. It is imperative that no doctor can access another patient’s data without proper authorization. Furthermore, user data must never be vulnerable in the database or during its transfer to and from the database. All user data must be encrypted to prevent unauthorized access. Any DICOM or NII data associated with the user must never contain identifiable user information or any link back to the user. These files should only be accessible via a key and path stored in an encrypted user database.

2.2.3 Performance

While performance and usability are distinct requirements, they are closely interconnected in this application. Doctors at HH typically spend around 15 minutes analyzing a patient’s MRI scans. Our application must demonstrate significantly greater speed or accuracy than manual analysis to ensure adoption. Thus, performance requirements include the following: The latency of the server when using the machine learning analysis and tracking feature for lesions must be less than 120 seconds. The 3D model viewer should run smoothly on modern PCs used at HH, maintaining a frame rate of at least 30 fps. Loading an MRI scan into the 3D viewer should take no more than 10 seconds. The application must also support multiple user accounts utilizing the machine learning service simultaneously without performance degradation. Additionally, any part of the user interface must load within five seconds on internet speeds of 9 Mbps or higher.

2.2.4 Scalability

Our system must be designed to handle increasing demands in terms of data, users, and computational requirements without compromising performance or reliability. It should adapt seamlessly to the growing needs of HH, ensuring continued usability and efficiency over time. Scalability requirements include the efficient storage and retrieval of large MRI datasets, supporting up to 100,000 scans. The system must handle at least 100 concurrent users without noticeable performance degradation. It should process high-resolution MRI data and larger datasets without impacting machine learning model accuracy or response time. The infrastructure must dynamically scale resources to handle peak traffic demands. Furthermore, the system should enable easy integration of new technologies, improved machine learning models, or additional imaging modalities. It must also ensure interoperability with external databases or systems for research or regulatory purposes.

2.2.5 Security & Privacy

Our application relies on sensitive data, such as patient information, to deliver accurate lesion detection and ensure reliable results. This data includes training and testing datasets for the machine learning model, making it critical to implement stringent security measures. To balance functionality and privacy, we will adhere to established privacy regulations, including the Kişisel Verilerin Korunması Kanunu (KVKK) and the General Data Protection Regulation (GDPR), while obtaining necessary approvals and user consent as required. Security measures include ensuring that all datasets used for training and testing the machine learning model comply with KVKK and GDPR guidelines. Patient data provided by HH for model training and testing must receive approval from the Ethical Board of HH. All passwords will be securely hashed using the SHA-256 algorithm to prevent unauthorized access. Session tokens will be issued using secure protocols and will remain valid for a maximum of eight hours. Additionally, refresh tokens will be provided to allow users to regenerate session tokens as needed.

2.2.5 Pseudo Requirements

- React will be used to develop the frontend.
- WebGL will be used to handle the 2D and 3D rendering.
- GoLang will be used as the language of the backend.
- GoLang will be used as the development environment for the backend.
- PostgreSQL will be used for database management.
- Orthanc will be used for DICOM file management.
- Swagger will be used to test endpoints in the backend.
- Python, Tensorflow and PyTorch will be used to develop the model.
- Google Colab will be used to train and test the model.

3. Final Architecture and Design Details

3.1. Overview

Lesion Lens is a web-based medical imaging application that aids in analyzing neurological lesions in brain and spine MRI scans. Developed to address increasing neurology workloads, it accelerates manual review while maintaining diagnostic accuracy.

The system uses machine learning to detect potential lesions, identifying abnormalities linked to conditions like multiple sclerosis or brain tumors. It processes MRI data to locate lesions, analyze characteristics, and provide initial measurements without manual input.

Lesion tracking monitors changes over time, calculating volumetric measurements and documenting disease progression. It maintains lesion records for quantitative comparisons between scans, helping clinicians assess treatment effectiveness.

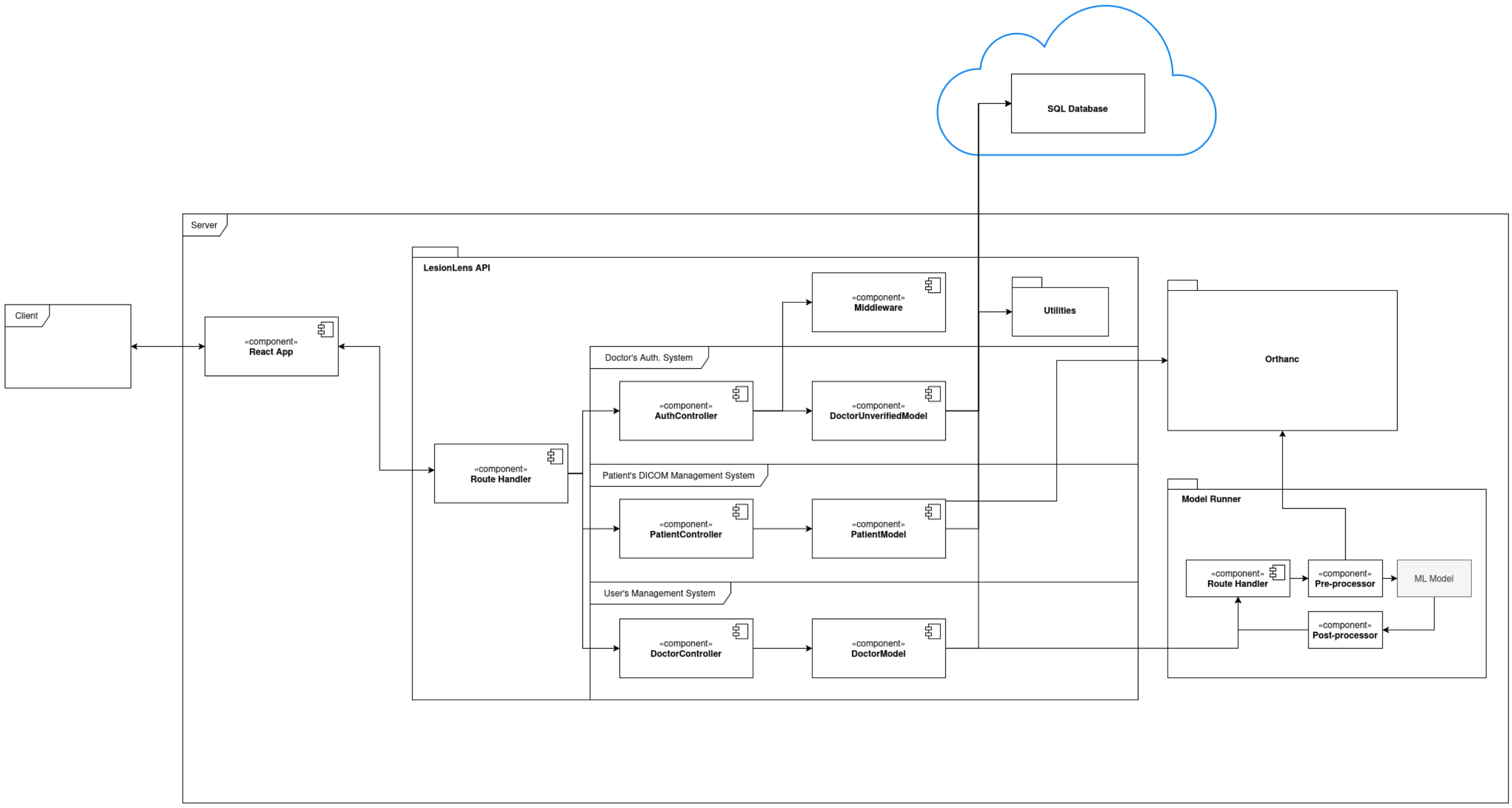
The web-based platform processes DICOM files using a GoLang backend, cloud storage, and TensorFlow/PyTorch models. It supports large datasets and multiple users, integrating with clinical workflows.

Designed for longitudinal lesion tracking and volumetric analysis, it combines detection, temporal comparison, and 3D visualization. It supports monitoring progressive neurological conditions, providing quantitative data for clinical decisions.

Development prioritized efficiency and clarity, incorporating feedback from medical professionals to ensure practical clinical use. The system continues to evolve based on user experience and technology advancements.

Additional features include data export and standardized reporting, facilitating lesion tracking and progression analysis for patient records and healthcare collaboration. Research applications enable anonymized data aggregation for studying disease progression and treatment outcomes.

3.2. Subsystem Decomposition



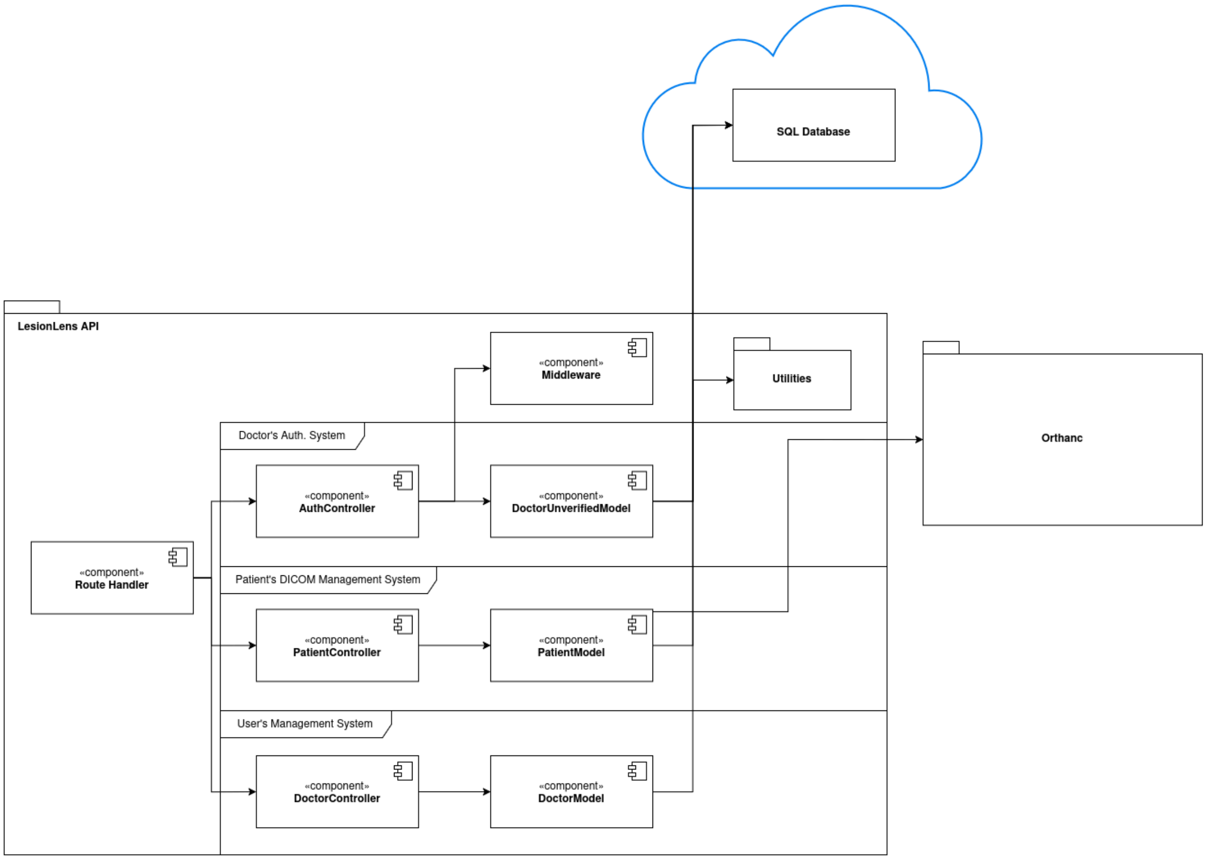
To build a robust and efficient medical imaging system, we have structured our platform into four fundamental components. Each component plays a critical role in ensuring seamless user interaction, data management, and machine learning inference. These components include:

- React Application (Frontend): Handles the user interface and user interactions.
- Golang API (LesionLens API, Backend): Serves as the core logic layer, facilitating communication between the frontend, databases, and machine learning model.
- Machine Learning Model (FastAPI, Python): Processes DICOM images to detect lesions.
- Orthanc DICOM Database and Cloud SQL Database: Stores and manages medical images and related metadata.

This modular architecture enhances maintainability, scalability, and efficiency by clearly separating concerns and responsibilities within the system.

3.3. Subsystem Services

3.3.1 API (LesionLens Backend)

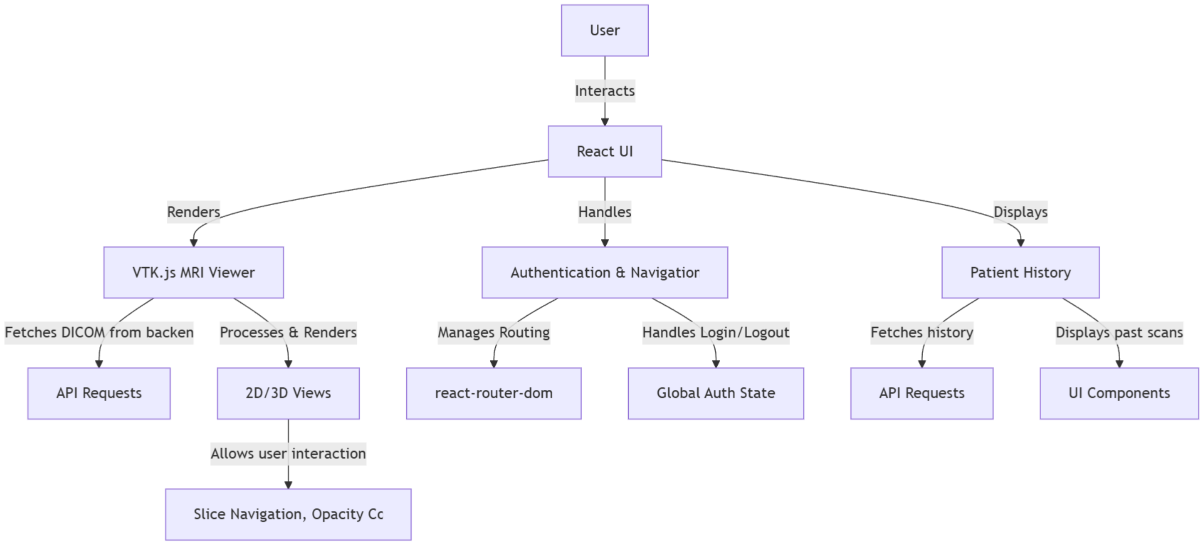


The API, built using Golang, serves as the central processing unit of the application. It manages user authentication, database interactions, and DICOM file operations.

- Controllers:
  - Patient Controller: Handles patient-related data, linking them to doctors and their medical records.
  - Doctor Controller: Manages doctor accounts, their access to patient records, and verification procedures.
  - Auth Controller: Implements JWT-based authentication for secure user access.
- Models:
  - Patient Model: Represents patient information and links to DICOM images in the database.
  - Doctor Model: Stores doctor details and associated patient records.
  - Unverified Doctor Model: Handles new doctor registrations and email verification workflows.
- Middleware & Utilities:
  - Middleware functions enforce authentication and authorization protocols.
  - Utility functions support database queries, JWT management, and DICOM file handling.

- Database Connectivity:
    - The API integrates with both a local Orthanc DICOM database for medical image storage and a Cloud SQL database to maintain structured patient and doctor information.

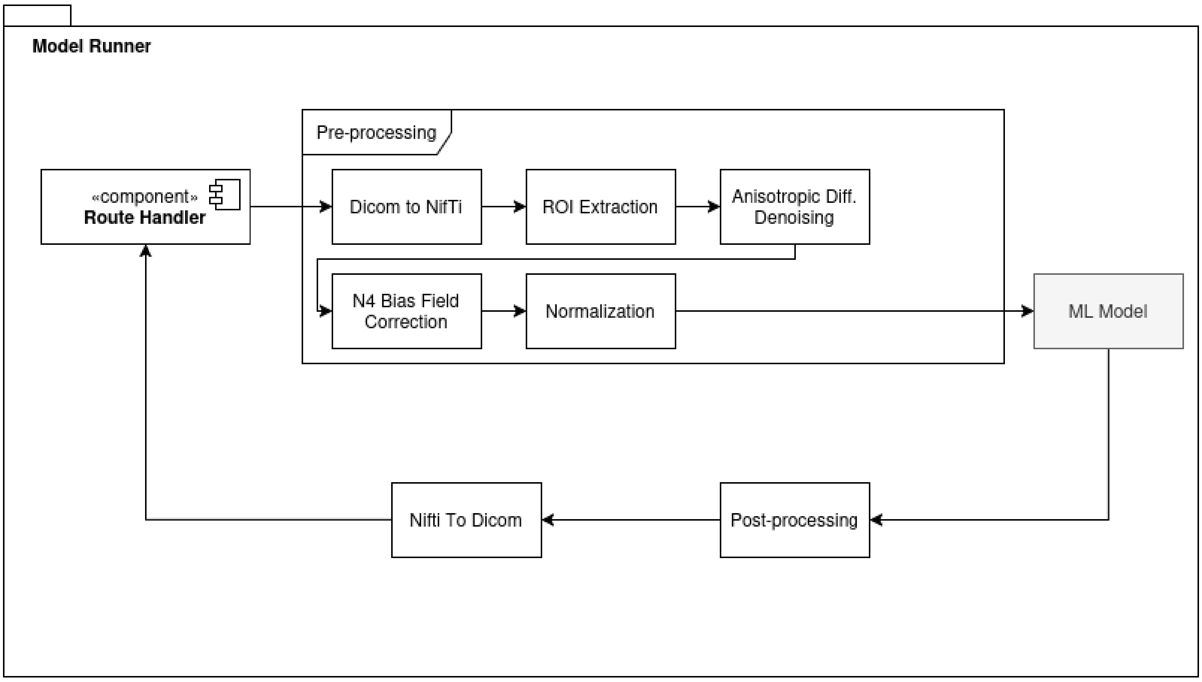
3.3.2 React Application (Frontend)



The frontend of LesionLens is a React-based web application built with TypeScript, designed for intuitive interaction with medical imaging data. It consists of three main components:

- **MRI Viewer**: Utilizes VTK.js to render 2D slice views (axial, sagittal, and coronal) and a 3D volume reconstruction of MRI scans. It supports interactive controls such as slice navigation, opacity adjustments, and linked views. The viewer fetches DICOM files from the backend for processing and visualization.
- **Authentication & User Management**: Implements role-based authentication with *react-router-dom*, allowing users to log in, register, and manage their accounts. It maintains session persistence and controls access based on user roles.
- **Dashboard & Patient History**: Provides an organized interface for navigating patient records, initiating DICOM processing, and tracking lesion development over time. Doctors can analyze historical MRI scans to monitor lesion progression, offering valuable insights into disease progression and treatment strategies.

3.3.3 Machine Learning Model Runner (FastAPI, Python)



The machine learning component processes DICOM images to detect lesions. This is implemented as a FastAPI service that the Golang API interacts with.

- Pre-processing: The input DICOM files undergo necessary transformations such as brain-area extraction (for brain MRIs only), denoising with anisotropic diffusion, bias field correction using N4, and normalization for intensity scaling.
- Model Inference: The processed images are run through a trained ML model that detects lesions and marks significant areas [3].
- Post-processing: The output results are formatted and prepared for visualization in the React application.

3.3.4 Databases (Orthanc & Cloud SQL)

- Orthanc DICOM Database: A specialized 3rd party medical imaging database used to store and retrieve DICOM files efficiently.
- Cloud SQL Database: A structured relational database that maintains user details, patient-doctor relationships, and associated metadata.

4. Development/Implementation Details

- **Golang REST API:** Implemented core backend services, including user authentication, DICOM management, and patient history endpoints, using Go modules and the standard library for high performance and concurrency.

- **Python FastAPI Service:** Built the lesion detection inference pipeline as a standalone FastAPI app, handling preprocessing, model inference, and postprocessing for mask generation.

- **React & TypeScript Frontend:** Developed the web UI with React and TypeScript, integrating VTK.js for interactive 2D slice views and 3D volume reconstruction of DICOM data.

- **VTK.js Visualization Library:** Utilized VTK.js to render large medical images smoothly in the browser, enabling slice navigation, opacity controls, and linked view synchronization.

- **Machine Learning Frameworks:** Employed PyTorch for training our custom lesion segmentation model and maintained a TensorFlow fallback implementation to ensure flexibility.

- **Orthanc DICOM Server:** Deployed Orthanc as our PACS solution for secure storage and retrieval of DICOM studies, interfaced via REST and DICOMweb protocols.

- **Cloud SQL Database:** Used Google Cloud SQL to store structured patient metadata, user roles, and lesion tracking records, with encrypted connections and backups.

- **Docker Containerization:** Containerized all services(Go API, FastAPI inference, Orthanc, frontend) with Docker Compose for consistent development environments, simplified deployment, and isolation of dependencies.

- **Git & GitHub:** Managed source code in private GitHub repositories, with feature branch workflows and regular code reviews.

- **Google Colab & Jupyter Notebooks:** Conducted ML experiments, data exploration, and preprocessing pipeline development interactively in notebooks, facilitating reproducibility and collaborative experimentation.

5. Test Cases and Results

5.1. Functional Test Cases

Test ID	01	Category	Functional	Severity	Minor
Objective	Test case for signing up with whitespace at the end of the email				
Steps	1. Go to the signup page 2. Fill in the email field with a valid email, followed by a space character 3. Fill in the password, name, and surname fields with valid. 4. Click the sign-in button 5. Check if there is an error message. 6. Check if the user received a verification email				
Expected	The request is accepted, and the spaces in the email are trimmed.				
Date-Result	April 11, Pass				

Test ID	02	Category	Functional	Severity	Major
Objective	Test sign-up with an already existing email				
Steps	1. Go to the sign-up page 2. Fill in the email, name, and surname fields with valid values. 3. Click the sign-up button. 4. Check if the verification email is received. 5. Fill in the verification code and click the verify button. 6. Log out. 7. Go to the sign-up page again. 8. Enter the same email and a different name. 9. Click sign-up 10. Check if an error message is obtained that says the user already exists				
Expected	An error message saying that the user already exists				
Date-Result	April 11, Pass				

Test ID	03	Category	Functional	Severity	Minor
Objective	Test sign-up with whitespace in the password				
Steps	1. Go to the sign-up page. 2. Fill in the email, name, and surname fields with valid values. 3. Fill in the password field with “aDrkp 12!PI” 4. Click sign-up. 5. Check for an error message saying that the password shouldn’t have a space character.				

	6. Check if an email is received
Expected	Error message saying that the password shouldn't have a space character. No email should be received
Date-Result	April 11, Pass

Test ID	04	Category	Functional	Severity	Major
Objective	Test sign-up with empty fields				
Steps	1. Go to the sign-up page. 2. Click the sign-up button without filling in any fields. 3. Check if there is an error message telling the user to fill in the fields.				
Expected	An error message telling the user to fill in the fields.				
Date-Result	April 11, Pass				

Test ID	05	Category	Functional	Severity	Minor
Objective	Test sign-up with empty name and surname fields				
Steps	1. Go to the sign-up page. 2. Fill in the email and password fields. 3. Click the sign-up button. 4. Check if there is an error message telling the user to enter their name.				
Expected	An error message telling the user to enter their name				
Date-Result	April 11, Pass				

Test ID	06	Category	Functional	Severity	Major
Objective	Test sign-up passwords with less than seven characters.				
Steps	1. Go to the sign-up page. 2. Fill in the name and password fields with valid values. 3. Enter "kRm1!" for the password field 4. Click the sign-up button. 5. Check for an error message telling the user to enter seven or more characters for the password.				

Expected	An error message telling the user to enter seven or more characters for the password.
Date-Result	April 11, Pass

Test ID	07	Category	Functional	Severity	Major
Objective	Test sign-up password without special characters				
Steps	1. Go to the sign-up page. 2. Fill in the name and password fields with valid values. 3. Enter “kRm1234” for the password field 4. Click the sign-up button. 5. Check for an error message telling the user that the password should contain a special character.				
Expected	An error message telling the user that the password should contain a special character.				
Date-Result	April 11, Pass				

Test ID	08	Category	Functional	Severity	Major
Objective	Test sign-up password without upper-case characters				
Steps	1. Go to the sign-up page. 2. Fill in the name and password fields with valid values. 3. Enter “krm1234!” for the password field 4. Click the sign-up button. 5. Check for an error message telling the user that the password should contain an upper-case character.				
Expected	An error message telling the user that the password should contain an upper-case character.				
Date-Result	April 18, Pass				

Test ID	09	Category	Functional	Severity	Major
Objective	Test sign-up password without a number				
Steps	1. Go to the sign-up page. 2. Fill in the name and password fields with valid values. 3. Enter “asDDdfg!” for the password field 4. Click the sign-up button.				

	5. Check for an error message telling the user that the password should contain a number.
Expected	An error message telling the user that the password should contain a number.
Date-Result	April 18, Pass

Test ID	10	Category	Functional	Severity	Critical
Objective	Test a valid sign-up request				
Steps	1. Go to the sign-up page. 2. Fill in the name as “Kerem” 3. Fill in the surname as “Sahin” 4. Fill in the email as a valid email 5. Fill in the password as “asDwer45?” 6. Click the sign-up button. 7. Check for success message 8. Check if an email has been received for verification.				
Expected	Success message, verification code email, redirected to email verification page.				
Date-Result	April 18, Pass				

Test ID	11	Category	Functional	Severity	Critical
Objective	Test sign-up with Turkish characters				
Steps	1. Go to the sign-up page. 2. Fill in the name as “Kerem” 3. Fill in the surname as “Şahin” 4. Fill in the email as a valid email 5. Fill in the password as “asDwer45?” 6. Click the sign-up button. 7. Check for success message 8. Check if an email has been received for verification.				
Expected	Success message, verification code email, redirected to email verification page.				
Date-Result	April 18, Pass				

Test ID	12	Category	Functional	Severity	Major

Objective	Test email verification with whitespace at the end of the email.
Steps	1. Sign up with valid account information. 2. Check if the verification email is received. 3. Check if it is redirected to the email verification page. 4. In the email field, enter the correct email followed by space characters. 5. Enter the correct verification code. 6. Click Verify. 7. Check if a success message is received.
Expected	A success message, is redirected to the dashboard.
Date-Result	April 18, Pass

Test ID	13	Category	Functional	Severity	Major
Objective	Test email verification with empty fields.				
Steps	1. Sign-up with valid account information. 2. Check if the verification code is received. 3. Check if it is redirected to the email verification page. 4. Click Verify. 5. Check for an error message telling the user to enter the verification code.				
Expected	An error message telling the user to enter the verification code.				
Date-Result	April 18, Pass				

Test ID	14	Category	Functional	Severity	Critical
Objective	Test email verification with incorrect code.				
Steps	1. Sign-up with valid account information. 2. Check if the verification code is received. 3. Check if it is redirected to the email verification page. 4. Enter an invalid verification code. 5. Check for an error message telling the user that the verification code is incorrect.				
Expected	An error message telling the user that the verification code is incorrect.				
Date-Result	April 18, Pass				

Test ID	15	Category	Functional	Severity	Critical

Objective	Test email verification with the valid verification code.				
Steps	1. Sign-up with valid account information. 2. Check if the verification code is received. 3. Check if it is redirected to the email verification page. 4. Enter the valid verification code. 5. Check if a success message is received.				
Expected	A success message, is redirected to the dashboard.				
Date-Result	April 18, Pass				

Test ID	16	Category	Functional	Severity	Major
Objective	Test log-in with an invalid email				
Steps	1. Go to the log-in page. 2. Enter a non-registered email for the email field. 3. Enter a password for the password field. 4. Click the log-in button. 5. Check for an error message saying that the email or password is incorrect.				
Expected	An error message saying that the email or password is incorrect.				
Date-Result	April 18, Pass				

Test ID	17	Category	Functional	Severity	Minor
Objective	Test log-in with whitespace after the email.				
Steps	1. Go to the log-in page. 2. Enter a valid email with spaces after it. 3. Enter the correct password. 4. Click log-in. 5. Check for a success message.				
Expected	Success message, is redirected to the dashboard. Cookie with authorization token is received.				
Date-Result	April 18, Pass				

Test ID	18	Category	Functional	Severity	Major
Objective	Test log-in with empty fields.				
Steps	1. Go to the log-in page.				

	2. Click the log-in button. 3. Check for an error message telling the user to fill in the email and password fields.
Expected	An error message telling the user to fill in the email and password fields.
Date-Result	April 18, Pass

Test ID	19	Category	Functional	Severity	Critical
Objective	Test valid log-in request				
Steps	1. Go to the log-in page. 2. Enter a registered email for the email field. 3. Enter the correct password for the password field. 4. Click the log-in button. 5. Check for a success message.				
Expected	Success message, is redirected to the dashboard. Cookie with authorization token is received.				
Date-Result	April 18, Pass				

Test ID	20	Category	Functional	Severity	Critical
Objective	Test log-in with an incorrect password.				
Steps	1. Go to the log-in page. 2. Enter a registered email for the email field. 3. Enter the incorrect password for the password field. 4. Click the log-in button. 5. Check for an error message saying that the email or password is incorrect.				
Expected	An error message saying that the email or password is incorrect.				
Date-Result	April 18, Pass				

Test ID	21	Category	Functional	Severity	Critical
Objective	Test valid DICOM study quick upload				
Steps	1. Log in to an account 2. Click Quick Upload. 3. Select all the DICOMs stored in a file and click OK. 4. Check for a success message.				

Expected	A success message telling the user the study has been uploaded. Redirects to the study page.
Date-Result	April 25, Pass

Test ID	22	Category	Functional	Severity	Critical
Objective	Test valid DICOM study quick uploads with zip.				
Steps	1. Log in to an account 2. Click Quick Upload. 3. Select a zip file that contains valid DICOM files. 4. Check for a success message.				
Expected	A success message telling the user the study has been uploaded. Redirects to the study page.				
Date-Result	April 25, Pass				

Test ID	23	Category	Functional	Severity	Major
Objective	Test invalid file type quick study upload.				
Steps	1. Log into an account. 2. Click Quick Upload. 3. Select a .txt file and click OK. 4. Check for an error message telling the user that the file type is invalid.				
Expected	An error message telling the user that the file type is invalid.				
Date-Result	April 25, Pass				

Test ID	24	Category	Functional	Severity	Major
Objective	Test empty DICOM file for quick study upload.				
Steps	1. Log into an account. 2. Click Quick Upload. 3. Select a DICOM file that is empty. 4. Click OK. 5. Check for an error message about the DICOM file being empty.				
Expected	Check for an error message about the DICOM file being empty.				

Date-Result	April 25, Pass

Test ID	25	Category	Functional	Severity	Major
Objective	Test NIFTI file quick upload				
Steps	1. Log in to an account 2. Click Quick Upload. 3. Select a NIFTI file and click OK. 4. Check for a success message.				
Expected	A success message telling the user the study has been uploaded. Redirects to the study page.				
Date-Result	TBD				

Test ID	26	Category	Functional	Severity	Critical
Objective	Test CT scan quick upload				
Steps	1. Log in to an account 2. Click Quick Upload. 3. Select a study containing a CT scan and click OK. 4. Check for an error message telling the user that only MRI’s are allowed.				
Expected	An error message telling the user that only MRI’s are allowed.				
Date-Result	April 25, Pass				

Test ID	27	Category	Functional	Severity	Critical
Objective	Test an MRI quick upload that is not brain or spine MRI.				
Steps	1. Log into an account 2. Click Quick Upload. 3. Select a hand MRI and click OK. 4. Check for an error message telling the user that only brain and spine MRIs are allowed				
Expected	An error message telling the user that only brain and spine MRIs are allowed				
Date-Result	April 25, Fail				

Test ID	28	Category	Functional	Severity	Critical
Objective	Test a brain MRI quick upload				
Steps	1. Log into an account. 2. Click Quick Upload. 3. Select a brain MRI DICOM series and click ok. 4. Check for a success message.				
Expected	A success message telling the user the study has been uploaded. Redirects to the study page.				
Date-Result	April 25, Pass				

Test ID	29	Category	Functional	Severity	Critical
Objective	Test a spine MRI quick upload				
Steps	1. Log into an account. 2. Click Quick Upload. 3. Select a spine MRI DICOM series and click ok. 4. Check for a success message.				
Expected	A success message telling the user the study has been uploaded. Redirects to the study page.				
Date-Result	TBD				

Test ID	30	Category	Functional	Severity	Critical
Objective	Test the creation of a valid patient				
Steps	1. Log into an account. 2. Click Patient History. 3. Click New Patient. 4. Fill in the patient's name and TCK number. 5. Click Create Patient. 6. Check for a success message 7. Check if the patient appears in the patient history tab.				
Expected	Success message telling the user that the patient record is created. The patient should appear in the patient history tab.				
Date-Result	April 28, Pass				

Test ID	31	Category	Functional	Severity	Major

Objective	Test creation of a patient with empty fields.
Steps	1. Log into an account. 2. Click Patient History. 3. Click New Patient. 4. Click Create Patient. 5. Check for an error message telling the user to fill in the name and TCK number.
Expected	An error message telling the user to fill in the name and TCK number.
Date-Result	April 28, Pass

Test ID	32	Category	Functional	Severity	Major
Objective	Test creation of a patient with a short TCK number.				
Steps	1. Log into an account. 2. Click Patient History. 3. Click New Patient. 4. Enter a name and an 8 digit TCK number. 5. Click Create Patient. 6. Check for an error message telling the user that the TCK number should be 11 characters long.				
Expected	An error message telling the user that the TCK number should be 11 characters long.				
Date-Result	April 28, Pass				

Test ID	33	Category	Functional	Severity	Critical
Objective	Test brain MRI upload for a specific patient.				
Steps	1. Log into an account. 2. Click patient history. 3. Select a patient. 4. Click Add Scan. 5. Select a brain MRI DICOM file and click OK. 6. Check for a success message. 7. Check if the new scan is added to the patient’s history.				
Expected	A success message telling the user the upload was successful. The scan should be added to the patient’s list.				
Date-Result	April 28, Pass				

Test ID	34	Category	Functional	Severity	Critical
Objective	Test 3D scan view.				
Steps	1. Log into an account. 2. Click patient history and select a patient. 3. Select a scan. 4. Check if it is redirected to the Scan View page. 5. Check if all three axes show the scan as expected, along with the 3D view. 6. Put the mouse on the coronal axis and scroll. Check if the coronal axis view shows the next slices as the user scrolls. 7. Check if the coronal plane on the 3D view is moving accordingly.				
Expected	Showing the chosen scan properly.				
Date-Result	April 28, Pass				

Test ID	35	Category	Functional	Severity	Critical
Objective	Test ML analysis of a study				
Steps	1. Log into an account. 2. Click patient history and select a patient. 3. Select a scan. 4. Check if it is redirected to the Scan View page. 5. Click the ML analysis button. 6. Check for a success message 7. Check if the mask overlay is obtained. 8. Check if there is a slider for mask opacity. 9. Minimize the mask opacity slider and see if the mask disappears.				
Expected	Obtained ML analysis and mask overlay.				
Date-Result	TBD				

Test ID	36	Category	Functional	Severity	Critical
Objective	Test to check ML analysis of a previously done analysis				
Steps	1. Log into an account. 2. Click patient history and select a patient. 3. Select a scan that already has an ML analysis. 4. Check if it is redirected to the Scan View page. 5. Check if the mask overlay is obtained. 6. Check if there is a slider for mask opacity. 7. Minimize the mask opacity slider and see if the mask disappears.				

Expected	Mask overlay persists for previously done analysis.
Date-Result	April 28, Pass

Test ID	37	Category	Functional	Severity	Critical
Objective	Test the tracking of lesion growth throughout several scans				
Steps	1. Log into an account. 2. Click patient history and select a patient. 3. Select a scan that already has an ML analysis. 4. Check if it is redirected to the Scan View page. 5. Click Track Lesion History. 6. Check if there are multiple scans throughout the years on the right side. 7. Check if the program shows the edges of lesions through the scans. 8. Check if the user can go to a specific scan from this view by clicking on it.				
Expected	The user can view the growth of the lesion through several scans				
Date-Result	April 28, Pass				

Test ID	38	Category	Functional	Severity	Critical
Objective	Test MRI upload with different manufacturers.				
Steps	1. Log into an account 2. Click Quick Upload. 3. Upload MRI scans from different manufacturers (Siemens, GE, Philips, Canon, etc.). 4. Request ML analysis. 5. Check lesion detection and tracking work with different manufacturers.				
Expected	The system should work with every MRI manufacturer				
Date-Result	April 28, Pass				

Test ID	39	Category	Functional	Severity	Major
Objective	Test uploading study with multiple brain scans.				

Steps	1. Log into an account. 2. Click Quick Upload. 3. Upload a study with multiple scans, all of which are brain scans. 4. Check for a success message. 5. Check if all scans of the study are visible. 6. Click ML study 7. Check if the optimal scan is selected from the set of scans for ML analysis.
Expected	If multiple scans are available, the system should select the most optimal one available for ML analysis. (FLAIR, then T2, then T1).
Date-Result	TBD

Test ID	40	Category	Functional	Severity	Major
Objective	Test uploading study with multiple scans, containing multiple body parts.				
Steps	1. Log into an account. 2. Click Quick Upload. 3. Upload a study with multiple scans, the first of which is a brain scan, the others are of non-accepted body parts. 4. Check for an error message.				
Expected	The user should get an error message telling the user that only brain and spine MRIs are accepted.				
Date-Result	April 28, Fail				

5.2. Non-Functional Test Cases

Test ID	41	Category	Non-functional	Severity	Major
Objective	Test for SQL injection.				
Steps	1. Go to the log-in and/or sign-in screen. 2. Enter SQL injection strings such as “OR 1=1” for the fields. 3. Check if an error message is received.				
Expected	Error warning the user that these values are not accepted.				
Date-Result	April 14, Pass				

Test ID	42	Category	Non-functional	Severity	Major

Objective	Test to check compatibility with different web browsers.
Steps	1. Iterate through different browsers such as Chrome, Opera, Firefox, Safari… 2. Check if UI objects are properly placed on various screens.
Expected	The web app works as expected in different web browsers.
Date-Result	April 14, Pass

Test ID	43	Category	Non-functional	Severity	Major
Objective	Test compatibility with different operating systems				
Steps	1. Use several operating systems such as Windows, MacOS, Linux and check if the web app is working correctly. 2. Check if UI objects are visible on the screen, the text is readable, and everything fits the screen.				
Expected	The application is compatible with all operating systems.				
Date-Result	April 14, Pass				

Test ID	44	Category	Non-functional	Severity	Major
Objective	Test compatibility with different window sizes.				
Steps	1. Use several window sizes and check if the web app is working correctly. 2. Check if UI objects are visible on the screen, the text is readable, and everything fits the screen.				
Expected	The application is usable in a variety of screen sizes.				
Date-Result	April 14, Pass				

Test ID	45	Category	Non-functional	Severity	Major
Objective	Stress testing				
Steps	1. Send 1000 sign-up requests at once. 2. Check if the server can handle this many requests.				
Expected	The server should be able to handle 1000 sign-up requests at once.				

Date-Result	April 14, Pass

Test ID	46	Category	Non-functional	Severity	Critical
Objective	Verify system stability and data integrity after unexpected failures.				
Steps	1. Simulate an unexpected system crash while processing an MRI scan. 2. Restart the system and check if the scan data remains intact. 3. Attempt a forced power outage or network disconnect during lesion analysis. 4. Verify if the system can resume operations without data corruption.				
Expected	The system maintains data integrity and resumes from the last valid state without loss or corruption.				
Date-Result	TBD				

Test ID	47	Category	Non-functional	Severity	Critical
Objective	Assess the system’s ability to handle large MRI datasets.				
Steps	1. Upload MRI datasets of increasing sizes (e.g., 1GB, 5GB, 10GB). 2. Monitor processing time and system memory usage. 3. Check for system slowdowns or crashes.				
Expected	The system efficiently processes large MRI datasets without crashes or excessive slowdowns.				
Date-Result	TBD				

Test ID	48	Category	Non-functional	Severity	Major
Objective	ML analysis time testing				
Steps	1. Request an ML analysis 2. Check that the mask is received within 2 minutes.				
Expected	The mask should be received within 2 minutes.				
Date-Result	May 01, Pass				

Test ID	49	Category	Non-functional	Severity	Critical
Objective	ML analysis accuracy testing				
Steps	1. Run the ML model on a large dataset 2. Compare the generated mask to the actual mask and compute the average IoU score. 3. Check that the IoU score is above 0.5				
Expected	IoU score should be above 0.5				
Date-Result	May 01, Pass				

Test ID	50	Category	Non-functional	Severity	Crtical
Objective	Test compliance with HIPAA and GDPR data protection standards.				
Steps	1. Check if MRI scans and patient records are encrypted in storage and transit. 2. Attempt to access restricted patient data without authentication. 3. Verify proper session timeout mechanisms. 4. Ensure user roles and access levels prevent unauthorized access.				
Expected	The system enforces strong encryption, access control, and compliance with medical data privacy regulations.				
Date-Result	April 28, Pass				

6. Maintenance Plan and Details

Our system comprises several components that evolve alongside advances in technology, and each module must be maintained to ensure ongoing reliability and performance. At the core of our analytical pipeline are the machine-learning segmentation and detection models. We currently leverage a custom trained PyTorch model to identify lesions in MRI scans. When novel model architectures or superior pre-trained weights become available, we can replace our existing model with minimal disruption by using the same FastAPI inference interface.

On the frontend, our React based viewer relies on VTK.js to render both 2D slice views and 3D volume reconstructions of MRI data. As the VTK.js project releases new versions, or if alternative WebGL based visualization libraries emerge, we simply update our npm dependency and verify that our TypeScript bindings remain compatible, ensuring that clinicians always benefit from the latest rendering improvements.

The backend is powered by a GoLang REST API and a Python FastAPI service for ML inference, each depending on versioned modules. To maintain security and performance, we upgrade to the latest long term support(LTS) releases of Go and Python, and apply patch releases for FastAPI, Orthanc, and our database drivers as they become available. This keeps our core services aligned with upstream fixes and best practices.

All medical images and associated metadata reside in an Orthanc DICOM server alongside a Cloud SQL database. Should Orthanc itself receive a major upgrade, or if a more advanced PACS solution becomes preferable, we will follow its release notes and migrate using our established backup-and-restore scripts, preserving both historical patient records and system continuity.

To coordinate these updates, we follow a structured maintenance cadence. On a monthly basis, we pull in minor and patch updates for all dependencies, run our full suite of automated tests, and deploy to a staging environment. Quarterly, we evaluate major version upgrades, such as Go 1.x → 1.y or VTK.js vX → vX+1, and carry out comprehensive compatibility testing. Finally, on an annual cycle, we reassess our machine learning architecture and the datasets used for training; if clinical validation justifies it, we retrain existing models or introduce cutting-edge alternatives.

Combining automated dependency checks, disciplined release schedules, and thorough testing at every stage, LesionLens will remain a secure, high performance, and clinically up-to-date platform throughout its operational lifespan.

7. Other Project Elements

7.1. Consideration of Various Factors in Engineering Design

In this section of the report, we discuss the various design considerations for our project. There are four significant ethical risks with using artificial intelligence in medical imaging: autonomy of patients and clinicians, transparency of clinical performance and limitations, fairness toward marginalized populations, and accountability of physicians and developers [4].

7.1.1 Public Health Considerations

Lesion Lens aims to improve public health conditions by providing a fast and reliable way for clinicians to analyse MRIs for MS lesions. While improving the lesion detection process, Lesion Lens respects the autonomy of doctors. It doesn't provide diagnosis but it serves rather as a tool to assist doctors. During our visits to Hacettepe University we have learned that doctors have a very limited time to analyze each MRI due to a large number of patients, which creates the risk of missing certain lesions. Our project serves as a second set of eyes to ensure these errors are minimized.

7.1.2 Public Safety Considerations

Lesion Lens stores sensitive patient information in the database. Therefore, the consent of patients and a secure database is of utmost importance. To ensure the security of this sensitive information, we provide an option to upload scans anonymously.

7.1.3 Public Welfare Considerations

For Lesion Lens to be in accordance with the welfare of the public, we need to ensure that our product is fair towards marginalized groups. This means that we need to select our ML datasets carefully to ensure they are representative of all parts of a population. Fortunately, our ML models were trained using robust and representative datasets from MICCAI [5].

7.1.4 Global Considerations

Even though Lesion Lens originates from the needs of Hacettepe Hospital, it is planned to be deployed in various healthcare systems worldwide, therefore it must be designed to function effectively across diverse settings. The application should support multiple languages to cater to users from different linguistic backgrounds. The machine learning model must adapt to variations in medical imaging techniques and standards used in different regions, such as MRI scans taken from different machines. Integration with globally recognized healthcare protocols and interoperability with regional hospital systems are essential. The system should also account for differences in infrastructure, such as varying internet speeds, ensuring usability even in areas with limited resources. Through addressing these considerations, Lesion Lens can be a universally accessible tool for medical imaging analysis.

7.1.5 Cultural Considerations

Cultural considerations play a key role in making Lesion Lens accessible and acceptable across different regions. The application's interface and communication materials should be localized for language, medical terminology, and visual cues familiar to local users. Doctors and patients may have varying levels of trust in automated systems, so transparency about the tool's role and data usage can help build confidence. Compliance with regional laws, ethical guidelines, and hospital protocols ensures that patient privacy and data handling respect local values and regulations. With the help of adapting to cultural norms, such as accommodating different communication styles, workflows, and trust in medical authority, Lesion Lens can integrate seamlessly into diverse healthcare environments.

7.1.6 Social Considerations

Social considerations involve understanding how Lesion Lens fits into existing relationships between patients, doctors, and hospital staff. The tool should support, not replace, the expertise of healthcare professionals, maintaining trust and respecting traditional roles. Training and clear communication ensure that users feel confident adopting the new technology, while patients remain informed and reassured. Additionally, considerations like equitable access, ensuring that all who need the tool can use it, help prevent disparities in care. Integrating smoothly with existing workflows and respecting the social fabric of healthcare settings can help Lesion Lens to promote positive collaboration and ultimately improve patient outcomes.

7.1.7 Environmental Considerations

Environmental considerations for Lesion Lens center on responsibly managing resource usage in both development and deployment. Efficient cloud computing strategies, such as using only necessary storage and computing power, help reduce the application's carbon footprint. Hardware optimization, like leveraging existing hospital equipment and avoiding unnecessary hardware upgrades, minimizes waste. Furthermore, proper data management and secure, remote access can reduce the need for physical travel or transfer of information, indirectly lessening environmental impact. Ultimately, by prioritizing energy efficiency, resource conservation, and minimal hardware dependency, Lesion Lens can contribute to a more sustainable healthcare technology ecosystem.

7.1.8 Economic Considerations

Economic considerations for Lesion Lens focus on cost effectiveness and long term sustainability. Reducing manual analysis time through automated lesion detection can lower operational expenses and potentially increase patient throughput. Using open source libraries and publicly available training datasets can minimize initial development costs, while the scalability of cloud-based infrastructure allows for flexible resource allocation as usage grows. Careful balancing of hardware investments, data storage expenses, and maintenance fees ensures that the tool remains affordable, accessible, and provides a return on investment for healthcare providers over time.

Title	Effect Level	Effect
Public Health	9	Improves early detection of

		lesions, potentially reducing disease progression and improving overall healthcare outcomes.
Public Safety	3	Ensures patient data is secure through anonymized uploads, protecting sensitive information [6].
Public Welfare	5	Reduces bias by using representative ML datasets, promoting fair and equitable healthcare.
Global	3	Supports diverse healthcare systems by ensuring multilingual support, regional medical imaging standards, and operability in varying infrastructure conditions.
Cultural	1	Builds trust and accessibility through localized interfaces, compliance with regional ethics and laws, and respecting cultural medical practices.
Social	1	Maintains harmony in healthcare environments by supporting professionals' expertise, providing training, and ensuring equitable access to all users.
Environmental	2	Reduces environmental impact with efficient computing, hardware reuse, and sustainable data management practices.
Economic	7	Enhances cost-effectiveness by minimizing manual work, leveraging FOSS resources, & implementing scalable cloud infrastructure for long-term affordability & ROI.

Table 1: Factors that can affect analysis and design.

7.2. Ethics and Professional Responsibilities

- The source code and project documentation have been maintained in private repositories on GitHub, accessible only to the project group members, supervisor, and course graders. The source code has not been shared with third parties during the development phase to ensure confidentiality and academic integrity.

- All external libraries, frameworks, and tools (including VTK.js, FastAPI, Orthanc, TensorFlow, and PyTorch) have been used in accordance with their respective licenses. Proper acknowledgments and attributions have been included where necessary, in compliance with open source license requirements.

- Strict measures have been taken to protect sensitive medical data used in testing. Only publicly available, anonymized datasets were used during model development and testing, ensuring that no real patient-identifiable information was handled improperly.

- Secure authentication, encryption, and role-based access controls have been integrated into the system to safeguard patient data privacy and comply with healthcare regulations such as GDPR and HIPAA.

- Every Wednesday during the Fall semester and every Friday during the Spring semester, weekly face-to-face or online meetings were conducted with the team and either our supervisor or innovation expert. These meetings ensured consistent communication, allowed for early feedback, and maintained alignment with ethical and technical goals.

- Model limitations, potential sources of error, and intended clinical use cases have been clarified within the system interface and user documentation to ensure transparency and informed usage by clinicians.

- Feedback from medical professionals was actively sought throughout the project to ensure that the system meets clinical standards and supports accurate and fair diagnostic workflows, minimizing risks of bias or misuse.

7.3. Teamwork Details

Successful execution of Lesion Lens depends on effective collaboration among team members with diverse expertise. Our approach to teamwork is structured to leverage individual strengths while providing a supportive, inclusive environment.

7.3.1 Contributing and functioning effectively on the team

- All team members meet weekly to discuss progress, challenges, and next steps.
- Responsibilities are assigned based on expertise: for instance, one subgroup focuses on medical image preprocessing and training ML models, while another develops the backend and user interface.
- Regular code reviews and joint brainstorming sessions ensure that every member contributes to both the technical and strategic aspects of the project.
- All team members contributed equally to the preparations of reports.
- Participated in the project demo sessions together as a group.

- The training of lesion detecting ML model is being done by Mert, with the help of all members.
- Implementation of required MRI preprocessing was done by Mert.
- The backend API was constructed by Berk, with the help of Kerem.
- Database construction was done by Kerem, with the help of Berk.
- The 3D viewer module was implemented by Shayan and Arda.
- The frontend of our project was designed and implemented by Shayan and Arda.

7.3.2 Helping creating a collaborative and inclusive environment

Throughout the development of Lesion Lens, our team has made a deliberate effort to enable a collaborative and inclusive work environment, in which all members contribute effectively while learning from one another. Our key strategies include:

1. **Establishing Clear Goals and Workload Distribution**
   At the outset, we defined clear, realistic objectives and set a timeline that allowed for steady progress. Roles and responsibilities were periodically revisited to ensure fair distribution of workload, and contributions were recognized in team meetings and project documentation. Therefore, we minimized misunderstandings and ensured that no team member was overburdened.
2. **Role Assignment Based on Individual Strengths**
   We assigned roles and responsibilities by leveraging each member's expertise and interests. This approach maximized our collective productivity and allowed team members to deeply engage with their areas of strength, consequently improving the overall quality of our project.
3. **Regular Peer Review and Collaborative Problem Solving**
   Despite working on different components independently to enhance parallel efficiency, we held frequent review sessions where members examined each other's work. This ensured that our diverse contributions were integrated well and that potential issues were identified and addressed early in the development cycle.
4. **Mutual Respect and Open Communication**
   The team maintained an open communication environment where all opinions were valued. When differences in ideas arose, we engaged in constructive discussions, often involving our advisors, to reach consensus.
5. **Providing Support and Mentorship**
   Throughout the project, more experienced members readily offered guidance to those facing challenges, whether in theoretical understanding or practical implementation. This mentorship helped build individual competencies and ensured that the entire team advanced together.

7.3.3 Taking lead role and sharing leadership on the team

- Leadership is distributed among team members, with different individuals taking the lead on various components such as model development, data preprocessing, and system integration.
- Decisions are made collectively after thorough discussion and consultation with our project supervisor, ensuring that multiple perspectives are considered.

- Being our project manager, Berk led the organization of meetings both within our team and with our advisors and customers. He also led the issue tracking of our project using GitHub Projects.
- In terms of implementation, Berk led the construction of backend API, and he led integration of backend with frontend.
- Mert finalized the required medical imaging preprocessing steps through literature review. He led the design, training and implementation of ML models.
- Kerem led the construction of our database, and implemented the testing modules for individual sections of our project. He led the integration of our ML models with the backend.
- Shayan led the frontend design and development. He also led the implementation of the 3D viewer module.
- Arda led the research in ML model architectures and he supported Mert in ML model implementation. He also led the integration processes between the backend and 3D viewer module.

7.3.4 Meeting Objectives

1. **Finalize lesion classification criteria and progress-tracking workflow**

   In our Analysis Report, we identified the need to establish clear criteria for categorizing multiple sclerosis lesions and a robust mechanism for logging project milestones. During our first progress meeting, we presented that we have agreed on the specific lesion types to target,namely MS lesions on FLAIR hyperintense areas. With our annotation scripts and PyTorch data loaders now implemented according to these definitions, this objective is complete.

2. **Establish documentation standards and set a deliverables schedule**

   The team intended to establish a common template for documenting the outcomes of ML experiments, feature development advancements, and official deliverables in order to guarantee uniform communication and accountability. We established a biweekly deliverable cadence, formalized our documentation templates, and made a commitment to updating our GitHub Projects logs on time during our first supervisory meeting. This milestone has been reached, and our documentation process is now completely functional.

3. **Design an ML model evaluation protocol and identify next generation architectures**

   The Analysis Report called for an objective testing procedure, using metrics such as Intersection over Union and Dice Similarity Coefficient, and a shortlist of advanced segmentation models, including transformer-based approaches. In our second meeting with our supervisor, we reviewed our initial evaluation scripts, discussed recent transformer-augmented architectures from the literature, and assembled a shortlist of candidates for benchmarking. Since our evaluation framework is in place, this objective is currently complete.

4. **Integrate core system modules into an end-to-end pipeline**

A critical early goal was to demonstrate a working workflow that uploads DICOM studies, performs ML inference, and displays lesion masks in the React/VTK.js viewer. During our second progress meeting this semester, we showcased the successful communication between our GoLang REST API, and the frontend visualization. All data flows have been validated, and the demo confirms that this integration milestone is complete.

5. **Develop comprehensive test suites and produce a demo video**

   Our project plan included delivering a full suite of functional and non-functional test cases alongside a short demonstration video. In our third progress meeting in the second semester, we presented over forty test cases covering criteria from DICOM ingestion to ML accuracy thresholds, as well as a draft of our highlight reel. Both our test coverage and demo video are in their final stages, tests have been executed successfully, and the video is ready, thereby fulfilling this objective.

6. **Deploy the system to AWS and optimize operational costs**

   We aimed to host LesionLens services in the cloud, complete with CI/CD pipelines and cost monitoring tools. In the last progress meeting, we walked through our AWS architecture and demonstrated API endpoints. Following our instructor's recommendations on reserved instances and autoscaling, we have applied some cost-optimization measures. The system is now live in AWS, indicating this milestone was achieved.

7. **Prepare for the CS Fair and finalize the project report**

   As the semester drew to a close, our final objective was to be ready for the CS Fair demo, poster, and comprehensive report. Also in the last meeting, we outlined the storyboard for our booth presentation, selected which features to highlight, and confirmed the structure for final report sections. With demo materials in development and report drafts nearly complete, this objective is underway and on track for final submission.

7.4. New Knowledge Acquired and Applied

- GoLang programming language, gained experience in building RESTful APIs, implementing JWT-based authentication, and handling concurrent requests efficiently. None of the team members had prior experience with Go before the project.

- VTK.js framework for medical image visualization, we learned to integrate advanced 2D and 3D MRI rendering capabilities into a React application. This included slice navigation, linked views, and interactive visualization of volumetric data, which are transferable skills for other medical imaging or scientific visualization projects.

- FastAPI framework in Python, we became proficient in developing a lightweight and high performance web API used to run machine learning inference models for lesion detection. This included handling asynchronous requests and optimizing image processing pipelines.

- DICOM standards and Orthanc server, we gained hands-on experience in managing, storing, and retrieving medical images in DICOM format. This required learning DICOM metadata structure and communication protocols, which are crucial in healthcare applications.

- TensorFlow and PyTorch frameworks for developing and deploying machine learning models for medical image segmentation. We explored training pipelines, model evaluation metrics (e.g., IoU for mask accuracy), and model inference optimizations.

- Cloud SQL and relational database design, we applied secure and efficient storage of patient records, user roles, and historical scan data while ensuring GDPR-compliant handling of sensitive medical information.

- Weekly use of Google Colab and Jupyter notebooks (for experimentation on machine learning preprocessing pipelines) and collaborative development with GitHub. This required careful coordination to avoid overwriting work and to maintain clean and reproducible codebases.

In the healthcare industry, where system transparency, maintainability, and clear communication of limitations are just as important as technical sophistication, we have developed a profound appreciation for striking a balance between the latest developments and dependability. In order to maintain consistent project progress while overcoming interdisciplinary obstacles, we also learned to give priority to iterative, outcome-oriented development workflows.

8. Conclusion and Future Work

LesionLens is a comprehensive web-based platform that combines AI-driven lesion detection and historical tracking with interactive 2D slice and 3D volume visualizations to simplify the analysis of MS lesions in brain FLAIR MRI scans. It incorporates a GoLang REST API, a machine learning inference service driven by FastAPI, a React/VTK.js frontend, and secure DICOM data management, all built on a modular architecture. LesionLens provides a dependable, easy-to-use tool that helps clinicians diagnose and track multiple sclerosis through role-based access control, encryption, and adherence to healthcare standards.

Looking forward, there are several avenues to extend and enhance LesionLens. In order to facilitate smooth patient data exchange and report distribution, we first intend to further integrate the system into hospital workflows by putting in place DICOMweb and FHIR-based APIs that connect directly to electronic health record (EHR) systems. Our second goal is to create an automated reporting module that reduces the amount of manual documentation work by producing longitudinal progression charts and standardized lesion measurement summaries.

Moreover, we will expand the AI component by incorporating additional imaging modalities, such as diffusion-weighted imaging(DWI) or functional MRI, and by retraining on larger, more diverse datasets, potentially using federated learning to preserve patient privacy across institutions. Finally, to improve the interface, maximize inference speed (including GPU acceleration), and make sure LesionLens satisfies practical clinical needs, we plan to carry out official clinical validation studies and user-experience assessments with medical experts. We believe that LesionLens can develop into a vital tool for neurological imaging and patient care with these improvements.

9. User Manual and Installation Guide

The comprehensive LesionLens user manual, including installation instructions, system requirements, and step-by-step guidance on uploading MRI documents, running ML analysis, and navigating the 2D/3D viewer, is available online. You can access the full user guide at:

https://scorpia2004.github.io/LesionLensWeb/userguide

10.Glossary

AI (Artificial Intelligence)
The simulation of human intelligence processes by machines, including learning, reasoning, and self-correction.

API (Application Programming Interface)
A set of protocols, routines, and tools for building software applications that enable different systems to communicate and interact.

Cloud SQL
A managed relational database service that provides scalable and secure storage and retrieval of structured data in a cloud environment.

DICOM (Digital Imaging and Communications in Medicine)
An international standard for handling, storing, printing, and transmitting information in medical imaging. It ensures interoperability between different imaging devices and systems.

FastAPI
A modern, high-performance web framework for building APIs with Python. It is known for its ease of use, speed, and automatic documentation generation.

FOSS
Free and open-source software (FOSS) is software available under a license that grants users the right to use, modify, and distribute the software – modified or not – to everyone free of charge.

Golang (Go)
A statically typed, compiled programming language developed by Google, designed for efficiency, scalability, and ease of maintenance in software systems.

JWT (JSON Web Token)
A compact, URL-safe means of representing claims between two parties. It is commonly used for securely transmitting information for authentication and authorization.

Lesion
An area of abnormal tissue in the body, such as those seen in brain MRIs, often associated with diseases like multiple sclerosis.

Machine Learning (ML)
A branch of artificial intelligence that uses algorithms and statistical models to enable computers to learn from and make predictions or decisions based on data, without explicit programming.

MRI (Magnetic Resonance Imaging)
A non-invasive imaging technology that uses magnetic fields and radio waves to produce detailed images of the organs and tissues within the body.

Orthanc
An open-source DICOM server designed to manage, store, and retrieve medical images efficiently, supporting interoperability between different imaging systems.

React
A widely used JavaScript library for building user interfaces, particularly for single-page applications. It facilitates the creation of interactive, dynamic web applications.

SOP (Service-Object Pair)
In the context of DICOM, a pairing of a service (e.g., image storage or retrieval) with an information object, defining how medical imaging data should be exchanged.

VTK.js
A JavaScript library for scientific visualization that supports rendering of 2D and 3D images. It is used in applications like Lesion Lens for interactive medical image visualization.

11. References

[1] National Electrical Manufacturers Association. (n.d.). *Digital Imaging and Communications in Medicine (DICOM) Standard (NEMA PS3/ISO 12052)* [Standard]. http://www.dicomstandard.org/. [Accessed Mar. 9, 2025].

[2] Ronneberger, O., Fischer, P., & Brox, T. (2015b). U-NET: Convolutional Networks for Biomedical Image Segmentation. In *Lecture notes in computer science* (pp. 234–241). https://doi.org/10.1007/978-3-319-24574-4_28. [Accessed Jan. 20, 2025].

[3] Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., & Wang, M. (2021). Swin-UNet: UNET-like pure transformer for medical image segmentation. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2105.05537. [Accessed Feb. 13, 2025].

[4] J. Herington et al., "Ethical Considerations for Artificial intelligence in Medical Imaging: Deployment and governance," Journal of Nuclear Medicine, vol. 64, no. 10, pp.
1509–1515, Aug. 2023, doi: 10.2967/jnumed.123.266110. [Accessed: Dec. 15, 2024].

[5] Commowick O et al., "Multiple sclerosis lesions segmentation from multiple experts: The MICCAI 2016 challenge dataset". Neuroimage. 2021 Dec 1;244:118589. doi: 10.1016/j.neuroimage.2021.118589. Epub 2021 Sep 24. PMID: 34563682. [Accessed: Oct. 30, 2024].

[6] Voigt, P., & Von Dem Bussche, A. (2017). The EU General Data Protection Regulation (GDPR). In *Springer eBooks*. https://doi.org/10.1007/978-3-319-57959-7. [Accessed: Mar. 7, 2025].