**Final Year Project SuperScan**

**Lesion Detection and Analysis in CT Scan**

**FYP Team**

### Slahuddin Chaudhary (16i-0060)

### Abdul Masood (16i-0312) Hajra Khan (16i-0326)

**Supervised by**

### Dr. Muhammad Adnan Tariq

### Department of Computer Science

### National University of Computer and Emerging Sciences Islamabad, Pakistan

2019

**Contents**

[Chapter 1 - Introduction 1](#_bookmark0)

[Chapter 2 – Literature Review 1](#_bookmark3)

[Chapter 3 – Project Vision 2](#_bookmark9)

* 1. [Problem Statement 2](#_bookmark10)
  2. [Business Opportunity 2](#_bookmark11)
  3. [Objectives 2](#_bookmark12)
  4. [Project Scope 3](#_bookmark13)

3.5 Constraints 3

3.6 Stakeholders Description 4

3.6.1 Stakeholders Summary 4

3.6.2 Key High Level Goals and Problems of Stakeholders 4

[Chapter 4 – Software Requirement Specifications 6](#_bookmark14)

* 1. [List of Features 6](#_bookmark15)
  2. [Functional Requirements 6](#_bookmark16)
  3. [Quality Attributes 6](#_bookmark17)
  4. [Non-functional Requirements 7](#_bookmark18)

[Chapter 5 – High Level Use cases 8](#_bookmark21)

5.1  [Use case Diagram 8](#_bookmark15)

[Chapter 6 – Iteration Plan 10](#_bookmark21)

[**Chapter 7 – Iteration 1 11**](#_bookmark21)

7.1 [Expanded Use Cases 11](#_bookmark15)

7.2 [Activity Diagram 14](#_bookmark15)

7.3 [Domain Model 15](#_bookmark15)

7.4 [System Sequence Diagram 16](#_bookmark15)

7.5 [Operation Contracts 17](#_bookmark15)

7.6 [Sequence Diagram 20](#_bookmark15)

7.7 [Class Diagram 21](#_bookmark15)

7.8 [Project Process Flow Diagram 22](#_bookmark15)

[**Chapter 8 – Implementation Details 23**](#_bookmark21)

[**Chapter 9 – User Manual 25**](#_bookmark21)

[References 25](#_bookmark22)

# Chapter 1 - Introduction

## 

The purpose of this report/document is to collect, analyze and define high-level needs and features of detection and analysis of lesion in CT scans. It focuses on the capabilities needed by the stakeholders and the target users, and why these needs exist. The details of how an application to detect and analyze lesion in CT scans fulfills these needs are detailed in the use-case and software requirement specifications.

# Chapter 2 – Literature Review

Various approaches have been applied to detect anomalies, with certain particularities in the medical image scenario, linked to other terms: content-based image retrieval, pattern recognition, classification, segmentation, outlier detection, image mining, as well as computer-assisted diagnosis, and computer aided surgery[[1]](#footnote-1). Computed Tomography (CT) scanning is an extensively used technique in Medical Diagnostic Imaging to detect and monitor conditions such as tumors, heart diseases, lung nodules and liver masses etc. Internal injuries counted in, CT scans are also a precursor to planning medical, surgical or radiation treatment. Lesion detection from computed tomography (CT) scans is an important task in medical imaging analysis. It is still very challenging due to similar appearances (e.g. intensity and texture) between lesions and other tissues, making it especially difficult to develop a universal lesion detector. Tumor measurements on CT scans, or the appearance of new lesions on any of a variety of imaging studies, are key determinants for assessing progression-free-survival as an endpoint in many clinical trials of therapies for solid tumors. Test-retest tumor measurement reproducibility may vary considerably across serial scans on the same patient unless rigorous attention is paid to standardization of image acquisition parameters and unless trained, experienced observers using validated objective methods make measurements[[2]](#footnote-2). Target lesion selection also must be done with care to choose lesions that are or will be reproducibly measurable. Likewise, new lesions will be missed or misinterpreted on follow-up imaging studies unless those imaging studies are obtained using techniques suitable for detecting early, small lesions. Reader variability is clearly a major component of the problem.

# Chapter 3 – Project Vision

## Problem Statement

While computerized tomography (CT) may have been the ﬁrst-imaging tool to study human body, but it has not yet been implemented into clinical decision-making process for diagnosis of numerous diseases. However, the gravity of the issue at hand enunciates that it takes an average of about an hour for a radiologist to deduce astute conclusions, or to write a report of a CT scan. Thereby, it takes about 24 hours for the report to get to the patient, regardless of the extremity of conditions. In the meanwhile, the diagnostics industry suffers inefficient utilization of CT machines due to long analysis processes of numerous similar CT images, that too manually. In the wake of said hurdles, the development of automation for detecting and analyzing lesions in CT scan has become imperative**.**

## Business Opportunity

Manual report generation is a time consuming process as mentioned above but it has a high demand for diagnosis. According to the authentic stats obtained from FCPS resident at PIMS pronounce that more than 50% of the CT scans are not formally labelled with reports because of lack of time and skilled human resources. It takes at minimum 5 days in major public health institutes like PIMS to give reports to patients. The amount of human resources required to bridge the gap isn’t filling up with the amount of recruitments been done. Instead, the gap has been increasing for the past 10 years. Therefore, we propose an automation tool to speed up the process. It is cost effective to use an automation toll instead of hiring human resources, which aren’t readily available when required.

## Objectives

The proposed system has the following objectives:

1. It aims to detect the exact location of the lesion in a patient’s body.
2. It will further proceed by analyzing the type of lesion visible in the image.
3. The analysis process will articulate special features, demonstrating the shape, dimensions, surface structure, and Volume-Weight density (standard deviations) of the lesion.
4. It will also validate the level of the lesion with respect to vertebrae, and its extent to and from the vertebrae.
5. It will forewarn the radiologists if the lesion has involved any organs.

## Project Scope

**Our proposed system accentuates to save the time and resources of radiologists by automatically detecting and analyzing lesions in a CT Image**. *Python, PyDicom, GDMC, ML & AI, and Electron JS* provide the viability to make this proposition feasible. Annotating further, we assert that this project does not proclaim to replace radiologists; rather it intends to make their job easier and efficient. It is quite evident that our proposal pronounces how science and technology correspond to each other, and how we plan on taking a step forward in advancing the scientific technology further. Our proposed solution does not portray a complete takeover of technology; rather it represents how we can play our part to aid medical practitioners, predominantly radiologists, in quickly detecting and analyzing lesions in CT scan images, with special features added to automatic analysis. We aim to reduce their time and effort by (whatever you guys claim). The faster a lesion is identified and analyzed accurately, the sooner the treatment would begin. Hence, we expect medical treatments and processes to be quicker than before, giving less time to the diseases to move from initial to chronic stages.

## Constraints

The constraints for our system are as follows:

### *Data collection*

### 

On data collection, we have gathered around 500 images from Anmol Hospital Lahore and Alrazi labs. The data from PIMS is still in collection phase. It is expected to give us around 3000 brain CT images by the end of October.

For now, we are working on detection and segmentation on a dataset from NIH which is available on NIH website with 10,000 images converted to png format. We have accomplished bounding box segmentation so far, which has segmented 30,000 lesions until now.

### *Data Labelling*

Though constructive, it is costly and time consuming.

### *Hardware resources*

The issue at hand is the unavailability of adequate resources for memory to store the CT scan image data, and to train model for detection and analysis of lesion in CT scan.

### *3D visualization*

Illuminating the lesion boundary is challenging, because one uploaded folder has numerous images. Successively, we stack slices to generate 3D matrix which is used to detect lesion then analyze it. The tough part again is to produce 3D visuals.

## Stakeholders Description

## 3.6.1 Stakeholders Summary

Following are the stakeholders of our system:

1. Radiologists- own their clinics, our system, thereby, would be beneficial for them.
2. Hospitals- hospital setups would run more efficiently and smoothly, considering urgent diagnostic reports.
3. Patients- delayed reports summon to delayed diagnosis, therefore, we can take a high moral ground by mentioning the patient’s plight.
4. Laboratories- it would be easier for laboratories to deduce reliable results in the absence of a radiologist.
5. CT scan machine manufacturing companies- the machines being costly limit the generation of more CT scans while other reports are being studied. One way or another, this is wastage of time and the CT scan machines too.

## 3.6.2 Key High Level Goals and Problems of Stakeholders

1. Laboratories and radiologists are not tech savvy, neither IT literate; hence, they require a system which is easier to deal with in a lesser span of time.
2. Utilization of resources
3. CT machine wastage due to time taken during report generation, radiologists face the same problem.

|  |  |  |
| --- | --- | --- |
| **High Level Goal(s)** | **Goal Priority** | **Current Problem(s) and Cause(s)** |
| Durable and Reliable | **High** | Not every lab can be trusted with producing reliable results. Our system tends to affirm durability and reliability unlike the improper machinery and instruments. |
| Flexible | **High** | The system needs to be flexible and adaptive to the rapidly changing technological trends. Currently, there is no technological feature in the system. |
| less time consumption | **High** | Currently, manual report generation takes ample amount of time. Our system will aid to reduce the time to a high extent. |

## 

# Chapter 4 – Software Requirement Specifications

## List of Features

* Lesion detection
* Analysis of lesion
  + Shape
  + Dimensions
  + Level with respect to vertebrae
  + Extent to and from vertebrae
  + Surface structure
  + Volume-Weight density (standard deviations)
  + Organs involved

## Functional Requirements

The proposed system has the following functional requirements:

1. Detect the exact location of the lesion in a patient’s body.
2. Analyze the type of lesion visible in the image.
3. Articulate special features, demonstrating the shape, dimensions, surface structure, and Volume-Weight density (standard deviations) of the lesion.
4. Validate the level of the lesion with respect to vertebrae, and its extent to and from the vertebrae.
5. Forewarn the radiologists if the lesion has involved any organs.

## Quality Attributes

* Correctness
* Adequacy
* Robustness
* Readability
* Efficiency
* Portability

## Non-functional Requirements

### ***Usability***

1. Graphical User Interface

* The system shall provide use of icons and toolbars.
* The system shall provide an authentic digital image for each uploaded folder.
* The 3D visualization shall be of a light color scheme in order to make the images more visible.

1. Accessibility

* The system shall be compatible with all devices.
* The system shall provide access on different hardware platforms.
* The system shall provide multi language support.

### ***Reliability***

1. Availability

* System shall not be down for maintenance for a time period longer than an hour.
* Users shall be informed at the time of maintenance.

### ***Performance***

* Responsiveness

### ***Supportability***

* Updates shall be convenient to make.

# Chapter 5 – High Level Use cases

## 5.1 Use case diagram

## 

## UC-01:

**Use Case Name**: Upload Folder

**Scope**: SuperScan

**Level**: User Level

**Type**: Primary

**Primary Actors**: Radiologist

**Description**: This module deals with the radiologist uploading the folder containing slices of an image. If the uploaded folder is invalid (incompatible format), the system disapproves to process it further and prompts the user to upload a valid one. Once a valid folder is upload, the image folder is read successfully and loaded in memory.

## UC-02:

**Use Case Name**: Detect Lesion

**Scope**: SuperScan

**Level**: User Level

**Type**: Primary

**Primary Actors**: Radiologist

**Description**: This module deals with the radiologist viewing the lesion detected in the folder uploaded. If the uploaded folder is invalid (incompatible format), the system disapproves to process it further and prompts the user to upload a valid one. Once a valid folder is upload, the system detects the lesion and displays it on the screen.

## UC-03:

**Use Case Name**: Create 3D Visualization

**Scope**: SuperScan

**Level**: User Level

**Type**: Primary

**Primary Actors**: Radiologist

**Description**: This module deals with the radiologist viewing the 3D visualization of the folder uploaded. If the uploaded folder is invalid (incompatible format), the system disapproves to process it further and prompts the user to upload a valid one. Once a valid folder is upload, the system creates 3D visualization and displays it on the screen.

# Chapter 6 – Iteration Plan

# Chapter 7 – Iteration 1

## 7.1 Expanded Use cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-1.1 | | | | |
| **Use Case Name:** | Upload Folder | | | | |
| Created By: | | | Hajra Khan | Last Updated By: | Hajra Khan |
| Date Created: | | | September 27, 2019 | Date Last Updated: | September 27, 2019 |
| **Actors:** | | Radiologist | | | |
| **Description:** | | The user shall upload the folder containing slices of an image. | | | |
| **Trigger:** | | User uploads the folder containing slices of an image. | | | |
| **Preconditions:** | | User has accessed the application. | | | |
| **Post conditions:** | | User shall successfully upload the folder. | | | |
| **Normal Flow:** | | 1. User shall press the Upload Folder button. 2. User shall then upload the folder. 3. The image folder was read successfully and loaded in memory. | | | |
| **Alternative Flows:** | | 2a. In step 1 of the normal flow, if user has not uploaded the folder   * + - 1. System shall prompt user to upload the folder. | | | |
| **Exceptions:** | | 2a. In step 2 of the normal flow, if the user uploads an invalid (incompatible format) folder   1. System disapproves to process the folder 2. System shall prompt the user to upload a valid folder 3. User re-uploads the folder   Use Case resumes on step 3 of normal flow | | | |
| **Includes:** | | N/A | | | |
| **Special Requirements:** | | Touch screen UI on a large fiat panel monitor. Text must be visible from 1 meter. | | | |
| **Assumptions:** | | N/A | | | |
| **Notes and Issues:** | | What will be the size of the uploaded folder? | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-1.2 | | | | |
| **Use Case Name:** | Detect Lesion | | | | |
| Created By: | | | Hajra Khan | Last Updated By: | Hajra Khan |
| Date Created: | | | September 27, 2019 | Date Last Updated: | September 27, 2019 |
| **Actors:** | | Radiologist | | | |
| **Description:** | | The system shall detect lesion in the uploaded folder (containing slices of an image) by the radiologist. | | | |
| **Trigger:** | | User selects the detect lesion option from the navigation bar. | | | |
| **Preconditions:** | | User has already uploaded the folder. | | | |
| **Post conditions:** | | User shall successfully view the lesion detected by the system. | | | |
| **Normal Flow:** | | 1. User shall upload the folder.  2. User shall press the Detect Lesion button.  3. System shall display the lesion in the uploaded folder. | | | |
| **Alternative Flows:** | | 2a. In step 1 of the normal flow, if user has not uploaded the folder and pressed the Detect Lesion button  1. System shall prompt user to upload the folder.  2. Use case shall resume to step 2 of normal flow | | | |
| **Exceptions:** | | 2a. In step 2 of the normal flow, if the user uploads an invalid (incompatible format) folder  1. System disapproves to process the folder  2. System shall prompt the user to upload a valid folder  3. User re-uploads the folder  Use Case resumes on step 3 of normal flow | | | |
| **Includes:** | | Upload Folder | | | |
| **Special Requirements:** | | Touch screen UI on a large fiat panel monitor. Text must be visible from 1 meter. | | | |
| **Assumptions:** | | N/A | | | |
| **Notes and Issues:** | | What will be the size of the uploaded folder? | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-1.3 | | | | |
| **Use Case Name:** | Create 3D Visualization | | | | |
| Created By: | | | Hajra Khan | Last Updated By: | Hajra Khan |
| Date Created: | | | September 27, 2019 | Date Last Updated: | September 27, 2019 |
| **Actors:** | | Radiologist | | | |
| **Description:** | | The system shall create 3D visualization of the uploaded folder (containing slices of an image) by the radiologist. | | | |
| **Trigger:** | | User selects the 3D visualization option from the navigation bar. | | | |
| **Preconditions:** | | User has already uploaded the folder. | | | |
| **Post conditions:** | | User shall successfully view the 3D visualization created by the system. | | | |
| **Normal Flow:** | | 1. User shall upload the folder.  2. User shall press the 3D visualization button.  3. System shall display 3D visualization of the uploaded folder. | | | |
| **Alternative Flows:** | | 2a. In step 1 of the normal flow, if user has not uploaded the folder and pressed the 3D visualization button  1. System shall prompt user to upload the folder.  2.Use case shall resume to step 2 of normal flow | | | |
| **Exceptions:** | | 2a. In step 2 of the normal flow, if the user uploads an invalid (incompatible format) folder   1. System disapproves to process the folder 2. System shall prompt the user to upload a valid folder 3. User re-uploads the folder   Use Case resumes on step 3 of normal flow | | | |
| **Includes:** | | Upload Folder | | | |
| **Special Requirements:** | | Touch screen UI on a large fiat panel monitor. Text must be visible from 1 meter. | | | |
| **Assumptions:** | | N/A | | | |
| **Notes and Issues:** | | What will be the size of the uploaded folder? | | | |

## 7.2 Activity Diagram

## 

## 7.3 Domain Model

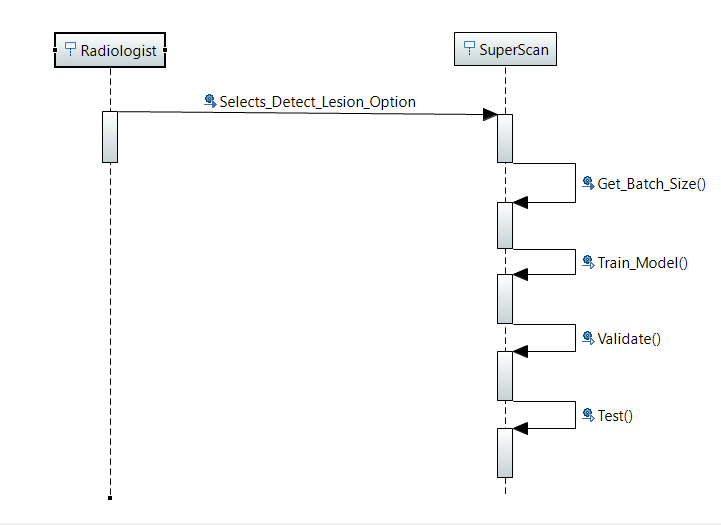
## 

## 7.4 System Sequence Diagram

### ***Data Preprocessing***

## 

### ***Detect Lesion***

****

### 

### ***3D Visualization***

### 

## 7.5 Operation Contracts

**Contract CO1:**

**Operation:** UploadFolder()

**Responsibility:** Upload the folder containing slices of an images

**Type:** System

**Cross Reference:** N/A

**Pre-Conditions:** User has access to the application

**Post-Conditions:**

* + The slice names were associated with the image\_list\_attribute.
  + The pixel data of each slice was assigned to 2D arrays.
  + All 2D arrays were stacked in same order as slices to form a 3D matrix.

**Contract CO2:**

**Operation:** List\_images()

**Responsibility:** List images given in the uploaded folder

**Type:** System

**Cross Reference:** UC01, UploadFolder()

**Pre-Conditions:** User has uploaded a valid folder

**Post-Conditions:**

* Read instance was created successfully.
* It was able to form associations image-set names.

**Contract CO3:**

**Operation:** Read\_image()

**Responsibility:** Read images given in the uploaded folder

**Type:** System

**Cross Reference:** UC02, List\_images()

**Pre-Conditions:** The images in the uploaded folder have been listed

**Post-Conditions:**

* Read instance was associated with pixel data.
* Each instance of 2D array was deleted after storing it in 3d matrix.

**Contract CO4:**

**Operation:** Stack\_slices()

**Responsibility:** Stack slices of the images given in the uploaded folder to form a 3D matrix

**Type:** System

**Cross Reference:** UC03, Read\_image()

**Pre-Conditions:** The images in the uploaded folder have been read

**Post-Conditions:**

* The preprocessing instance was created successfully.
* It was able to generate and associate with 2D matrices.
* It was able to delete each 2D matrix instance as soon as it was passed to 3D matrix.

**Contract CO5:**

**Operation:** Generate\_Pixel\_Matrix()

**Responsibility:** Generate pixel matrix of the uploaded folder

**Type:** System

**Cross Reference:** UC04, Stack\_slices()

**Pre-Conditions:** Slices of the images in the uploaded folder have been stacked

**Post-Conditions:**

* 3D matrix instances for numpy were created and associated with pre-processing instance.
* Each 3D matrix was deleted after it passed its data on to cnn.

**Contract CO6:**

**Operation:** Get\_batch()

**Responsibility:** Make a batch of the images to be read, given in the uploaded folder

**Type:** System

**Cross Reference:** UC01, UploadFolder(); UC04, Generate\_Pixel\_Matrix()

**Pre-Conditions:** User has uploaded a valid folder and generated a pixel matrix

**Post-Conditions:**

* N instances of 3D Matrices were created and populated with data received from preprocessing module.
* The instances of N 3D NumPy arrays containing pixel data of unique image-set were received and associated successfully with the N-sized batch.
* Each instance was successfully destroyed to save memory as soon as the corresponding 3D matrix was fed to the cnn model.

**Contract CO7:**

**Operation:** Train\_model()

**Responsibility:** Train the model for all the work being done previously

**Cross Reference:** UC06, Get\_batch()

**Pre-Conditions:** Batch of the images that have to be read has been made.

**Post-Conditions:**

* An instance of the cnn model was created successfully.
* The cnn instance was successfully associated with each instance of 3D numpy arrays which it received as input.
* Each association was destroyed as soon as the it's corresponding batch was processed.
* The cnn instance was able to save the tuned parameters and was destroyed safely.

**Contract CO8:**

**Operation:** Validation()

**Responsibility:** Validate the previous work done in training the model.

**Type:** System

**Cross Reference:** UC07, Train\_model()

**Pre-Conditions:** Model has been trained.

**Post-Conditions:**

* Validation instance was created successfully.
* Its associations with cnn instance were created and it was able to fine tune parameters of cnn.
* Validation instance was deleted successfully.

**Contract CO9:**

**Operation:** Create3DVisualization ()

**Responsibility:** Create 3D visualization of the uploaded folder

**Type:** System

**Cross Reference:** UC01, UploadFolder()

**Pre-Conditions:** User has uploaded a valid folder

**Post-Conditions:**

* + The upload folder instance was successfully associated with 3D\_visualization instance
  + The analyzed\_image instance was successfully associated with 3D\_visualization instance.
  + The pixel\_data in visualize\_data instance was copied and saved for future reference after output was generated.

## 7.6 Sequence Diagram

## 

## 7.7 Class Diagram

## 

## 7.8 Project Process Flow Diagram

## 

# Chapter 8 – Implementation Details

## Prelude:

The product consists of four distinct modules:

1. Data Pre-Processing
2. Lesion Detection
3. Lesion Analysis
4. 3D Visualization
5. Report Generation

## Abstract:

Data Pre-Processing module reads and prepares pixel-data from storage to be fed to a deep learning model. The Lesion detection module takes data and trains a deep learning model on it to be able to predict if a test image contains any lesions or not. If any lesions are found, the Lesion Analysis module runs image-processing to analyze the shape, surface-type, location and roots of lesion if there were any. 3D visualization module sets the values of pixels at the boundary of the lesion to maximum aloud value – 2048. It saves a copy of modified CT – image and opens it with Node.js module which gives a 3D preview of the image, ready to be manipulated by the user for a deeper look. This image would show lesion outlined and illuminated. The image can also be previewed in 3D without detecting, analyzing and outlining lesion. Finally, the results of Lesion Analysis are formatted and made ready for printing by the Report Generation Module.

## Data Pre-Processing:

The module starts with reading a folder containing subfolders of image-sets, each having a unique id. An image-set is one Computed Tomography scan consisting of 100 – 500 slices; each slice being 512 x 512 two dimensional image in dicom format with extension dcm. [PyDicom](https://pydicom.github.io/pydicom/stable/api_ref.html) is used for reading image-sets one-by-one. We extract pixel-data of each slice in two dimensional [NumPy](https://numpy.org/devdocs/) arrays to be stacked one upon another. [GDCM](http://gdcm.sourceforge.net/) makes it possible to stack two dimensional pixel-data to form a 3D matrix with pixels equispaced from all six of its immediate neighbors in three-dimensional space. Pre-Processing module reads a batch of labeled image-sets to feed the Convolution Neural Network in Lesion Detection Module. Labels come from a folder containing reports written by professional radiologists against each CT scan – termed herein as an image-set. The reports are extracted from PDF files using [pdttotext](https://pypi.org/project/pdftotext/) and tokenized to extract meaningful text to be used for labeling input for CNN.

## Lesion Detection:

Lesion detection module consists of a convolution neural network with 3Dconvolution layers, max-Pooling layers, drop-out layers and a fully connected layer to classify data. The model is expected to predict lesions with more than ninety percent accuracy so it is subject to major architectural changes.

## Lesion Analysis:

The lesion detection module passes the image-set to lesion analysis module if lesions were found. The Lesion Analysis module segments the lesion using semantic segmentation. The pixel values of lesion that are not expected to be found in the given region make it obvious for the deep learning model to identify it as a lesion. Further, deep learning techniques are yet to be analyzed in order to determine the properties under lens.

## 3D Visualization:

3D visualization module takes an image-set as input, and creates a 3D preview of the image-set to be manipulated by the user for a deeper look. The visualization module uses Node.js module med3web to generate a 3D view in web Browser. The interface consists of a volume rendering section that allows user to change intensity under focus so that different parts like bones and organs can be viewed. In case where lesion was analyzed before generating 3D visualization, the module shows lesion illuminated since the pixels on its boundary are set to maximum values.

## Report Generation:

The lesion analysis module sends its output to the report generation module. It formats the strings in meaningful human readable form by arranging it under relevant titles and produces a pandas data-frame. This data-frame can be saved to disc in csv format or can directly be sent for printing.

# Chapter 9 – User Manual

The purpose of user manual is to assist clients with functionality and capabilities of the system. The user manual will help the clients in how to manage system easily. The manual shall be interactive and easy to read and understand.

# References

Taboada-Crispi, Alberto, et al. "Anomaly detection in medical image analysis." *Handbook of Research on Advanced Techniques in Diagnostic Imaging and Biomedical Applications*. IGI Global, 2009. 426-446.

Sullivan, Daniel Carl, Lawrence H. Schwartz, and Binsheng Zhao. "The imaging viewpoint: how imaging affects determination of progression-free survival." (2013): 2621-2628.

1. Taboada-Crispi, Alberto, et al. "Anomaly detection in medical image analysis." *Handbook of Research on Advanced Techniques in Diagnostic Imaging and Biomedical Applications*. IGI Global, 2009. 426-446. [↑](#footnote-ref-1)
2. Sullivan, Daniel Carl, Lawrence H. Schwartz, and Binsheng Zhao. "The imaging viewpoint: how imaging affects determination of progression-free survival." (2013): 2621-2628. [↑](#footnote-ref-2)