**COMSATS University Islamabad,   
Abbottabad Campus**

**SOFTWARE DESIGN DESCRIPTION   
(SDD DOCUMENT)**

**for**

**Liver Tumor Segmentation in CT Scan images Using Light Weight Deep Learning Model**  
Version 1.0

***By***

**Faizan CIIT/FA21-BSE-011/ATD**

**Fawad Iqbal CIIT/FA21-BSE-012/ATD**

***Supervisor*Dr. Mubashir Ahmad**

***Bachelor of Science in Computer Science (2024-2025)***

Table of Contents

[Introduction 5](#_Toc186502841)

[Design methodology and software process model 5](#_Toc186502842)

[Design Methodology 5](#_Toc186502843)

[Software Process Model 5](#_Toc186502844)

[System overview 6](#_Toc186502845)

[Functionality 6](#_Toc186502846)

[Context 7](#_Toc186502847)

[Design 7](#_Toc186502848)

[Background Information 7](#_Toc186502849)

[Architectural design 8](#_Toc186502850)

[Process flow/Representation 9](#_Toc186502851)

[Design models 10](#_Toc186502852)

[Class Diagram 10](#_Toc186502853)

[Sequence Diagram’ 11](#_Toc186502854)

[State Transition Diagram 12](#_Toc186502855)

[Data Design 13](#_Toc186502856)

[Data Storage Needs 13](#_Toc186502857)

[Data Storage Approach 13](#_Toc186502858)

[What We Won’t Store 13](#_Toc186502859)

[Algorithm & Implementation 13](#_Toc186502860)

[Human interface design 16](#_Toc186502861)

[Screen images 16](#_Toc186502862)

[**Appendix I** 18](#_Toc186502863)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

**Application Evaluation History**

|  |  |
| --- | --- |
| **Comments (by committee)**  **\*include the ones given at scope time both in doc and presentation** | **Action Taken** |
|  |  |
|  |  |

**Supervised by**

**Dr. Mubashir Ahmad**

Signature\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Introduction

The liver tumor segmentation project automates tumor segmentation in 2D CT scans using nnFormer architecture. Due to computational limitations, the project processes 2D images extracted as images from 3D CT scans. nnFormer integrates convolution layers and attention mechanisms, effectively combining local and global contexts to enhance segmentation accuracy and efficiency.

The scope of the project includes the following modules:

1. **Data Preprocessing**: Conversion of segmentation.nii and volume.nii files into 2D images of 512x512 resolution, excluding patches without relevant information.
2. **Segmentation Models**: Two distinct nnFormer models adapted for 2D inputs—one for liver extraction and another for tumor segmentation.
3. **Frontend Interface**: A professional yet simple UI for doctors, built with React and Material UI, enabling image upload, segmented result display, and analytical visualization.
4. **Analytical Insights**: Statistical analysis of tumor characteristics to aid in medical decision-making.

This project uses the LITS17 dataset and emphasizes improving segmentation outcomes compared to architectures like CNN-based UNet, TransUNet, and SwinUNet.

# Design methodology and software process model

Design Methodology  
Our project follows an Object-Oriented Programming (OOP) approach. The use of OOP is justified by the modular and reusable nature of its design principles, which align with the complexity of the system. Key components, such as data preprocessing, model training, and frontend development, are encapsulated into distinct classes and objects. This methodology supports better abstraction, enhances code readability, and simplifies the integration of different modules. The OOP approach also allows us to efficiently manage the interactions between the nnFormer-based segmentation models and the frontend interface.

Software Process Model

We are adopting the **Incremental Development Model** for our project. This choice is justified by the modular nature of our system and the iterative improvements required to handle resource constraints effectively. Each module—data preprocessing, model training, and UI development—is treated as a separate increment, allowing us to deliver functional components progressively.

1. **Flexibility**: Allows iterative refinement of nnFormer adaptations, especially for 2D input handling and performance optimization.
2. **Risk Management**: Early delivery of functional modules, such as preprocessing or frontend interfaces, helps identify and address risks early.
3. **Resource Optimization**: Dividing the project into increments helps manage limited computational resources effectively during model training and testing.
4. **Feedback Integration**: Each increment provides an opportunity to incorporate feedback from stakeholders or test outcomes.

# System overview

The liver tumor segmentation system is designed to automate the segmentation of liver and tumor regions in 3D CT scans. The core objective of the system is to enhance segmentation accuracy and efficiency by leveraging the nnFormer architecture, which combines convolutional layers with self-attention mechanisms. This hybrid approach captures both local and global context, improving the overall segmentation performance. Due to computational constraints, the system processes 2D image patches extracted from 3D CT scans rather than utilizing full 3D volumes.

## Functionality

The system operates through a two-phase process: preprocessing and segmentation. The preprocessing phase involves converting the 3D CT scans into 2D images (512x512). These patches are then input into the nnFormer model for segmentation, generating masks that highlight the liver and tumor regions. These segmented outputs are visualized and analyzed through a user-friendly frontend.

1. **Image Upload and Preprocessing**: Users can upload 3D CT scans, which are pre-processed into 2D patches to serve as input for the model.
2. **Segmentation**: The pre-processed 2D patches are fed into the nnFormer architecture, which segments the liver and tumor regions.
3. **Visualization**: The segmented output, including tumor statistics such as volume and area, is displayed to the user for clinical analysis.
4. **Analytics**: Tumor-related metrics such as size, volume, and location are extracted and presented to assist in the diagnosis process.

## Context

This system is intended to improve the efficiency and accuracy of medical image segmentation by automating the process of tumor identification in liver CT scans. By leveraging advanced techniques such as convolution layers and attention mechanisms, the system is able to provide accurate results even with the challenges of computational limitations. The system will be tested using the LITS17 dataset, a well-known benchmark in liver tumor segmentation research.

## Design

The system follows a modular design approach, ensuring separation of concerns between model training, data preprocessing, and frontend development. This modularity makes the system maintainable and scalable. The nnFormer architecture incorporates both convolutional layers and self-attention mechanisms to handle both local and global contexts effectively. The frontend is built using **React** and **Tailwind**, designed to provide an intuitive, professional interface suitable for medical applications, following Nielsen’s usability principles.

## Background Information

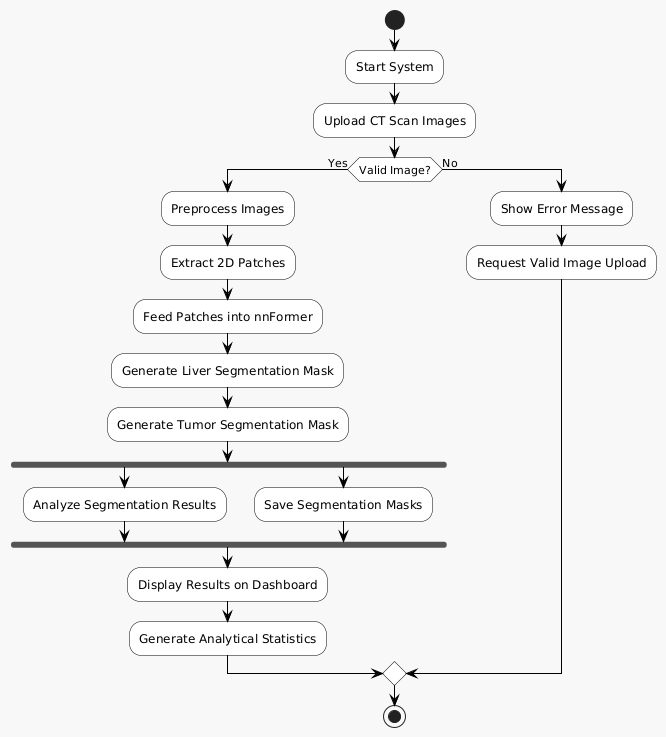
* **nnFormer**: nnFormer is an advanced deep learning architecture specifically designed for medical image segmentation. It integrates convolutional layers with attention mechanisms, enabling the model to capture both local features and global context from the input images.
* **LITS17 Dataset**: The LITS17 dataset consists of 3D CT images annotated with liver and tumor masks, widely used for benchmarking medical segmentation models. It provides a standard for evaluating the performance of liver and tumor segmentation systems.

# Architectural design

A diagram of a machine

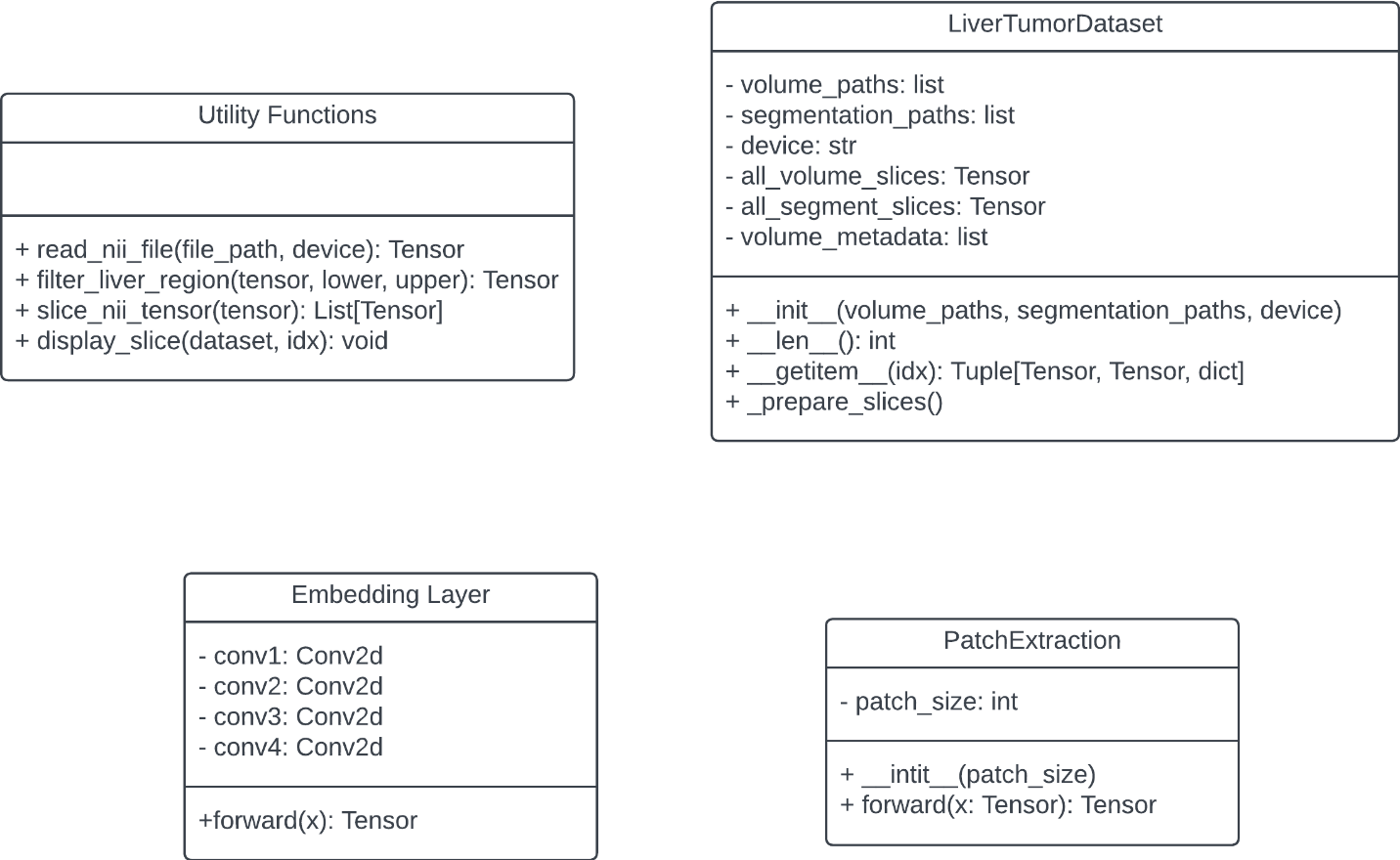
Description automatically generated

## Process flow/Representation



# Design models

## Class Diagram



## Sequence Diagram’

A diagram of a project

Description automatically generated

## State Transition Diagram

A diagram of a process

Description automatically generated

# Data Design

In our liver tumor segmentation system, we don’t need to store the raw CT images from the LITS17 dataset because they are already available. Instead, we focus on storing the processed results and metadata.

## Data Storage Needs

1. **Processed Segmentation Results**:
   * After segmenting the liver and tumor, we’ll store the segmented images and masks (e.g., nii, .png) for later use.
2. **Metadata**:
   * This includes info like image dimensions, format, and whether the image has been processed or not. We’ll store this in simple formats like **JSON** or **CSV**.
3. **Session Data (optional)**:
   * If needed, we might store session-related info like user ID, uploaded image names, and links to processed results.

## Data Storage Approach

* **File-based Storage**: We’ll store processed images and masks as files with clear names linking them to the original images.
* **Metadata Storage**: Metadata can be stored in **JSON** or **CSV** files, or in a small database like **SQLite**.

## What We Won’t Store

1. **Raw CT Images**: We don’t need to store the LITS17 dataset again as it’s already available.
2. **Intermediate Data**: Any temporary data from the segmentation process will not be stored.

# Algorithm & Implementation

In this section, we explain how each part of the system works, step by step. The main components include functions for loading, processing, and segmenting CT images, as well as storing the results.

#### 1. Load and Preprocess Image

This function loads a CT image, resizes it to the required size, and normalizes the pixel values so the image is ready for segmentation.

##### Pseudo-code:

Function LoadAndPreprocessImage(image\_path):

Load image from image\_path

Resize image to 512x512

Normalize pixel values to range 0-1

Return preprocessed\_image

* **Inputs**: image\_path (path to the CT image).
* **Outputs**: preprocessed\_image (image ready for segmentation).

#### 2. Liver Segmentation

This function uses the trained model to segment the liver region from the CT image. It processes the image and creates a mask of the liver.

##### Pseudo-code:

Function SegmentLiver(preprocessed\_image):

Run preprocessed\_image through the liver segmentation model

Get liver mask

Return liver\_mask

* **Inputs**: preprocessed\_image (the processed CT image).
* **Outputs**: liver\_mask (mask showing liver region).

#### 3. Tumor Segmentation

After the liver is segmented, this function segments the tumor region inside the liver using a trained model. It focuses only on the area identified as the liver.

##### Pseudo-code:

Function SegmentTumor(preprocessed\_image, liver\_mask):

Focus on liver region using liver\_mask

Run it through the tumor segmentation model

Get tumor mask

Return tumor\_mask

* **Inputs**: preprocessed\_image (processed CT image), liver\_mask (mask of the liver).
* **Outputs**: tumor\_mask (mask showing tumor region).

#### 4. Save Results

This function saves the liver and tumor masks along with any other information (like image size or processing time) to files.

##### Pseudo-code:

Function SaveResults(liver\_mask, tumor\_mask, metadata):

Save liver\_mask as file

Save tumor\_mask as file

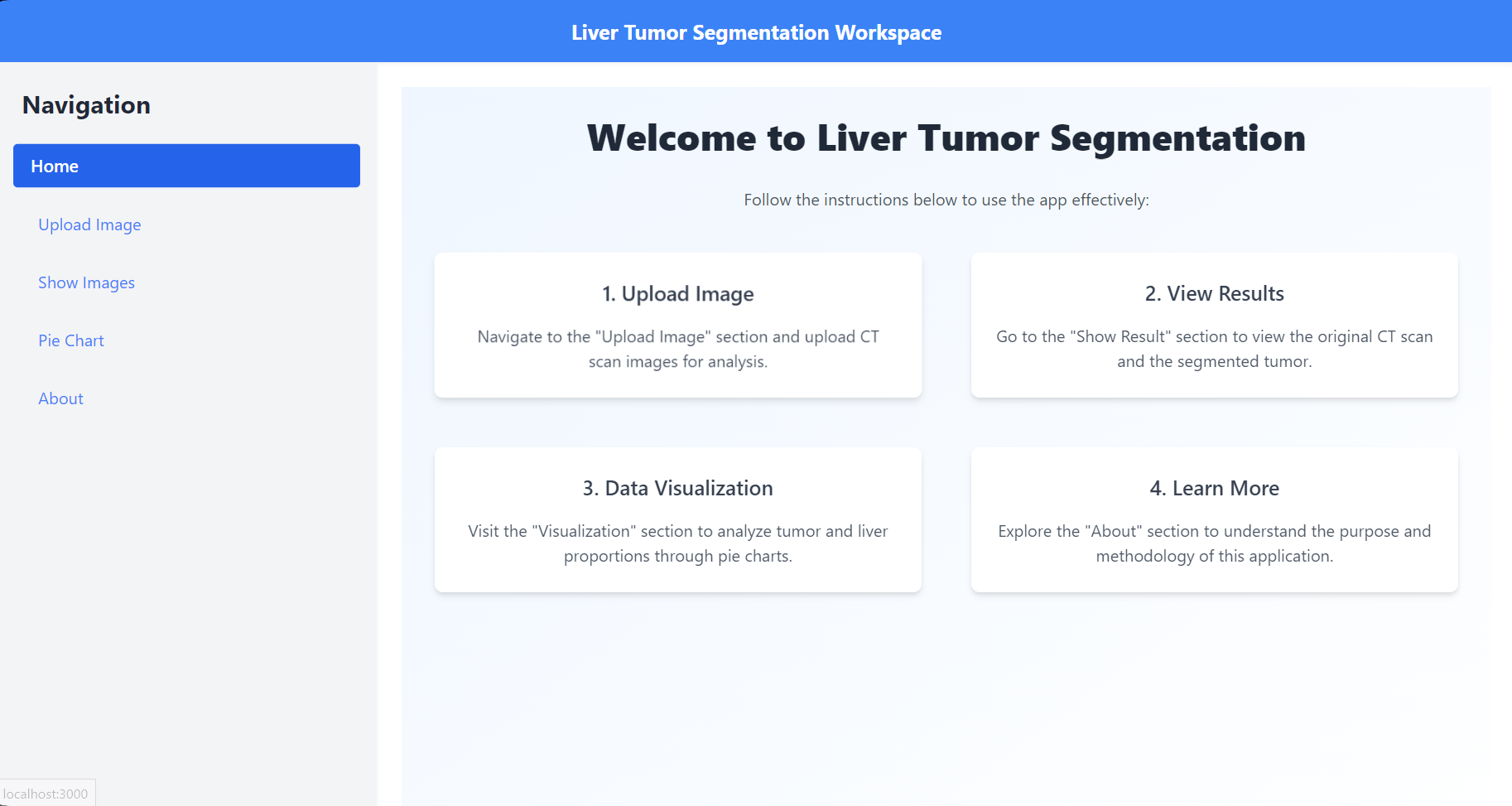
Save metadata (image size, processing time)

* **Inputs**: liver\_mask, tumor\_mask (segmented masks), metadata (additional info).
* **Outputs**: Stored masks and metadata.

# Human interface design

From the user’s perspective, the liver tumor segmentation system provides an intuitive, easy-to-navigate interface designed for medical professionals, such as radiologists or doctors.

## Screen images



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a graph

Description automatically generated

A screenshot of a computer

Description automatically generated

**Appendix I**

[1] <https://www.kaggle.com/datasets/javariatahir/litstrain-val>

[2] <https://arxiv.org/pdf/2109.03201v6>

[3] <https://pytorch.org/docs/stable/index.html>