A PROJECT REPORT ON

Implementation of Lung Cancer Nodule Detection

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

BACHELOR OF ENGINEERING

In

COMPUTER ENGINEERING

Of

SAVITRIBAI PHULE PUNE UNIVERSITY

By

Patil Sarang	**BECB005**
Maramwar Siddharaj	**BECA028**
Gawande Mahesh	**BECA018**
Damkondwar Shubham	**BECB028**

Under the guidance of
Prof. Neha Jamdar

SINHGAD TECHNICAL EDUCATION SOCIETY -PUNE
तमसो मा ज्योतिर्गमय
Sinhgad Institutes

DEPARTMENT OF COMPUTER ENGINEERING

SKN SINHGAD INSTITUTE OF TECHNOLOGY AND SCIENCE, LONAVALA

Accredited by NAAC

2019-20

Sinhgad Techical Education Society,
Department of Computer Engineering,
SKN Sinhgad Intitute of Technology Kusgaon
Bk. Lonavala

Sinhgad Institutes

Date:

CERTIFICATE

This is to certify that the project report entitled

“ Implementation of Lung Cancer Nodule Detection”

Submitted by

Patil Sarang	**BECB005**
Maramwar Siddharaj	**BECA028**
Gawande Mahesh	**BECA018**
Damkondwar Shubham	**BECB028**

is a bonafide work carried out by him/her under the supervision of Prof. Neha Jamdar and it is approved for the partial fulfillment of the requirements of Savitribai Phule Pune University, Pune for the award of the degree of Bachelor of Engineering (Computer Engineering) during the year 2017-18.

Prof. Neha Jamdar
Internal Guide

Prof. G. M. Kadam
Head
Department of Computer Engineering

Dr. M. S. Rohokale
Principal
SKN Sinhgad Institute of Technology and Science

Acknowledgement

Working on this project is one of the wonderful and existing experiences in our lives. This report not only bears the testimony of existing efforts but also reflects the cooperation, help and guidance, which we received time to time. It is obvious but that we acknowledge them for their help without which the report would never have been completed.

We would like to express our deep gratitude to Prof. Neha Jamdar for her profound guidance and support which helped us understanding the nuances of the project and make it work. We are thankful to her for her timely suggestions which helped us a lot in completion of this report.

Our heart felt thankful for our Head of Department Prof. G. M. Kadam who has been a consistent motivator throughout the project.

Lastly, and the most important, we extend our thanking to our parents and friends who have constantly guided and motivated us to accomplish the task successfully.

Patil Sarang
Maramwar Siddharaj
Gawande Mahesh
Damkondwar Shubham

Abstract

With the recent development of high resolution time-of-flight (TOF) positron emission tomography (PET) and resolution recovery incorporated in the reconstruction algorithm, accurate detection, and quantification of sub-centimeter nodules might become feasible. In this paper, we performed a comprehensive simulation to explore the quantitative accuracy of sub-centimeter nodules using the Otsu, Watershed and GLCM technique. We simulated nodules ranging from 4 to 10 mm in diameter, with 2:1 to 8:1 contrast level, at 1% to 100% (70 million) count-level, and with realistic respiratory motion amplitudes of 5/3, 10/6, and 20/12 mm. Images were reconstructed using motion-compensation ordered subset expectation Otsu thresholding algorithm. The results from this studies were consistent. Segmentation using the watershed transform works better to identify foreground objects and background locations. GLCM feature will be computed from the detected lung nodule in CT image, finally by using machine learning algorithm we detect actual lung cancer nodule.

List of Figures

List of Tables

Abbreviations

UML	Unified Modeling Language
py	Python
SDLC	Systems development life cycle

Contents

Chapter 1

INTRODUCTION

1.1 Background and Basics

Nowadays the cancer disease patients are increasing rapidly. The main reason behind it that the disease is not detect at early stage. The doctors or any medical consultant can manually analyze the nodule for their CT images. It is not gives the correct diagnosis in the early stage of disease. Sometime it detects in the critical stage. There are many existing system which can work for detecting the lung nodule. So we proposed a system which can predict the lung nodule from the lung CT images by using SVM machine learning algorithm.

1.2 Literature Survey

1. A morphological operation-based approach for Subpleural lung nodule detection from CT images, Rekka Mastouri, Henda Neji, Saoussen Hantous-Zannad, Nawres Khlifa, This paper is focus on an automatic segmentation approach of sub-pleural lung nodules from Computed Tomography (CT) scans based on morphological operations. Because the extraction of sub-pleural nodules is challenging and a computer-aided diagnosis system is, hence, indispensable. The proposed system is divided into three steps: pre-processing, initial detection of sub-pleural lung nodule and post-processing.

2. A Computer Aided Diagnosis for detection and classification of lung nodules, Lakshmi N arayanan A , Prof. Jeeva J.B, The aim of proposed system is to develop an efficient Computer Aided Diagnosis (CAD) for detection of lung nodules from parenchyma region of lung and classify the nodule into either cancerous (Malignant) or non-cancerous (Benign). The proposed system consists of following steps: i) the image taken is enhanced initially and then the region of interest is cropped, where the user can select the area to be cropped. ii) Morphological operation is performed to suppress the blood vessels and enhance the nodules. iii) Nodules are identified by labeling. iv) Features extraction. v) Neural networks are implemented for features classification. The proposed work was able to detect the lung nodule that falls in close proximity to the lung wall. The system is able to achieve an overall accuracy of 92.2%.

3.Automatic lung nodules detection in computed tomography images using nodule filtering and neural networks, A.R. Talebpour, H.R. Hemmati, M. Zarif Hosseinian, The proposed system implements a computer-aided detection (CAD) system that detects small size nodules (larger 3 mm) in High Resolution CT (HRCT) images. It used a cylindrical filter for filtering nodule cases from other objects in images. They use a lung LIDC image database.
4. Lung nodule detection based on 3D convolutional neural networks, Lei Fan, Zhaoqiang Xia, Xiaobiao Zhang, Xiaoyi Feng, This paper proposes a method to detect lung nodule of lung CT images using 3D convolutional neural networks. It combines the traditional morphological preprocessing methods and 3D convolutional neural networks are applied to lung CT images.

1.3 Project Undertaken

1.3.1 Problem Definition

Develop a system that is capable of analyzing CT images of lung nodules using convolutional neural network and providing nodule details for advanced stages of cancer.

1.3.2 Scope Statement

This project applicable to government or private hospitals.

Chapter 2

PROJECT PLANNING AND MANAGEMENT

2.1 Detail System Requirement Specification (SRS)

2.1.1 System Overview

Recently, image processing techniques are widely used in several medical areas for image improvement in earlier detection and treatment stages, where the time factor is very important to discover the abnormality issues in target images, especially in various cancer tumors such as lung cancer, breast cancer. Computer aided segmentation for computed tomography (CT) and magnetic resonance imaging (MRI) are finding the application in computer aided diagnosis, clinical studies, and treatment planning. A new automatic method has been proposed based upon black circular neighborhood algorithm and image processing techniques to extract the nodules. Feature extraction is an important step in algorithms. These separate the area which is then analyzed for detection of nodules to diagnose the disease. Computed tomography (CT), allows effective mapping .CT Image decreases the time complexity. We have used the GLCM Features Which Helps in detection of the nodule .The use of Otsu's Algorithm helps in detection of size and stage of the tumor.

2.1.2 Functional Requirements

- The application is user friendly.
- It provides an easy interface to user.
- The accessibility or response time of the application should be fast.

2.1.3 Non-Functional Requirements

- Interface Requirements: User interface should be simple and easy to understand.
- Performance Requirements: Our system should work properly on large no of users access.

- Software quality attributes such as availability
  - **Reliability :** The system should be reliable i.e there should be machine failure though load comes
  - **Modifiabilty :** The system should be scalable, rewritable,reusable, and can be updated
  - **Performance :**The performance should be real time and consistent
  - **Security :**The system should be secure i.e one should not be able to hack or change the system until there is no identity
  - **Reusablity :**The system should be reusable i.e it can be used by multiple users
  - **Portablility :**The system should be portable i.e enviornment should not matter to it in any situation.
  - **Adaptability :**It should be self adabtable as well as user adaptable

2.2 Project Process Modelling

2.2.1 Waterfall Model

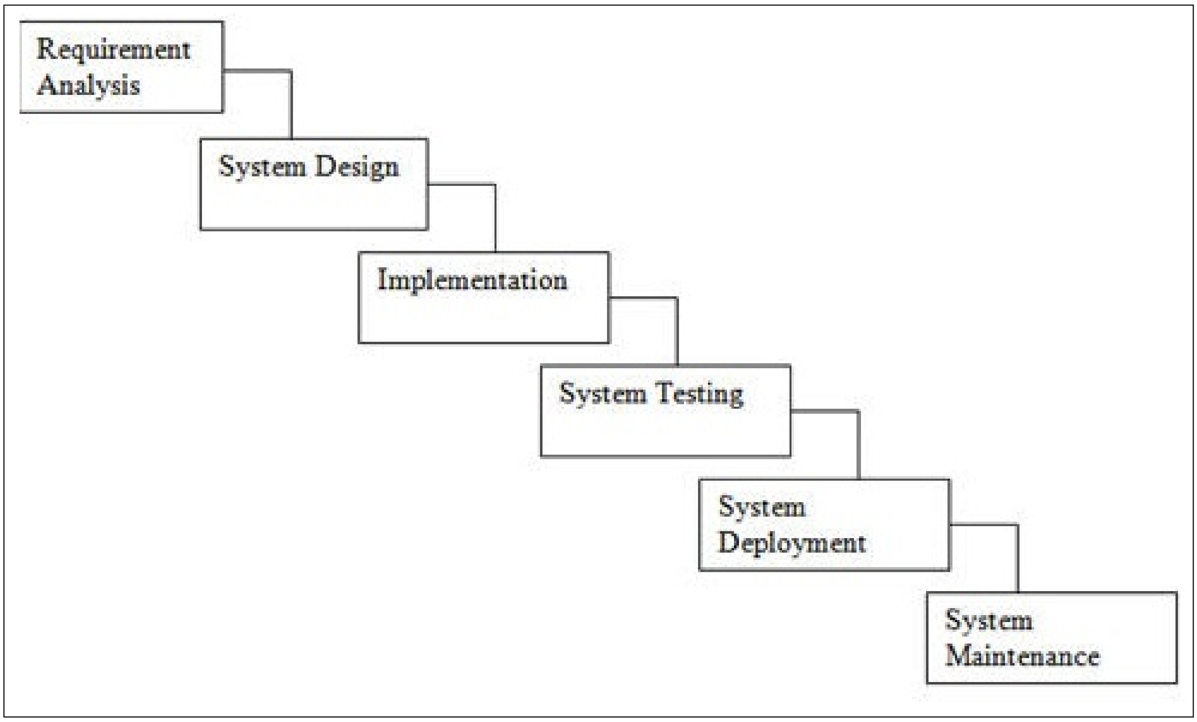

Figure 2.1: Waterfall Model

1. **Requirement gathering and analysis:**
   In this step of waterfall we identify what are various requirements are need for our project such are software and hardware required, database, and interfaces.

2. **System Design:**
   In this system design phase we design the system which is easily understood for end user i.e. user friendly. We design some UML diagrams and data flow diagram to understand the system flow and system module and sequence of execution.

3. **Implementation:**
   In implementation phase of our project we have implemented various module required of successfully getting expected outcome at the different module levels.With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

4. **Testing:**
   The different test cases are performed to test whether the project module are giving expected outcome in assumed time.All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

5. **Deployments of System:**
   Once the functional and non functional testing is done, the product is deployed in the customer environment or released into the market.

6. **Maintenance:**
   There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.All these phases are cascaded to each other in which progress is seen as flowing steadily downwards like a waterfall through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name ”Waterfall Model”. In this model phases do not overlap.

2.3 Cost & Efforts Estimates

COCOMO Model:

COCOMO (Constructive Cost Estimation Model) was proposed by Boehm [1981]. According to Boehm, software cost estimation should be done through three stages: Basic COCOMO, Intermediate COCOMO, and Complete COCOMO.

Basic COCOMO Model :

The basic COCOMO model gives an approximate estimate of the project parameters. The basic COCOMO estimation model is given by the following expressions:

Effort = a1 * (KLOC)a2 PM

Tdev = b1 x (Effort)b2 Months

Where,

1. KLOC is the estimated size of the software product expressed in Kilo Lines of Code,
2. a1, a2, b1, b2 are constants for each category of software products,
3. Tdev is the estimated time to develop the software, expressed in months,
4. Effort is the total effort required to develop the software product, expressed in person months (PMs).

The effort estimation is expressed in units of person-months (PM). It is the area under the person-month plot. It should be carefully noted that an effort of 100 PM does not imply that 100 persons should work for 1 month nor does it imply that 1 person should be employed for 100 months, but it denotes the area under the person-month curve.

Estimation of development effort:

For the three classes of software products, the formulas for estimating the effort based on the code size are shown below:

Organic : Effort = 2.4(KLOC)1.05 PM

Semi-detached : Effort = 3.0(KLOC)1.12 PM

Embedded : Effort = 3.6(KLOC)1.20 PM

Estimation of development time:

For the three classes of software products, the formulas for estimating the development time based on the effort are given below:

Organic : Tdev = 2.5(Effort)0.38 Months

Semi-detached : Tdev = 2.5(Effort)0.35 Months

Embedded : Tdev = 2.5(Effort)0.32 Months

Sr. No.	Month schedule	Phase	Work Status
1.	June-August	Topic searching	Topic Searched
2.	August-September	Topic selection	Topic Selected
3.	August-September	Project confirmation	Project Confirmed
4.	August-September	Literature Survey	Literature Survey Done
5.	September-October	Requirement Analysis	Requirement Analysis Done
6.	September-October	Requirement Gathering	Requirements Gathered
7.	November-December	Designing	Architecture Design
8.	November-December	Designing Test	GUI Tested
9.	November-December	Database Creation	Database Tested
10.	January-February	Coding	Coded Different modules
11.	January-February	Database And Modules Connectivity	Connectivity Done
12.	March	Testing of project	Project Tested
13.	April	Result Analysis	Result Analysis

Table 2.1: Project Scheduling

2.4 Project Scheduling

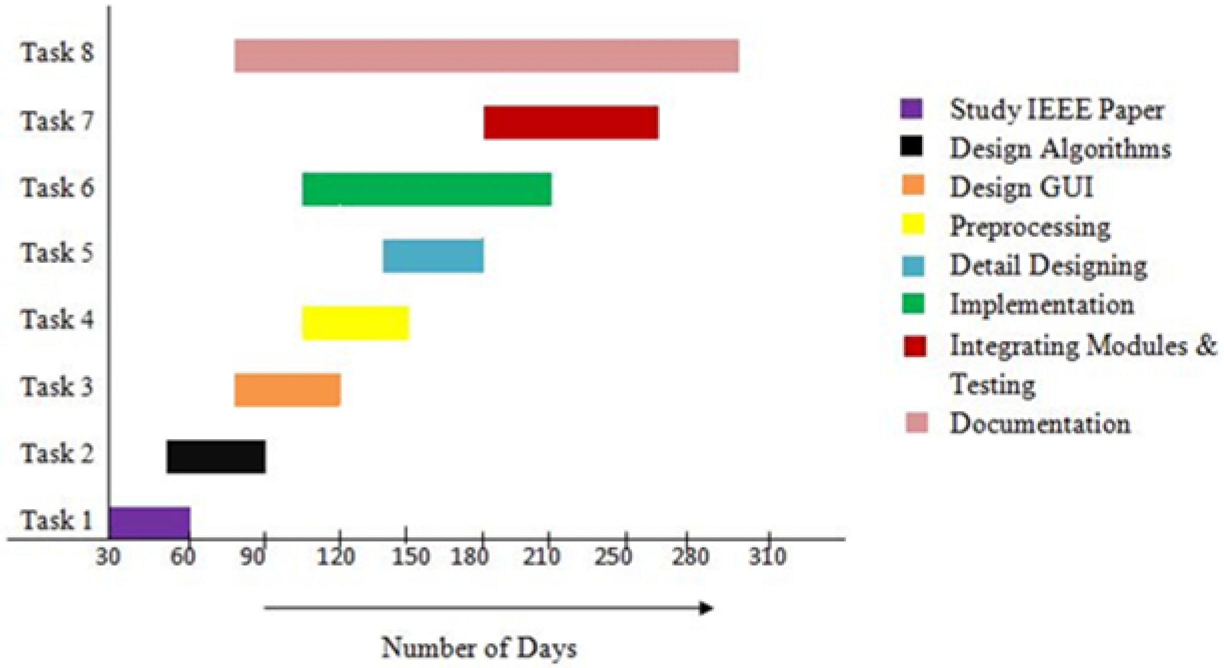

Figure 2.2: Time Line Chart

2.4.1 Time Line Chart

Chapter 3

ANALYSIS & DESIGN

3.1 IDEA Matrix

I	D	E	A
Increase The communication overhead will be increased to bits. We omit a detailed description of the variant scheme due to space limitations	**Drive** The necessity of Improving system by supporting noisy images	**Educate** we first preprocess, segmentation, feature extraction on image and then apply classification.	**Accelerate** We proposed an application to detect lung nodules
Improve The results are improved	**Deliver** High Performance in terms of machine learning process.	**Evaluate** Sensitive data being exposed	**Associate** System is associate with doctors to resolve issue in lungs.
Ignore Giving access to invalid User.	**Decrease** Chances of losing sensitive data.	**Eliminate** Noisiness in image.	**Avoid** Unwanted part of image

Table 3.1: IDEA Matrix

3.2 Mathematical Model

System Description:

S={I,O,F,success,failure} where,

- I=Input

- O=Output

- F=Functions

- I={U, Li }
  U:set of attributes
  Li: Lung Images

- Output : Detecting Lung Nodule.

- Functions : {UpImg, Pre-pross, Seg, FE, classify}

  1. UpImg: User can upload Lung Image as a input.
  2. Pre-pross: apply Pre-possessing on image using otsu threshoulding.
  3. Seg: Apply watershed for nodule segmentation.
  4. FE: Feature Extract done using GLCM
  5. Classify: finally using SVM we classify the given image.

- success: Our system predict nodule correctly in very short time.

- failure: If system crashed then can not get the answer.

3.3 Feasibility Analysis

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.
Three key considerations involved in the feasibility analysis are
• ECONOMICAL FEASIBILITY
• TECHNICAL FEASIBILITY
• SOCIAL FEASIBILITY

1. ECONOMICAL FEASIBILITY
   This study is carried out to check the economic impact that the system will have on

the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.
Our main goal is to develop our application within 10 months. here we estimate the project cost near to 14,000. So that amount is not grater for any organization to spend on the project.

2. TECHNICAL FEASIBILITY
This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.
In our project we use Lung image dataset we are using for detecting Lung nodule. We use python for developing our project. Here we develop our application on Anaconda IDE.

3. SOCIAL FEASIBILITY
The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.
We develop a application that will easy to understand, basically our app use by doctors that identity the Lung nodule by uploading lung image and get result from our application.

3.4 Architecture Diagram

In this system we use Lung image as a input and apply some techniques to identify the nodule of lung. Here first we use Otsu's thresholding method involves iterating through all the possible threshold values and calculating a measure of spread for the pixel levels each side of the threshold, i.e. the pixels that either fall in foreground or background. and then that images are filtered for image segmentation by using watershed algorithm. Segmentation using the watershed transform works better to identify, foreground objects and background locations. By applying GLCM feature that compute from the detected lung nodule in CT image. and finally we apply the SVM machine learning algorithm for detecting nodule of lung.

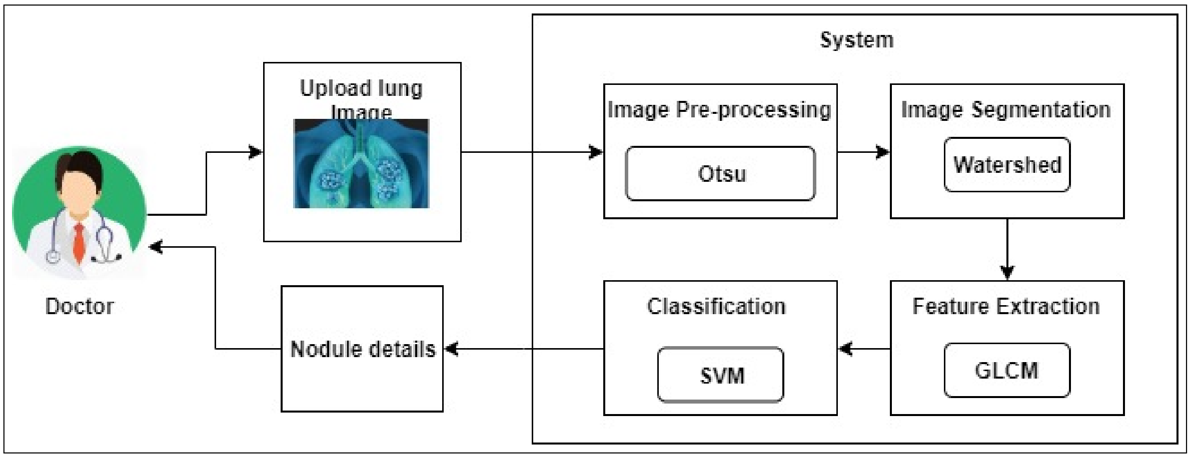

Figure 3.1: System Architecture

3.5 Data Flow Diagrams

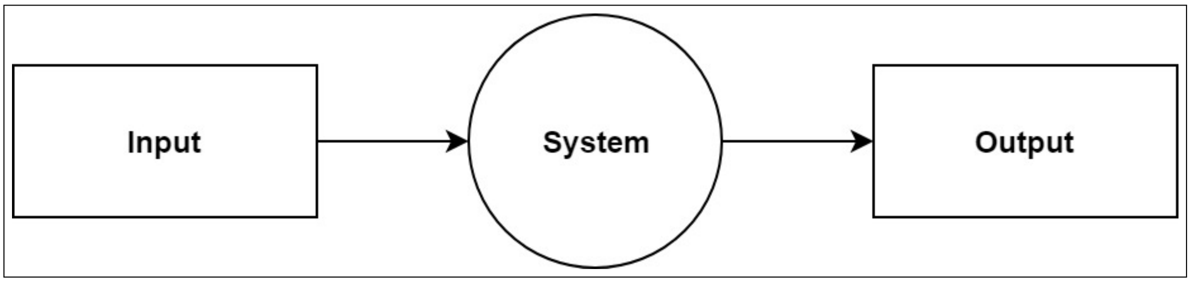

Figure 3.2: Data Flow diagram 0

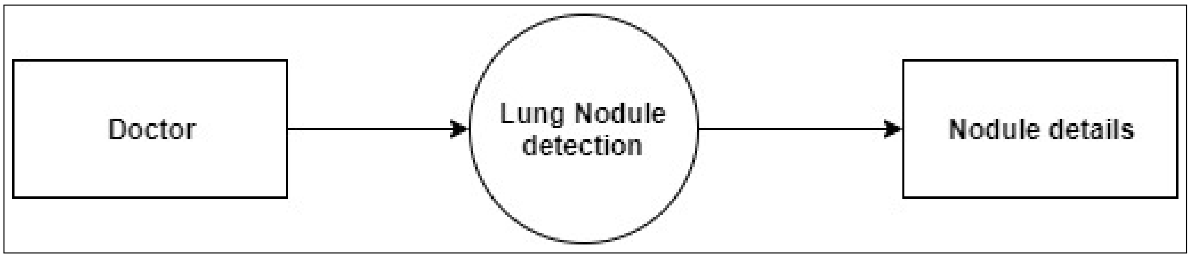

Figure 3.3: Data Flow diagram 1

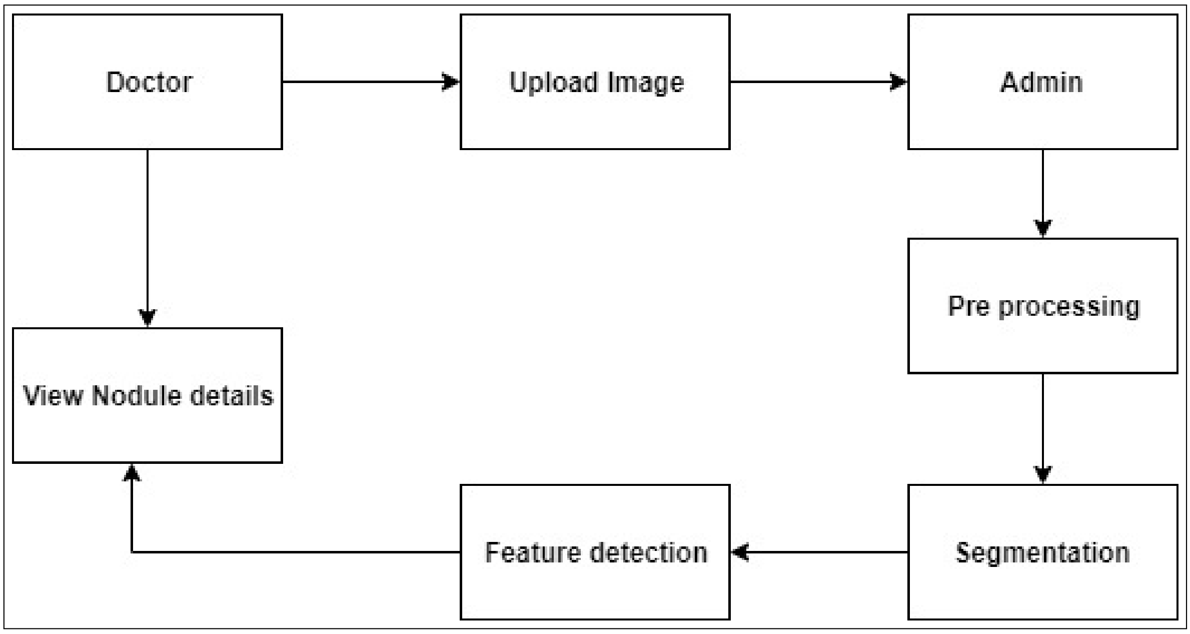

Figure 3.4: Data Flow diagram 2

3.6 ER Diagram

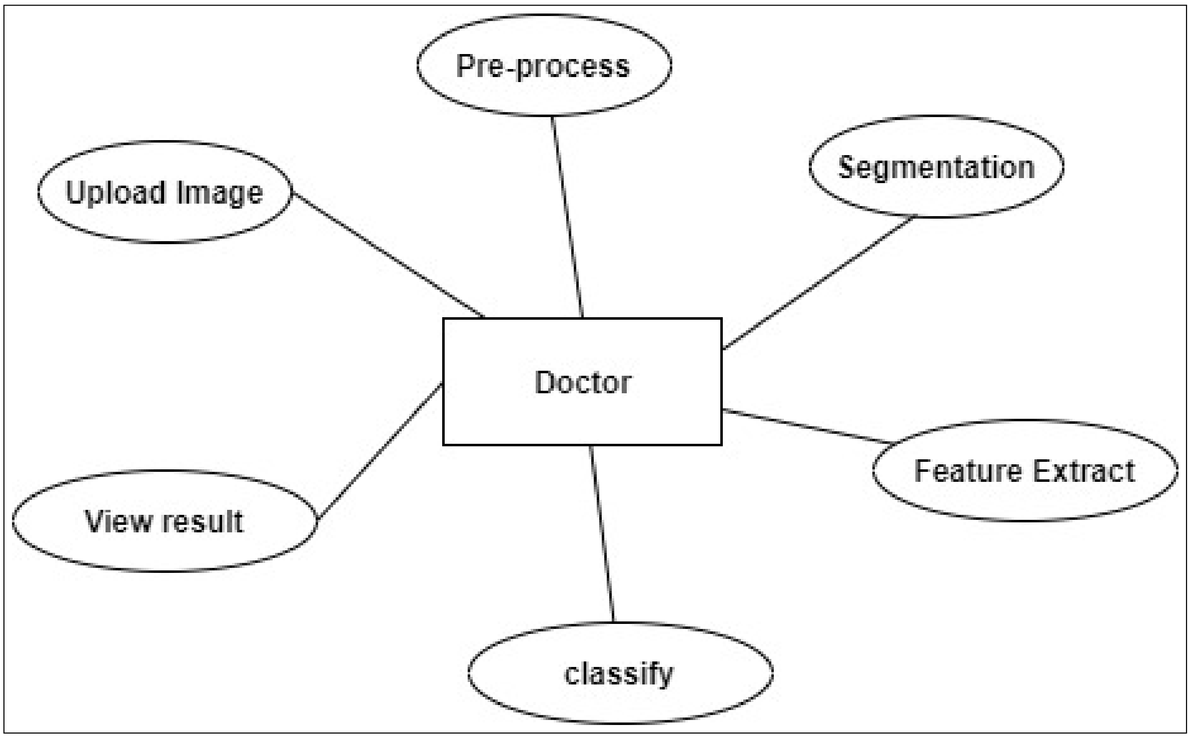

Figure 3.5: ER diagram

3.7 UML Diagrams

3.7.1 Class Diagram

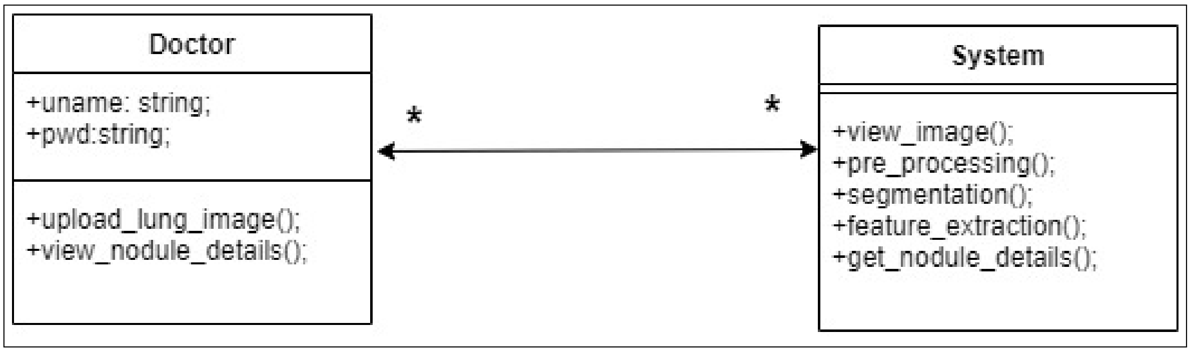

Figure 3.6: Class diagram

3.7.2 Use Case Diagram

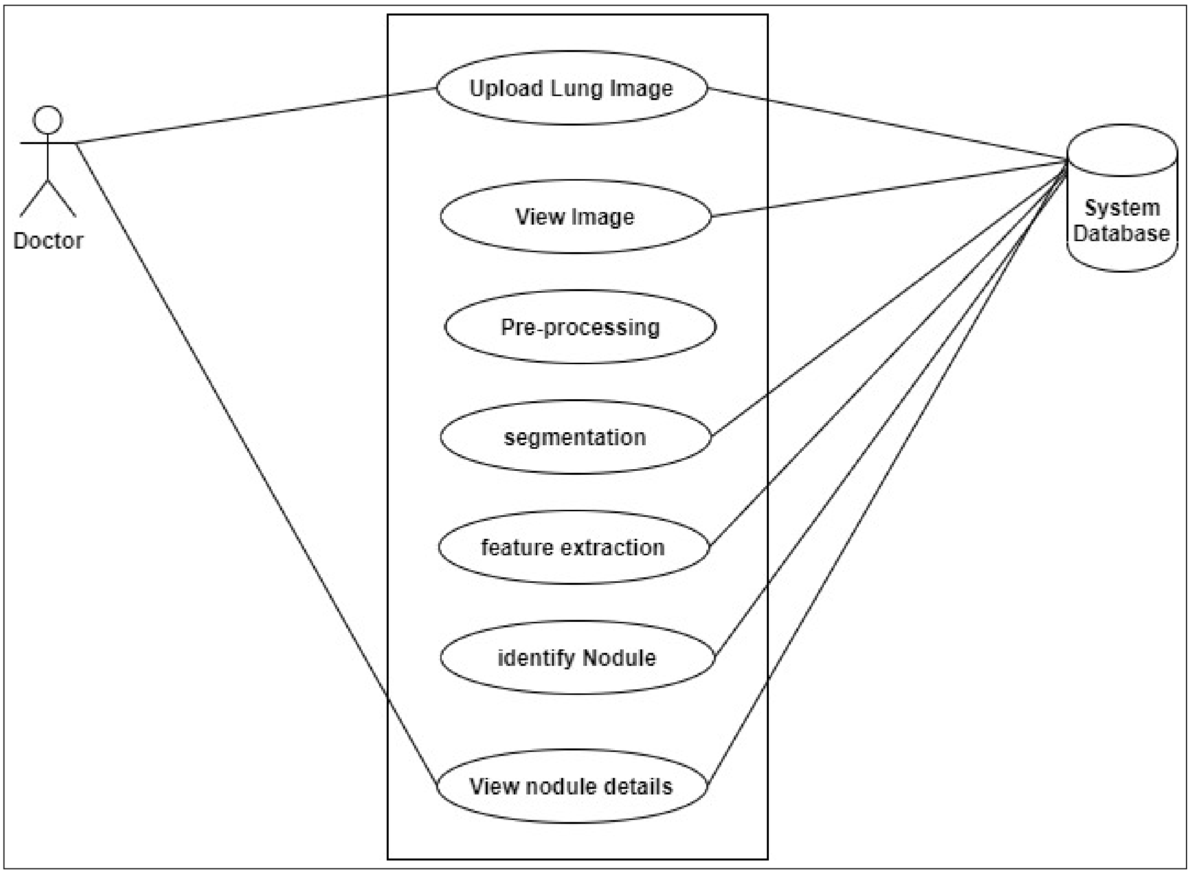

Figure 3.7: Use case diagram

3.7.3 Activity Diagram

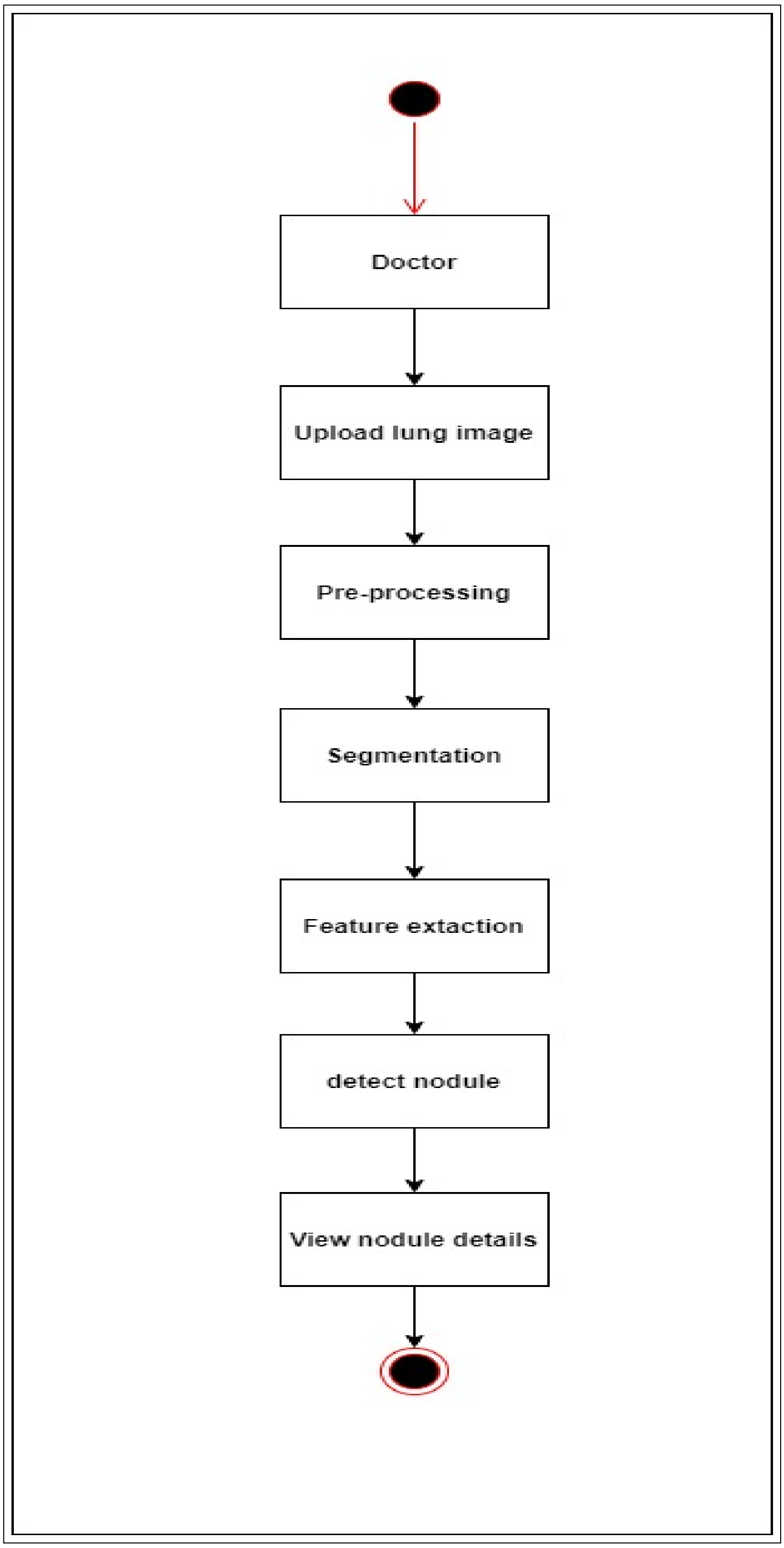

Figure 3.8: Activity diagram

3.7.4 Sequence Diagram

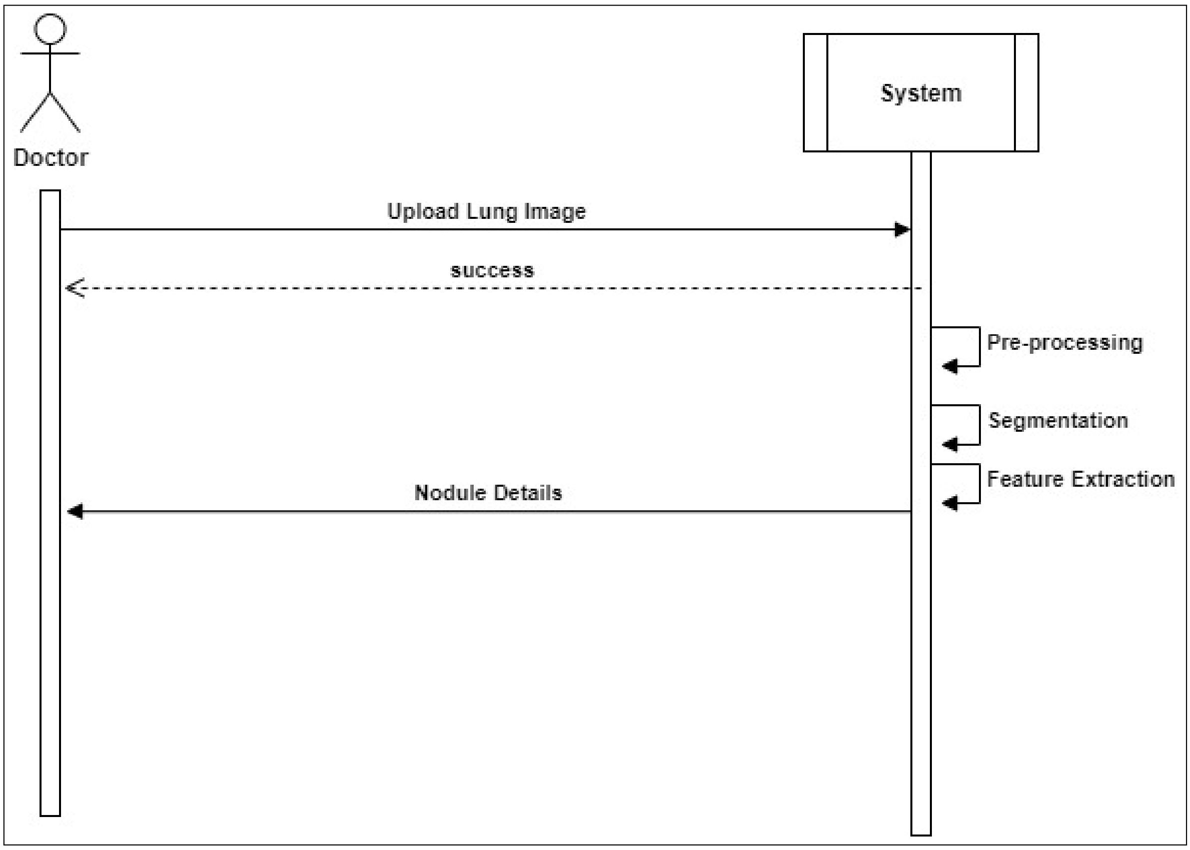

Figure 3.9: Sequence Diagram

3.7.5 State Transition Diagram

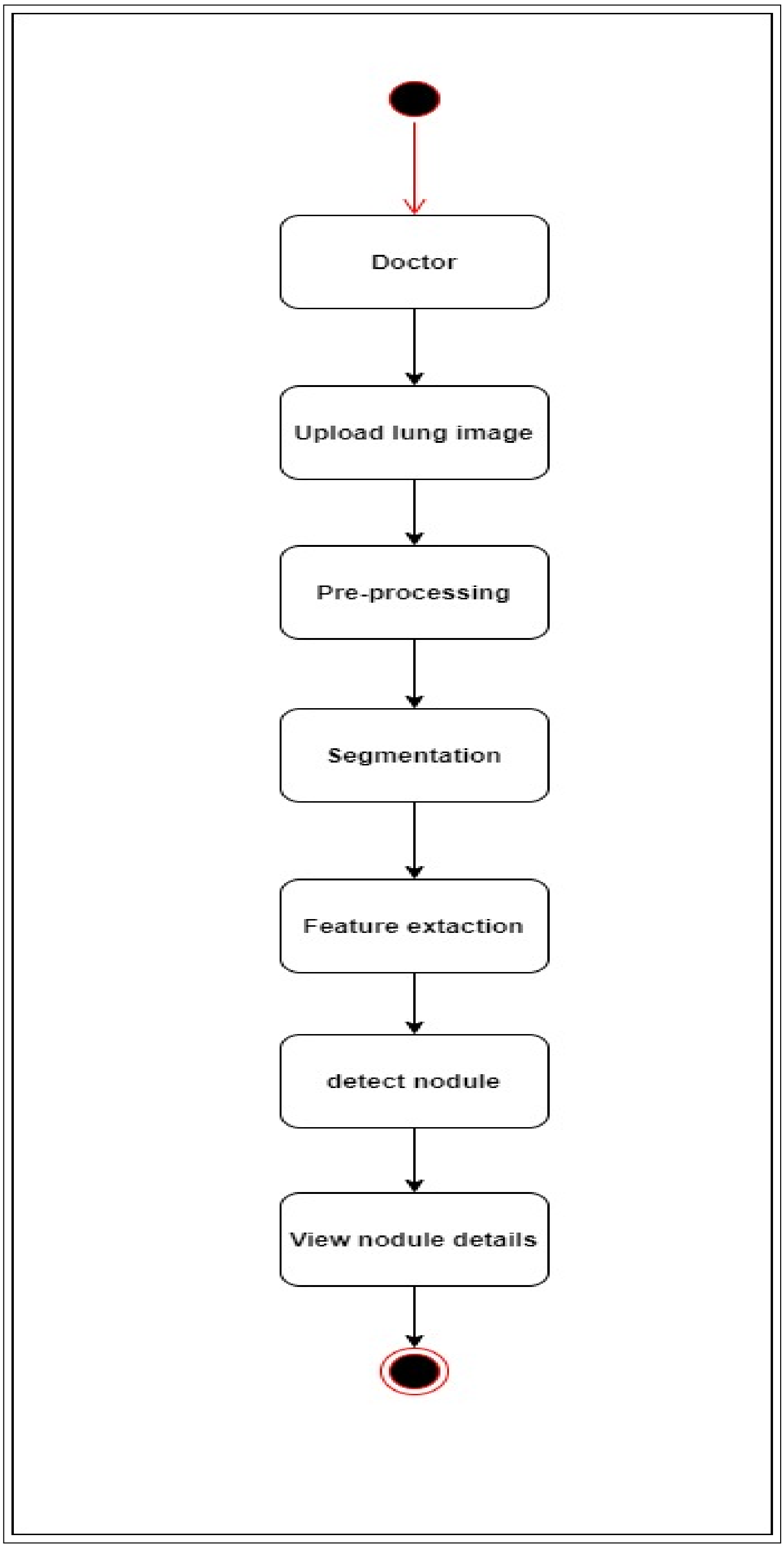

Figure 3.10: State Transition diagram

3.7.6 Component Diagram

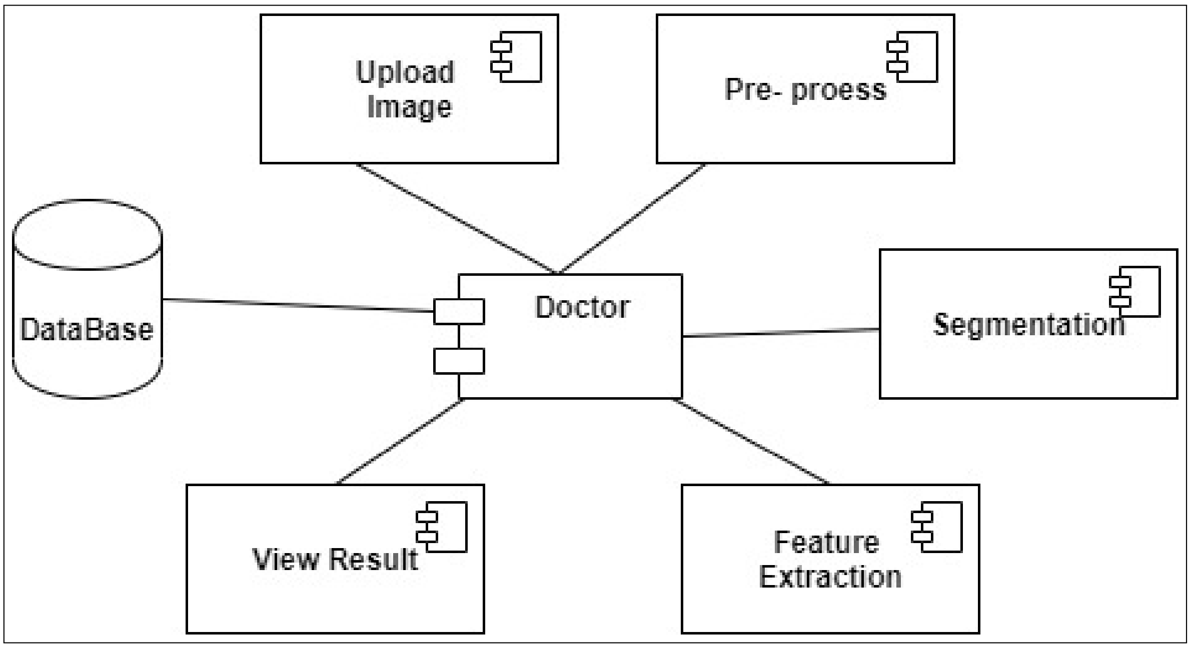

Figure 3.11: Component Diagram

3.7.7 Deployment Diagram

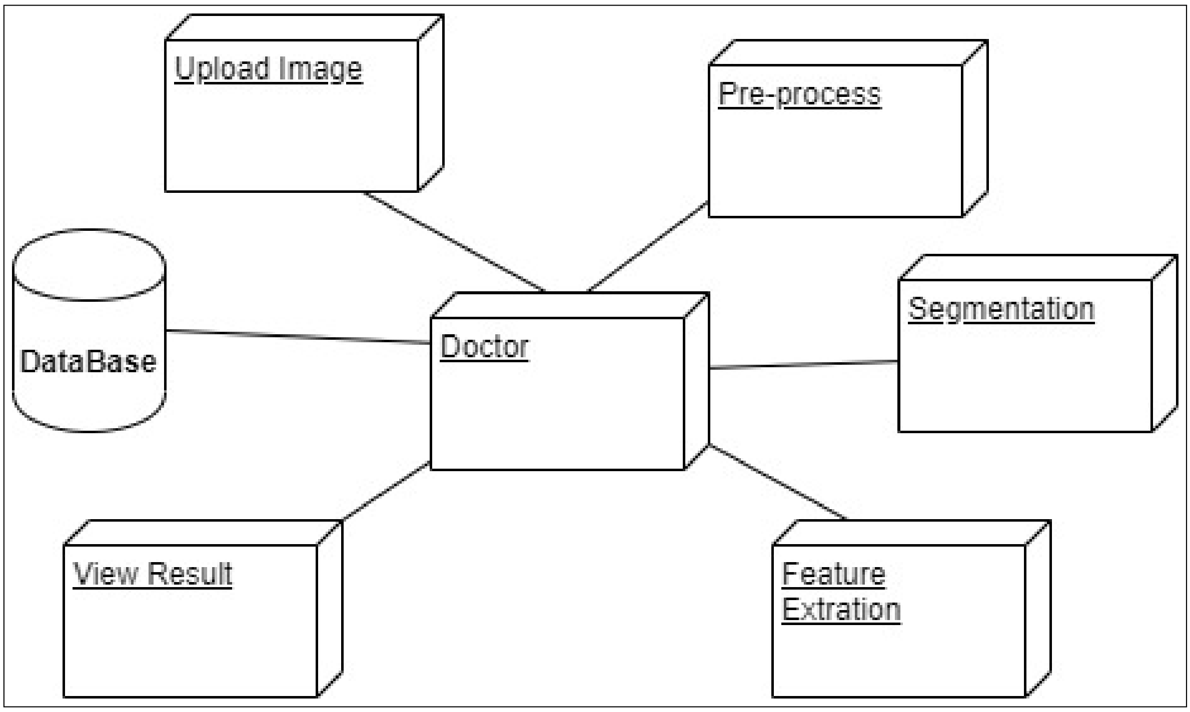

Figure 3.12: Deployment Diagram

Chapter 4

IMPLEMENTATION AND CODING

4.1 Introduction

- Upload lung image
- Preprocessing using Otsu Thresholding
- Segmentation
- Feature Extraction
- Get nodule details.
- View Result Nodule details

4.2 Operational Details

4.2.1 Hardware Resource Requirement

- Hardware : Pentium Dual Core
- Speed : 2.80 GHz
- RAM : 1GB
- Hard Disk : 20 GB

4.2.2 Software Resource Requirement

- Operating System : Windows 7 and above
- Technology : Python
- IDE : Anaconda or Spider

4.3 Code Listing

4.4 Screen Shots

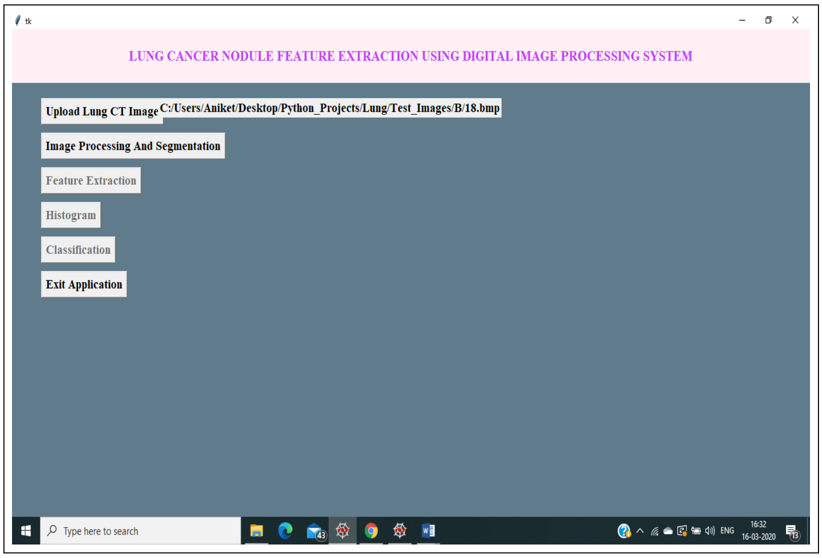

Figure 4.1: Screen Shot 1

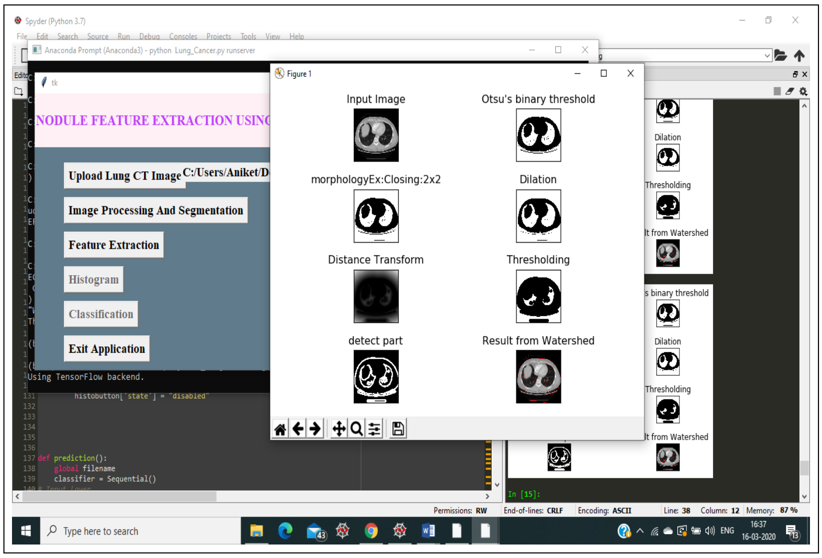

Figure 4.2: Screen Shot 2

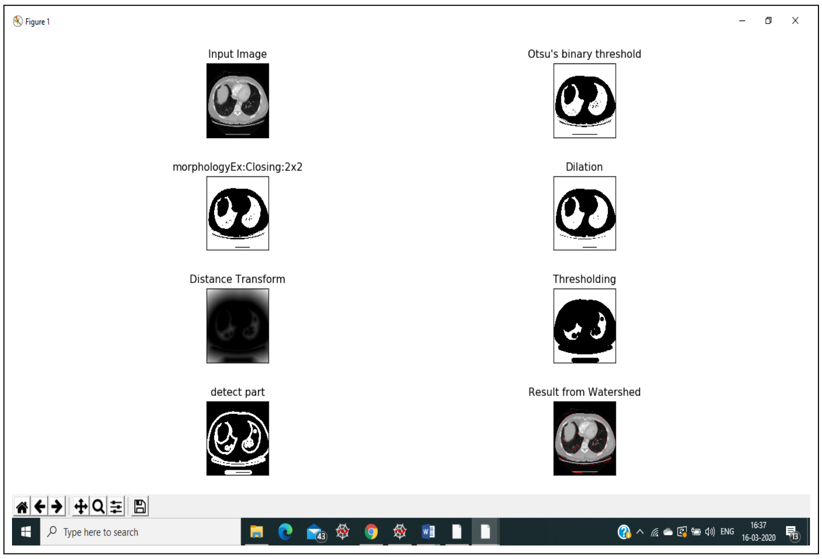

Figure 4.3: Screen Shot 3

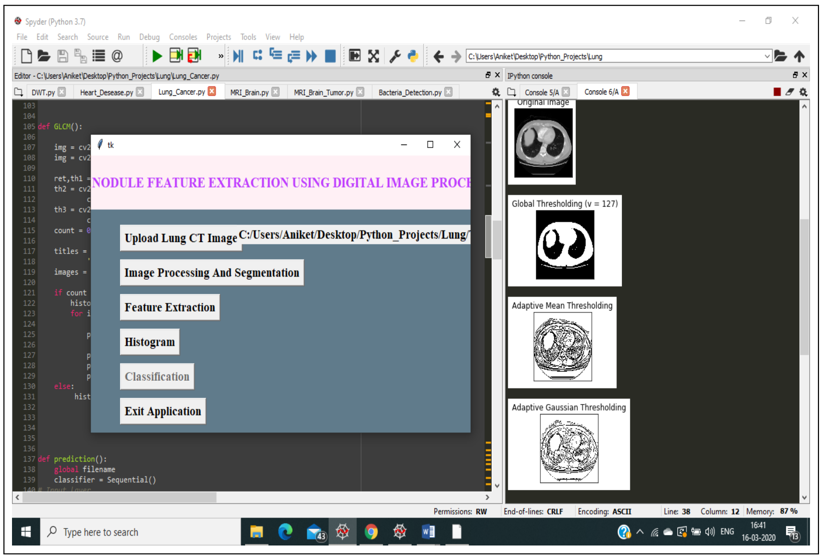

Figure 4.4: Screen Shot 4

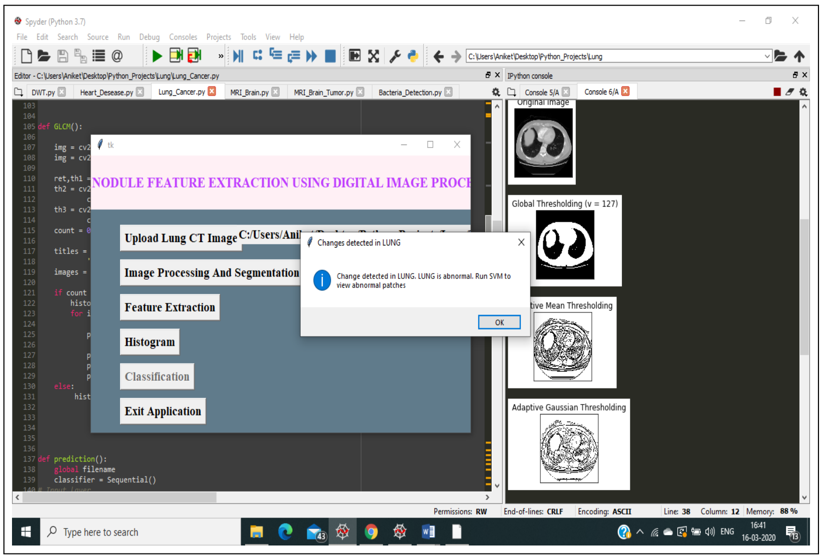

Figure 4.5: Screen Shot 5

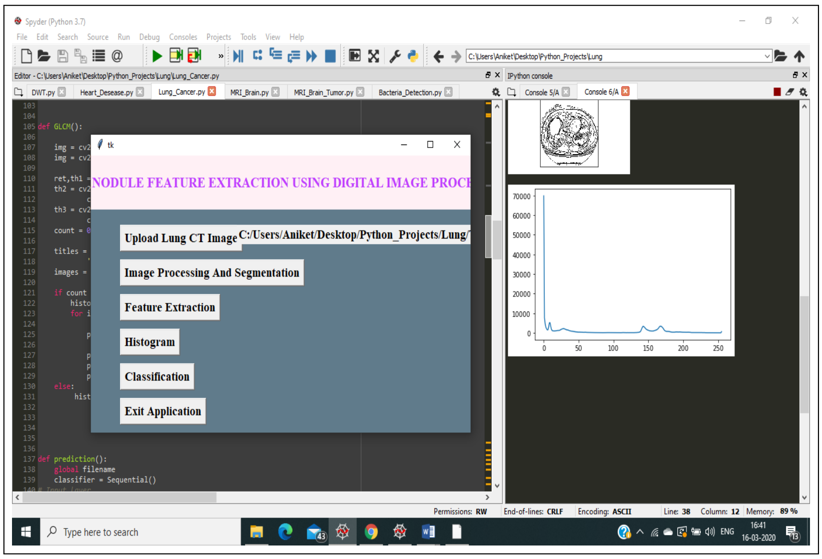

Figure 4.6: Screen Shot 6

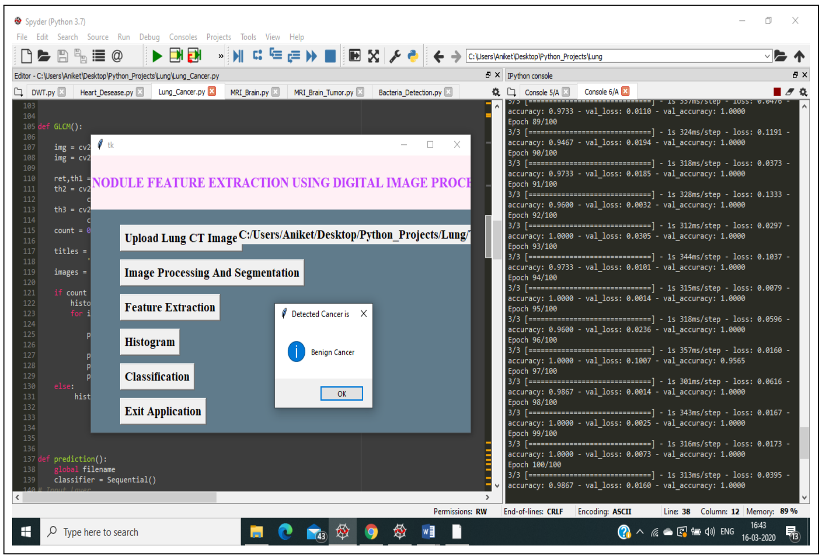

Figure 4.7: Screen Shot 7

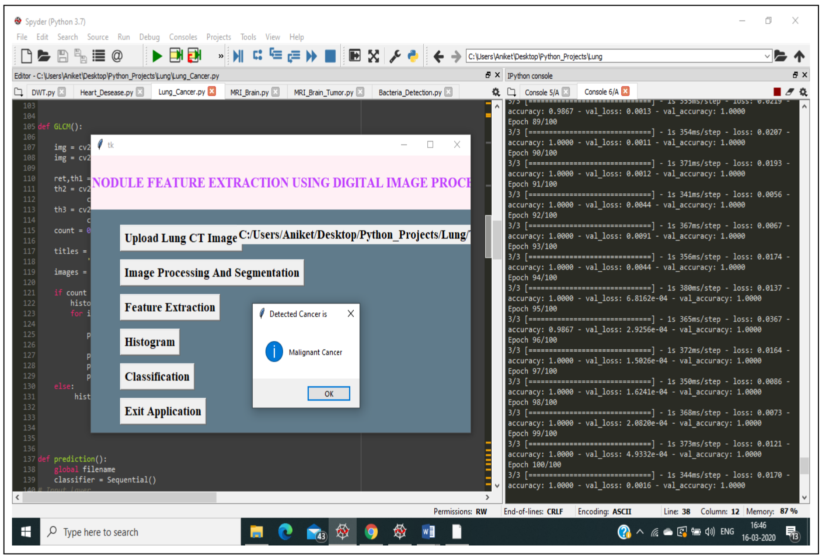

Figure 4.8: Screen Shot 8

Chapter 5

TESTING

5.1 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

5.2 Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

5.3 Acceptance Testing

The acceptance test suite is run using predefined acceptance test procedures to direct the testers which data to use, the step-by-step processes to follow and the expected result following execution. The actual results are retained for comparison with the expected results.[7] If the actual results match the expected results for each test case, the test case is said to pass. If the quantity of non-passing test cases does not breach the project's predetermined threshold, the test suite is said to pass. If it does, the system may either be rejected or accepted on conditions previously agreed between the sponsor and the manufacturer.

5.4 Test Cases and Test Results

Test Case	Upload Lung Image
Objective	System user should be upload Lung Image
Result	Upload Lung Image successfully
Test Case	Pre-Process
Objective	Perform Pre-Processing using otsu
Result	Pre-Process perform successfully
Test Case	Segmentation
Objective	Perform segmentation using Watershed
Result	Segmentation perform successfully
Test Case	Extract Feature from Image
Objective	Extract feature using GLCM
Result	Extract Feature successfully
Test Case	Apply SVM
Objective	SVM Apply for classification
Result	Classification done successfully
Test Case	View Result
Objective	Dr should view result correctly
Result	Dr view correct result successfully

Chapter 6

RESULTS AND DISCUSSION

6.1 Main GUI Snapshots

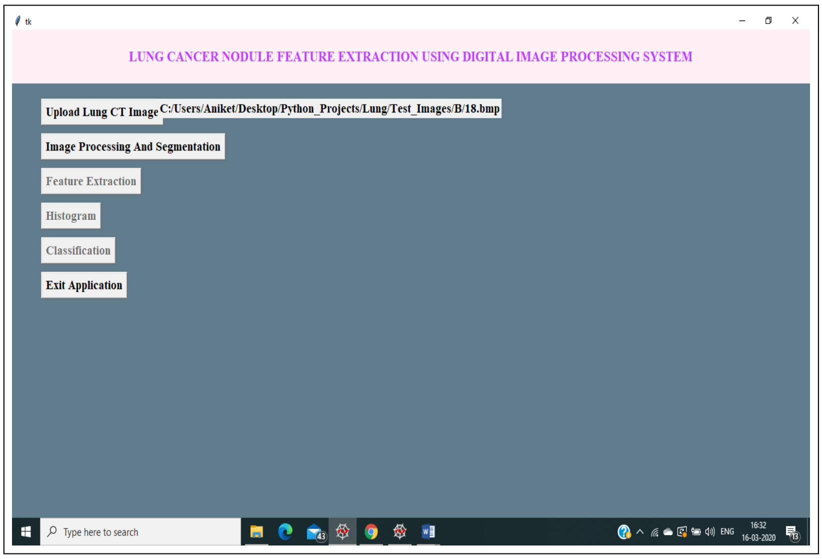

Figure 6.1: Screen Shot 1

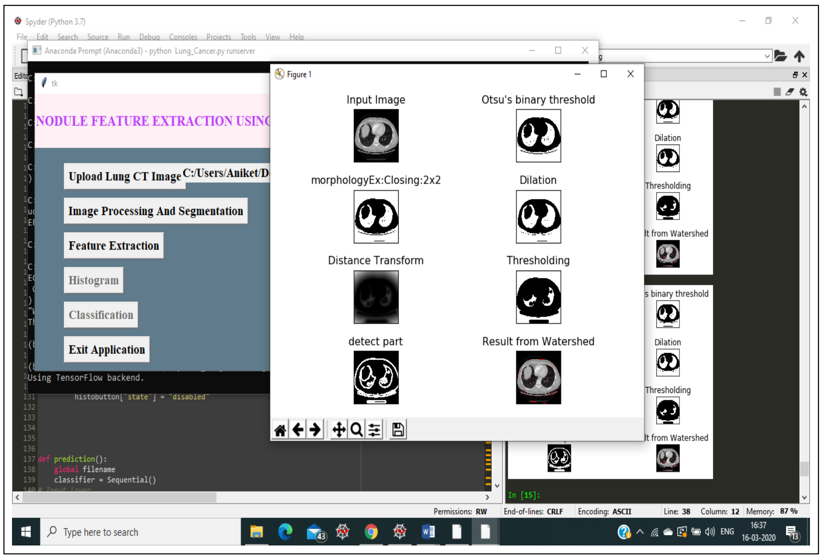

Figure 6.2: Screen Shot 2

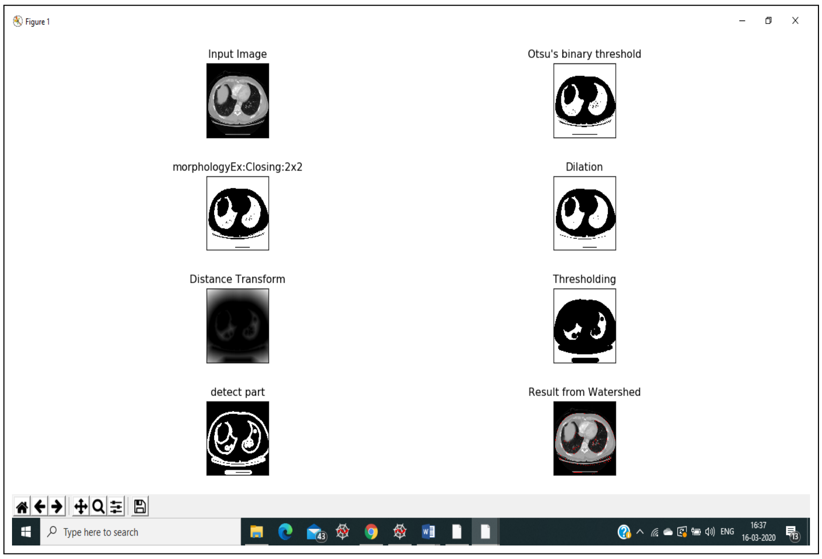

Figure 6.3: Screen Shot 3

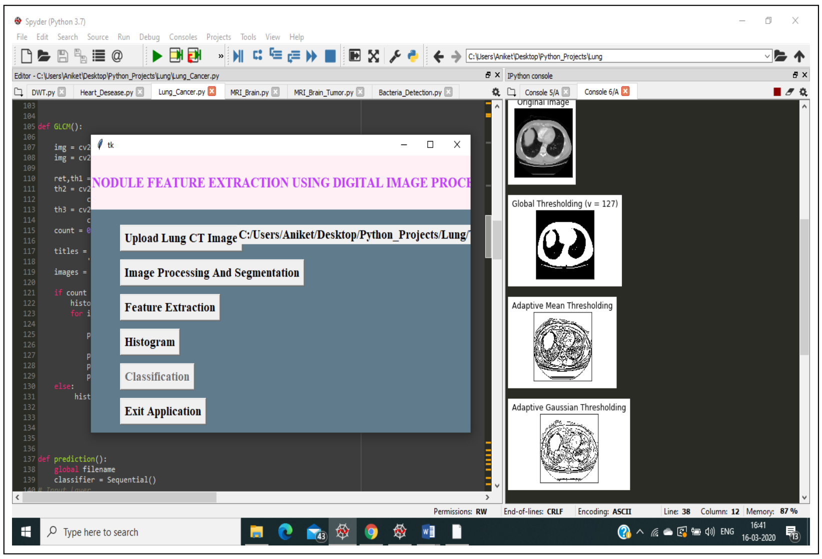

Figure 6.4: Screen Shot 4

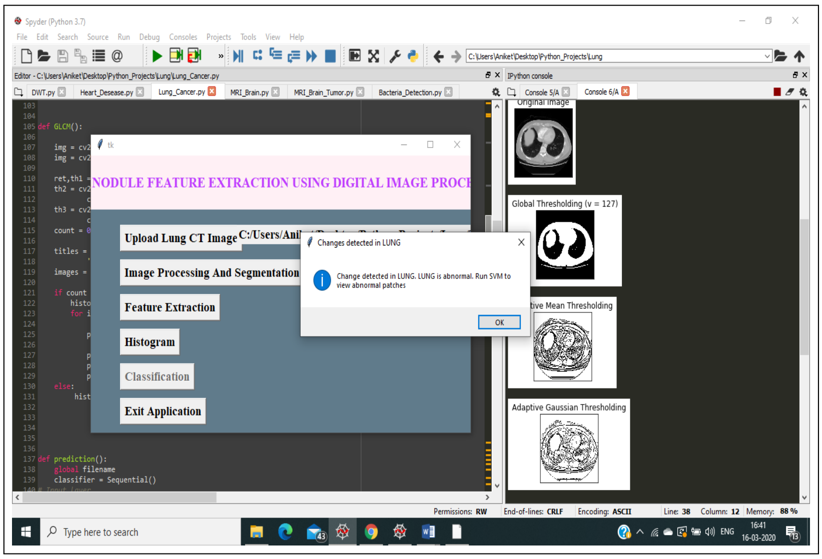

Figure 6.5: Screen Shot 5

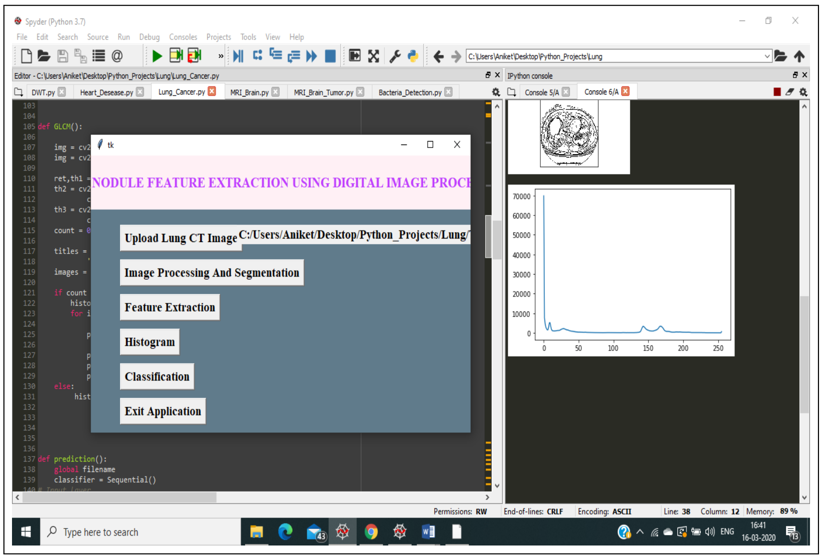

Figure 6.6: Screen Shot 6

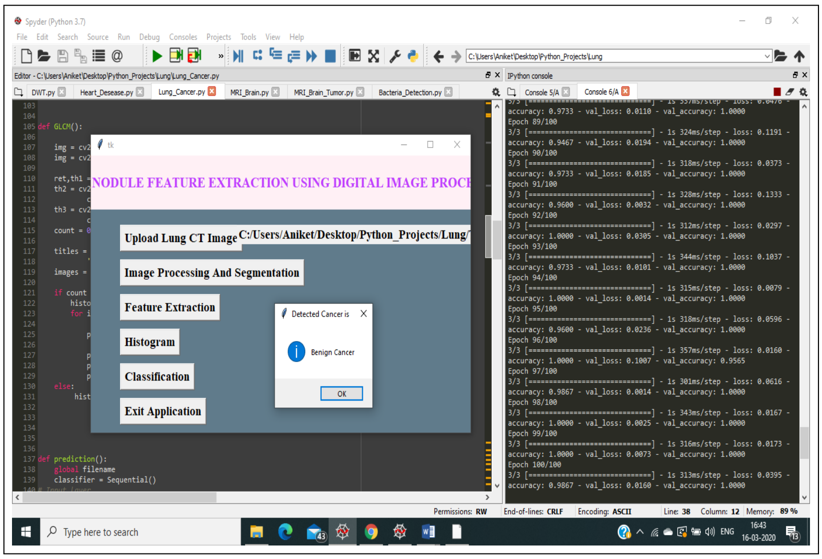

Figure 6.7: Screen Shot 7

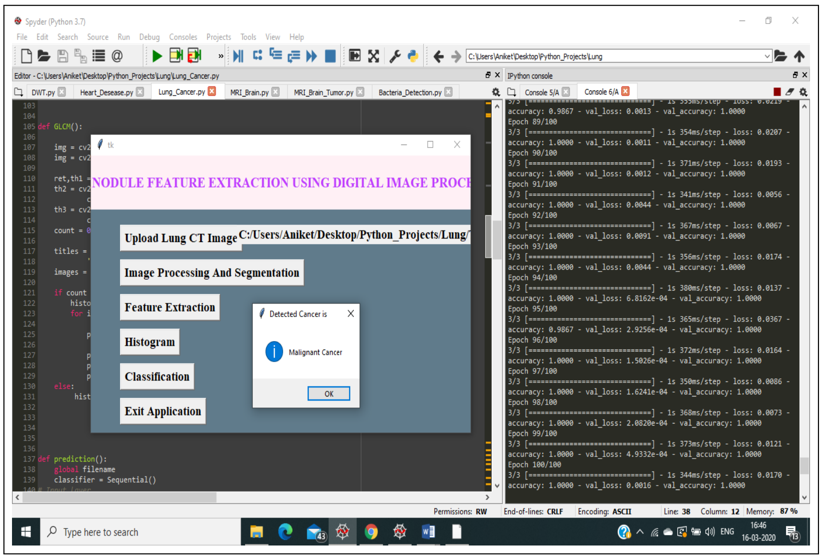

Figure 6.8: Screen Shot 8

6.2 Discussions

Here we detect lung nodule using machine learning techniques. Otsu threshouling used for the pre-proessing, watrshed use for segmentation an GLCM used for feature extration an finally get output using SVM Classification.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Our simulations showed that it is feasible to quantify sub-cm nodules within 20% bias using low-dose PET in the simulation. However, more thorough validations using clinically realistic phantoms should be performed. Reconstruction voxel size of 1 mm is recommended for small nodules. Respiratory motion correction is critical for nodules in the lung and abdomen. this project get accurate nodule size by using Otsu, Watershed, GLCM, and SVM.

7.2 Future Work

In future we would like to apply different techniques an compare them.

References

[1] Rekka Mastouri, Henda Neji, Saoussen Hantous-Zannad, Nawres Khlifa, " *A morphological operation-based approach for Subpleural lung nodule detection from CT images* ", 4th Middle East Conference on Biomedical Engineering (MECBME), IEEE, 2018.

[2] R. L. Siegel, K. D. Miller, and A. Jemal,"*Cancer statistics, 2016*" CA Cancer J. Clinicians, vol. 66, no. 1, pp. 7–30, 2016.

[3] F. C. Detterbeck, P. J. Mazzone, D. P. Naidich, and P. B. Bach, "*Screening for lung cancer: Diagnosis and management of lung cancer, 3rd ed: American College of chest physicians evidence-based clinical practice guidelines*", Chest, vol. 143, no. 5, pp. E78S–E92S, May 2013.

[4] P. B. Bach et al., "*Benefits and harms of CT screening for lung cancer: A systematic review*", J. Amer. Med. Assoc., vol. 307, no. 22, pp. 2418–2429, Jun. 2012.

[5] W. C. Hanna et al., "*Minimal-dose computed tomography is superior to chest X-ray for the follow-up and treatment of patients with resected lung cancer,*" J. Thoracic Cardiovasc. Surgery, vol. 147, no. 1, pp. 30–35, Jan. 2014.

[6] D. R. Aberle et al., "*Reduced lung-cancer mortality with low-dose computed tomographic screening,*" New England J. Med., vol. 365, no. 5, pp. 395–409, Aug. 2011.

[7]D. Shlomi, R. Ben-Avi, G. R. Balmor, A. Onn, and N. Peled, "Screening for lung cancer: Time for large-scale screening by chest computed tomography," Eur. Respiratory J., vol. 44, no. 1, pp. 217–238, Jul. 2014.

[8] M. Infante et al., "*Lung cancer screening with spiral CT: Baseline results of the randomized DANTE trial,*" Lung Cancer, vol. 59, no. 3, pp. 355–363, Mar. 2008.

[9] M. Infante et al., "*A randomized study of lung cancer screening with spiral computed tomography: Three-year results from the DANTE trial,*" Amer. J. Respiratory Critical Care Med., vol. 180, no. 5, pp. 445–453, Sep. 2009.

[10] D. S. Gierada et al., "*Projected outcomes using different nodule sizes to define a positive CT lung cancer screening examination,*" J. Nat. Cancer Inst., vol. 106, no. 11, 2014, Art. no. dju284, doi: 10.1093/jnci/dju284.