A

Major Project Report

on

Prediction of Lung Cancer Using VGG19 Transfer Learning

submitted in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology

By

B. Sai Shashank

(20EG105641)

B. Pranaya

(20EG105643)

R. Nikhitha Reddy

(20EG105656)

K. Anvitha Rao

(20EG105709)

Under The Guidance
Of

Mrs. K. Rashmi

Assistant Professor,

Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ANURAG UNIVERSITY

VENKATAPUR (V), GHATKESAR (M), MEDCHAL (D), T.S 500088

(2023-24)

I

DECLARATION

We hereby declare that the Report entitled **Prediction of Lung Cancer Using VGG19 Transfer Learning** submitted for the award of Bachelor of technology Degree is our original work and the Report has not formed the basis for the award of any degree, diploma, associateship or fellowship of similar other titles. It has not been submitted to any other University or Institution for the award of any degree or diploma.

Place: Anurag University, Hyderabad

Date: 12/04/2024

B. Sai Shashank

(20EG105641)

B. Pranaya

(20EG105643)

R. Nikhitha

(20EG105656)

K. Anvitha Rao

(20EG105709)

II

CERTIFICATE

This is to certify that the report entitled **Prediction of Lung Cancer Using VGG19 Transfer Learning** that is being submitted by **Mr. B. Sai Shashank** bearing the hall ticket number **20EG105641**, **Ms. B. Pranaya** bearing the hall ticket number **20EG105643**, **Ms. R. Nikhitha Reddy** bearing the hall ticket number **20EG105656, Ms. K. Anvitha Rao** bearing the hall ticket number **20EG105709** in partial fulfilment for the award of B. Tech degree in Computer Science and Engineering to Anurag University is a record of bonafide work carried out by them under my guidance and supervision.

      The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

Mrs. K. Rashmi   Dr. G. Vishnu Murthy

Assistant Professor   Dean, CSE

Department of CSE

External Examiner

ACKNOWLEDGEMENT

IV

ABSTRACT

Lung cancer is one of the deadly diseases whose prediction is required to reduce the death rate. So, Artificial intelligence is used on CT scan images are used for achieve better accuracy in an automated manner. Deep Learning is one of the emerging trends for predicting values. Convolution Neural Networks is one of the deep learning algorithms implemented to sample produces better outcomes than other machine learning algorithms. In this paper, the dataset has been taken with 1000 images of chest scans for different types of lung cancers such as Adenocarcinoma, Benign, and Squamous Cell Carcinoma. Multiple machine learning algorithms have been compared and then it has been confirmed that CNN is one of the best among all to check the accuracy of the prediction.

The existing system includes VGG-16 but the model achieves only 77.62% which is not very effective, so, the proposed system implements the VGG-19 transfer learning model on datasets with different types of lung cancer, thus helping to check the severity and precautions for the same in a distinct manner.

Keywords: VGG199, Transfer Learning, Deep Learning, Small Lung Cancer.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

Lung cancer poses a significant global health threat, ranking as a leading cause of mortality worldwide. Its asymptomatic nature in early stages complicates timely diagnosis. This project focuses on leveraging transfer learning and computer-aided diagnosis (CAD) techniques to detect lung cancer at its onset. By adapting pre-trained models and employing deep learning and image processing algorithms, the aim is to enhance the accuracy and efficiency of lung cancer detection. Through the integration of advanced technology, this project strives to revolutionize early diagnosis, providing clinicians with a powerful tool to improve patient outcomes.

1.2   PURPOSE OF THE PROJECT

The purpose of this project is to develop a cross-platform web application that is capable of generation and verification of educational documents. This application aims to make the experience of certificate generation and verification very easy and fool-proof. Whenever an employer or an institution wants to verify a candidate's credentials and scores in their degreesor certificates, they can simply use this application to verify the file given by the student, and find out whether it is legitimate page

1.3 MOTIVATION

The project aims to revolutionize the early detection of lung cancer through the integration of advanced technology and innovative methodologies. Recognizing the significant global health threat posed by lung cancer, the primary objective is to improve patient outcomes by addressing the challenges associated with its asymptomatic nature in early stages. Leveraging transfer learning techniques, the project seeks to analyze image datasets efficiently for early classification and detection of lung cancer. By exploring machine learning, deep learning, and image processing techniques, researchers aim to develop a robust computer-aided diagnosis (CAD) system capable of accurately detecting lung cancer. Through this endeavor, the project strives to provide clinicians with a powerful tool for timely intervention, ultimately saving lives of lung cancer worldwide.

CHAPTER 2
LITERATURE SURVEY

Multiple researchers surveyed and resulted to some outcomes such as C. Yao et al. [11] designed a CNN model taking self-made data set achieved 90% accuracy but the validity of data and accuracy on real time data sets remained doubtful. So, the method needs to be tested at several other data sets to check its reliability. In the same manner M. Norouzi [1] claimed that nanotechnology is one of the efficient methods to be applied for therapies for lung cancer rather than complicated chemotherapies andit can also reduce the amount of toxicity. The survey has proved nanotechnology a great method to be considered with conditions applied because patient selection and combinations of multiple treatment provides an advanced enhancement.

Multiple papers have been surveyed in J. Wang et al. [9]and verified that false positive rates of those are quite high which decrease the accuracy, so to overcome the problem, 3 dimensional - convolutional neural network, VGG 16, Alex Net and Multi Crop Net L.ye et al. [12] advanced from 2-dimensional architecture which extracted 8.28% which is relatively low when compared with other algorithms.

A data set of 1000+ images has been taken in J. Wang et al. [13] with stage T1a-3N0M0, NSLC where it has been claimed that non-small lung cancer is more dangerous than small lung cancer, the death rate compared, the former is considered more harmful. Survival rate has been checked by A. Agaimy et al. [6] after the first diagnosis of non-small lung cancer to check the criticality of it. The data contains both the genders which includes 8 males and 6 females under the age group of 52to 85 years (median 60). The maximum survival rate after the diagnosis was of one patient who was having cerebral metastasis, alive even after 45 months.

VGG 16 implementation is also done in paper S. T. M. Sheriff et al. [14] where the result only reveals the presence of lung cancer or not with not so good accuracy, so the inspiration has been taken from the paper to implement not just on cancerous images but also the types of lung cancer so as to check that how critical the consequences would be and what precautions need to take care.

Multiple CNN models have been checked in M. Phankokkruad et al. [15] such as VGG16, ResNet50V2, and DenseNet201 which are based on transfer learning. Every model predicted its accuracy provided as 62%, 90%, and 89%, respectively.

2.1 EXISTING SYSTEM

The VGG-16 model demonstrates the ability to discern between different types of Non-Small Cell lung cancer, offering valuable insights for diagnosis and threat assessment. Despite this capability, its effectiveness remains constrained, with an accuracy rate of only 77.62%. To enhance its performance and reliability, the integration of multiple other algorithms is imperative. By leveraging complementary approaches, a more efficient and robust diagnostic tool can be developed, ultimately improving the accuracy and utility of lung cancer diagnosis.

Researchers recognize the potential of the VGG-16 model in providing nuanced results for Non-Small Cell lung cancer subtypes, aiding clinicians in making informed decisions. However, its current accuracy rate of 77.62% falls short of optimal performance standards. Thus, there is a pressing need to explore and incorporate additional algorithms to augment the model's capabilities and enhance its efficacy in diagnosing lung cancer. Through collaborative efforts and innovative approaches, researchers aim to develop a more accurate and reliable diagnostic solution for improved patient care.

2.2 DISADVANTAGES OF EXISTING SYSTEM:

- Lack of specificity regarding alternative algorithms.
- Insufficient context for accuracy assessment.
- Limited discussion of implications.
- Ignoring potential challenges.

CHAPTER 3

3.1 PROPOSED SYSTEM

Lung Cancer has a severe effect on generation with the growing death rates. It has the second largest death among all the cancer types. As per, the American Cancer Society, it has been surveyed that Lung cancer needs to be diagnosed at an early stage so as to achieve recovery otherwise, recovery is suspected while prediction in late stages. There are multiple ways to detect Lung Cancer with medical Images such as CT scans, MRIs, and X-rays among which CT scan is considered one of most efficient one among these. But instead of manual checkups, automatic detection can help in providing better results. The VGG19 transfer learning model is considered one of the best methods to work on images for prediction. So, this system includes multiple CT scan images of multiple cancer types such as Lung adenocarcinoma, benign, and squamous cell carcinoma.

3.2 ADVANTAGES OF PROPOSED SYSTEM

The proposed system has the following advantages:

- Clarity on lung cancer severity: Highlights the significant impact of lung cancer on mortality rates, emphasizing the urgency for effective detection and treatment.

- Emphasis on early diagnosis: Stresses the importance of early detection in improving patient outcomes, thereby raising awareness about the critical need for timely screening.

- Recognition of advanced imaging techniques: Acknowledges the role of advanced medical imaging modalities such as CT scans in enhancing diagnostic accuracy, providing a foundation for effective lung cancer detection.

- Affirmation of automated detection systems: Recognizes the potential benefits of automated detection systems over manual methods, including increased efficiency and consistency in diagnosis.

- Validation of VGG19 transfer learning model: Affirms the effectiveness of the VGG19 transfer learning model in analyzing medical images and predicting lung cancer, highlighting its value as a diagnostic tool.

- Inclusion of diverse cancer types: Ensures comprehensive coverage of various lung cancer types, facilitating more accurate and personalized diagnosis and treatment planning.

3.3 SYSTEM REQUIREMENTS

The system requirements for the development and deployment of the project as an applicationare specified in this section. These requirements are not be confused with the end-user systemrequirements. There are no specific, end-user requirements as the intended application is cross-platform and is supposed to work on devices of all form-factors and configurations.

3.3.1 SOFTWARE REQUIREMENTS

Operating System   :   Windows family

Technology   :   Python 3.8

Front-end Technology   :   HTML, CSS, JS

Back-end Technology   :   MySQL

Code Development Tool   :   PyCharm IDE

Web framework   :   Flask

3.3.2   HARDWARE REQUIREMENTS

Processor   :   Any Update Processor

Ram   :   Min 4 GB

Hard Disk   :   Min 250 GB

3.4 IMPLEMENTATION TECHNOLOGIES

ALGORITHM USED: VGG-19:

VGG-19 is a 19-layer deep convolutional neural network (CNN). It is a popular method for image classification because it uses multiple 3×3 filters in each convolutional layer. VGG-19 is trained on the ImageNet database, which contains a million images of 1000 categories. A pre-trained version of the network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. Here VGG-19 can achieve an accuracy of 95% with a loss of 17%. Compared with existing methods, VGG-19 has a faster training speed, fewer training samples per time, and higher accuracy.

The VGG network is constructed with very small convolutional filters. The VGG-19 consists of 16 convolutional layers and three fully connected layers.

Let's take a brief look at the architecture of VGG:

Input: The VGG Net takes in an image input size of 224×224. For the ImageNet competition, the creators of the model cropped out the center 224×224 patch in each image to keep the input size of the image consistent.

Convolutional Layers: VGG's convolutional layers leverage a minimal receptive field, i.e., 3×3, the smallest possible size that still captures up/down and left/right. Moreover, there are also 1×1 convolution filters acting as a linear transformation of the input. This is followed by a Re LU unit, which is a huge innovation from Alex Net that reduces training time. ReLU stands for rectified linear unit activation function; it is a piecewise linear function that will output the input if positive; otherwise, the output is zero. The convolution stride is fixed at 1 pixel to keep the spatial resolution preserved after convolution (stride is the number of pixels shifts over the input matrix).

Hidden Layers: All the hidden layers in the VGG network use ReLU. VGG does not usually leverage Local Response Normalization (LRN) as it increases memory consumption and training time. Moreover, it makes no improvements to overall accuracy.

Fully-Connected Layers: The VGGNet has three fully connected layers. Out of the three layers, the first two have 4096 channels each, and the third has 1000 channels, 1 for each class.

Block 1 · Block 2 · Block 3 · Block 4 · Block 5 · Block 6
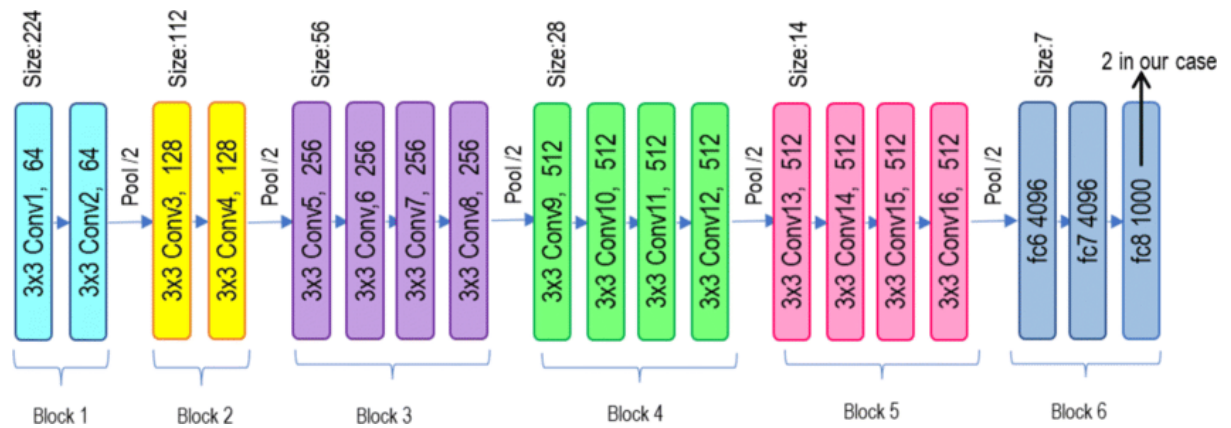
VGG-19 has 16 convolution layers grouped into 5 blocks. After every block, there is a Max pool layer that decreases the size of the input image by 2 and increases the number of filters of the convolution layer also by 2. The dimensions of the last three dense layers in block 6 are 4096, 4096, and 1000 respectively. VGG classifies the input images into 1000 different categories. As there are two output classes in this study the dimension of fc8 is set to two.

Feasibility study

A feasibility analysis evaluates the project's potential for success; therefore, perceived objectivity is an essential factor in the credibility of the study for potential investors and lending institutions.

Technical Feasibility

Technical resources need for project Development

Operating System	:	Windows family
Technology	:	Python 3.8
Front-end Technology	:	HTML, CSS, JS
Back-end Technology	:	MySQL
Code Development Tool	:	PyCharm IDE
Web framework	:	Flask

Economic Feasibility

Cost/ benefits analysis of the project as over project is academic project we will not have only basic cost for learning of the technologies

Operational Feasibility

This assessment involves undertaking a study to analyze and determine whether and how well the organization's needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.

Functional Requirements

- Admin Login
- Training
- Testing
- Prediction

NON- Functional Requirements

- Serviceability requirement
- Manageability requirement
- Recoverability requirement
- Security requirement

SOFTWARE OVER VIEW:

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Small Talk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Input as CSV File

Reading data from CSV (comma separated values) is a fundamental necessity in Data Science. Often, we get data from various sources which can get exported to CSV format so that they can be used by other systems. The Panadas library provides features using which we can read the CSV file in full as well as in parts for only a selected group of columns and rows.

The CSV file is a text file in which the values in the columns are separated by a comma. Let's consider the following data present in the file named input.csv. You can create this file using windows notepad by copying and pasting this data. Save the file as input.csv using the save As All files (*. *) option in notepad.

```python
import pandas as pd

data= pd. read_csv('path/input.csv')

print(data)
```

Operations using NumPy

NumPy is a Python package which stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Using NumPy, a developer can perform the following operations −

- Mathematical and logical operations on arrays.

Key Features of Pandas

- Fast and efficient Data Frame object with default and customized indexing.

- Tools for loading data into in-memory data objects from different file formats.

- Data alignment and integrated handling of missing data.

- Reshaping and pivoting of date sets.

- Label-based slicing, indexing and sub-setting of large data sets.

- Columns from a data structure can be deleted or inserted.

- Group by data for aggregation and transformations.

- High performance merging and joining of data.

- Time Series functionality.

Python Flask Tutorial

Flask Tutorial provides the basic and advanced concepts of the Python Flask framework. Our Flask tutorial is designed for beginners and professionals.

Flask is a web framework that provides libraries to build lightweight web applications in python. It is developed by **Armin Ronacher** who leads an international group of python enthusiasts (POCCO).

What is Flask?

Flask is a web framework that provides libraries to build lightweight web applications in python. It is developed by **Armin Ronacher** who leads an international group of python enthusiasts (POCCO). It is based on WSGI toolkit and jinja2 template engine. Flask is considered as a micro framework.

What is WSGI?

It is an acronym for web server gateway interface which is a standard for python web application development. It is considered as the specification for the universal interface between the web server and web application.
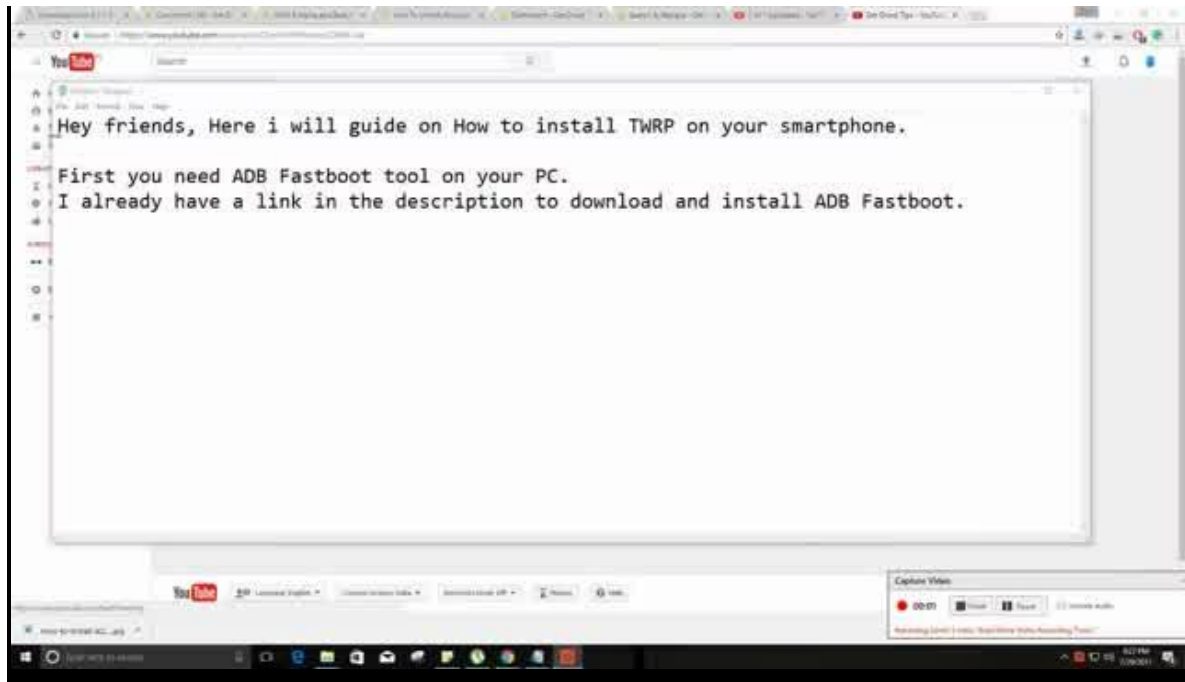
What is Jinja2?

Jinja2 is a web template engine which combines a template with a certain data source to render the dynamic web pages.

Flask Environment Setup

To install flask on the system, we need to have python 2.7 or higher installed on our system. However, we suggest using python 3 for the development in the flask.

Install virtual environment (virtual env)

Virtual env is considered as the virtual python environment builder which is used to create the multiple python virtual environment side by side. It can be installed by using the following command.

1. $ pip install virtualenv

Once it is installed, we can create the new virtual environment into a folder as given below.

1. $ mkdir new
2. $ cd new
3. $ virtualenv venv

To activate the corresponding environment, use the following command on the Linux operating system.

1. $ venv/bin/activate

On windows, use the following command.

1. $ venv\scripts\activate

We can now install the flask by using the following command.

1. $ pip install flask

However, we can install the flask using the above command without creating the virtual environment.

PyCharm Tutorial

PyCharm is the most popular IDE for Python, and includes great features such as excellent code completion and inspection with advanced debugger and support for web programming and various frameworks. PyCharm is created by Czech company, Jet brains which focusses on creating integrated development environment for various web development languages like JavaScript and PHP.

Audience

This tutorial has been prepared for Python developers who focus on using IDE with complete package of running, debugging and creating projects in various python frameworks. Also, interested learners with a basic knowledge of any IDE can take up this tutorial.

Prerequisites

Before proceeding with this tutorial, you need a basic knowledge of any integrated development environment of Python like Sublime Text or most popular IDE like NetBeans. If you are a beginner, we suggest you to go through tutorials related to these topics first before proceeding further on this tutorial. PyCharm is the most popular IDE used for Python scripting language. This chapter will give you an introduction to PyCharm and explains its features.

PyCharm offers some of the best features to its users and developers in the following aspects

- Code completion and inspection
- Advanced debugging
- Support for web programming and frameworks such as Django and Flask

Features of PyCharm

Besides, a developer will find PyCharm comfortable to work with because of the features mentioned below −

Code Completion

PyCharm enables smoother code completion whether it is for built in or for an external package.

SQL Alchemy as Debugger

You can set a breakpoint, pause in the debugger and can see the SQL representation of the user expression for SQL Language code.

Git Visualization in Editor

When coding in Python, queries are normal for a developer. You can check the last commit easily in PyCharm as it has the blue sections that can define the difference between the last commit and the current one.

Code Coverage in Editor

You can run **.py** files outside PyCharm Editor as well marking it as code coverage details elsewhere in the project tree, in the summary section etc.

Package Management

All the installed packages are displayed with proper visual representation. This includes list of installed packages and the ability to search and add new packages.

Local History

 Local History is always keeping track of the changes in a way that complements like Git. Local history in PyCharm gives complete details of what is needed to rollback and what is to be added.

Refactoring

Refactoring is the process of renaming one or more files at a time and PyCharm includes various shortcuts for a smooth refactoring process.

User Interface of PyCharm Editor

The user interface of PyCharm editor is shown in the screenshot given below. Observe that the editor includes various features to create a new project or import from an existing project. You can download the PyCharm Editor and read its official documentation at this link − https://www.jetbrains.com/pycharm/

CHAPTER 4

SYSTEM DESIGN

4.1 UML DIAGRAMS

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

Global Use Case Diagrams:

Identification of actors:

Actor: Actor represents the role a user plays with respect to the system. An actor interacts with, but has no control over the use cases.

Graphical representation:

An actor is someone or something that:

Interacts with or uses the system.

- Provides input to and receives information from the system.

- Is external to the system and has no control over the use cases.

Actors are discovered by examining:

- Who directly uses the system?

- Who is responsible for maintaining the system?

- External hardware used by the system.

- Other systems that need to interact with the system.

Questions to identify actors:

- Who is using the system? Or, who is affected by the system? Or, which groups need help from the system to perform a task?

- Who affects the system? Or, which user groups are needed by the system to perform its functions?

- And, finally, how do users use the system (use case)? What are they doing with the system?

The actors identified in this system are:

a.   System Administrator

b.   Customer

c.   Customer Care

Identification of use cases:

Use case: A use case can be described as a specific way of using the system from a user's (actor's) perspective.

Graphical representation:
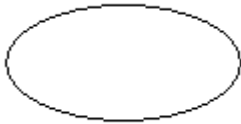
A more detailed description might characterize a use case as:

- Pattern of behaviour the system exhibits
- A sequence of related transactions performed by an actor and the system
- Delivering something of value to the actor

Use cases provide a means to:

- capture system requirements
- communicate with the end users and domain experts
- test the system

   Use cases are best discovered by examining the actors and defining what the actor will be able to do with the system.

Guide lines for identifying use cases:

- For each actor, find the tasks and functions that the actor should be able to perform or that the system needs the actor to perform. The use case should represent a course of events that leads to clear goal
- Name the use cases.
- Describe the use cases briefly by applying terms with which the user is familiar.

This makes the description less ambiguous

Questions to identify use cases:

- What are the tasks of each actor?

- Will any actor create, store, change, remove or read information in the system?

- What use cases will support and maintains the system?

Flow of Events

A flow of events is a sequence of transactions (or events) performed by the system. They typically contain very detailed information, written in terms of what the system should do, not how the system accomplishes the task. Flow of events are created as separate files or documents in your favourite text editor and then attached or linked to a use case using the Files tab of a model element.

A flow of events should include:

- When and how the use case starts and ends

- Use case/actor interactions

- Data needed by the use case

- Normal sequence of events for the use case

- Alternate or exceptional flows

Construction of Use case diagrams:

Use-case diagrams graphically depict system behaviour (use cases). These diagrams present a high-level view of how the system is used as viewed from an outsider's (actor's) perspective. A use-case diagram may depict all or some of the use cases of a system.

A use-case diagram can contain:

- actors ("things" outside the system)

- use cases (system boundaries identifying what the system should do)

- Interactions or relationships between actors and use cases in the system including the associations, dependencies, and generalizations.

Relationships in use cases:

1. Communication:

The communication relationship of an actor in a use case is shown by connecting the actor symbol to the use case symbol with a solid path. The actor is said to communicate with the use case.

2. Uses:

A Uses relationship between the use cases is shown by generalization arrow from the use case.

3. Extends:

The extend relationship is used when we have one use case that is similar to another use case but does a bit more. In essence it is like subclass.

4.2 SEQUENCE DIAGRAMS

A sequence diagram is a graphical view of a scenario that shows object interaction in a time-based sequence what happens first, what happens next. Sequence diagrams establish the roles of objects and help provide essential information to determine class responsibilities and interfaces.

There are two main differences between sequence and collaboration diagrams: sequence diagrams show time-based object interaction while collaboration diagrams show how objects associate with each other. A sequence diagram has two dimensions: typically, vertical placement represents time and horizontal placement represents different objects.

Object:

An object has state, behaviour, and identity. The structure and behaviour of similar objects are defined in their common class. Each object in a diagram indicates some instance of a class. An object that is not named is referred to as a class instance.

The object icon is similar to a class icon except that the name is underlined:

An object's concurrency is defined by the concurrency of its class.

Message:

A message is the communication carried between two objects that trigger an event. A message carries information from the source focus of control to the destination focus of control. The synchronization of a message can be modified through the message specification. Synchronization means a message where the sending object pauses to wait for results.

Link:

A link should exist between two objects, including class utilities, only if there is a relationship between their corresponding classes. The existence of a relationship between two classes symbolizes a path of communication between instances of the classes: one object may send messages to another. The link is depicted as a straight line between objects or objects and class instances in a collaboration diagram. If an object links to itself, use the loop version of the icon.

CLASS DIAGRAM:

Identification of analysis classes:

A class is a set of objects that share a common structure and common behaviour (the same attributes, operations, relationships and semantics). A class is an abstraction of real-world items.

There are 4 approaches for identifying classes:

a.   Noun phrase approach:

b.   Common class pattern approach.

c.   Use case Driven Sequence or Collaboration approach.

d.   Classes, Responsibilities and collaborators Approach

1.   Noun Phrase Approach:

The guidelines for identifying the classes:

*   Look for nouns and noun phrases in the use cases.

*   Some classes are implicit or taken from general knowledge.

*   All classes must make sense in the application domain; Avoid computer
    implementation classes – defer them to the design stage.

*   Carefully choose and define the class names After identifying the classes we have to eliminate the following types of classes:

*   Adjective classes.

2.   Common class pattern approach:

The following are the patterns for finding the candidate classes:

*   Concept class.

*   Events class.

*   Organization class

*   Peoples class

*   Places class

*   Tangible things and devices class.

3.   Use case driven approach:

We have to draw the sequence diagram or collaboration diagram. If there is need for some classes to represent some functionality then add new classes which perform those functionalities.

4.   CRC approach:

The process consists of the following steps:

- Identify classes' responsibilities (and identify the classes)
- Assign the responsibilities
- Identify the collaborators.

Identification of responsibilities of each class:

The questions that should be answered to identify the attributes and methods of a class respectively are:

a.   What information about an object should we keep track of?

b.   What services must a class provide?

Identification of relationships among the classes:

Three types of relationships among the objects are:

Association: How objects are associated?

Super-sub structure: How are objects organized into super classes and sub classes?

Aggregation: What is the composition of the complex classes?

Association:

The **questions** that will help us to identify the associations are:

a.   Is the class capable of fulfilling the required task by itself?

b.   If not, what does it need?

c.   From what other classes can it acquire what it needs?

Guidelines for identifying the tentative associations:

- A dependency between two or more classes may be an association. Association often corresponds to a verb or prepositional phrase.

- A reference from one class to another is an association. Some associations are implicit or taken from general knowledge.

   We have to eliminate the unnecessary association like implementation associations, ternary or n-array associations and derived associations.

Super-sub class relationships:

Super-sub class hierarchy is a relationship between classes where one class is the parent class of another class (derived class). This is based on inheritance.

Guidelines for identifying the super-sub relationship, a generalization are

1. **Top-down***:

   Look for noun phrases composed of various adjectives in a class name. Avoid excessive refinement. Specialize only when the sub classes have significant behaviour.

2.**Bottom-up***:*

Look for classes with similar attributes or methods. Group them by moving the common attributes and methods to an abstract class. You may have to alter the definitions a bit.

3.**Reusability***:*

Move the attributes and methods as high as possible in the hierarchy.

4. **Multiple inheritances***:*

Avoid excessive use of multiple inheritances. One way of getting benefits of multiple inheritances is to inherit from the most appropriate class and add an object of another class as an attribute.

Aggregation or a-part-of relationship:

It represents the situation where a class consists of several component classes. A class that is composed of other classes doesn't behave like its parts. It behaves very difficultly. The major properties of this relationship are transitivity and anti-symmetry.

The **questions** whose answers will determine the distinction between the part and whole relationships are:

- Does the part class belong to the problem domain?

- Is the part class within the system's responsibilities?

- Does the part class capture more than a single value? (If not then simply include it as an attribute of the whole class)

- Does it provide a useful abstraction in dealing with the problem domain?

There are three types of aggregation relationships. They are:

Assembly:

It is constructed from its parts and an assembly-part situation physically exists.

Container:

A physical whole encompasses but is not constructed from physical parts.

Collection member:

A conceptual whole encompasses parts that may be physical or conceptual. The container and collection are represented by hollow diamonds but composition is represented by solid diamond.

Use Case:
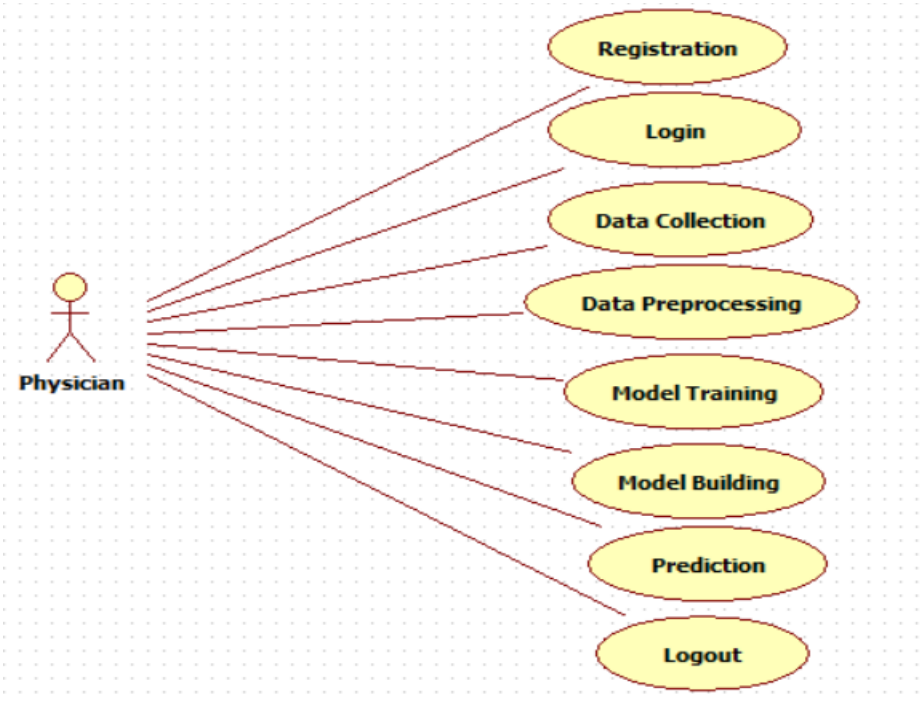
Sequence Diagram:
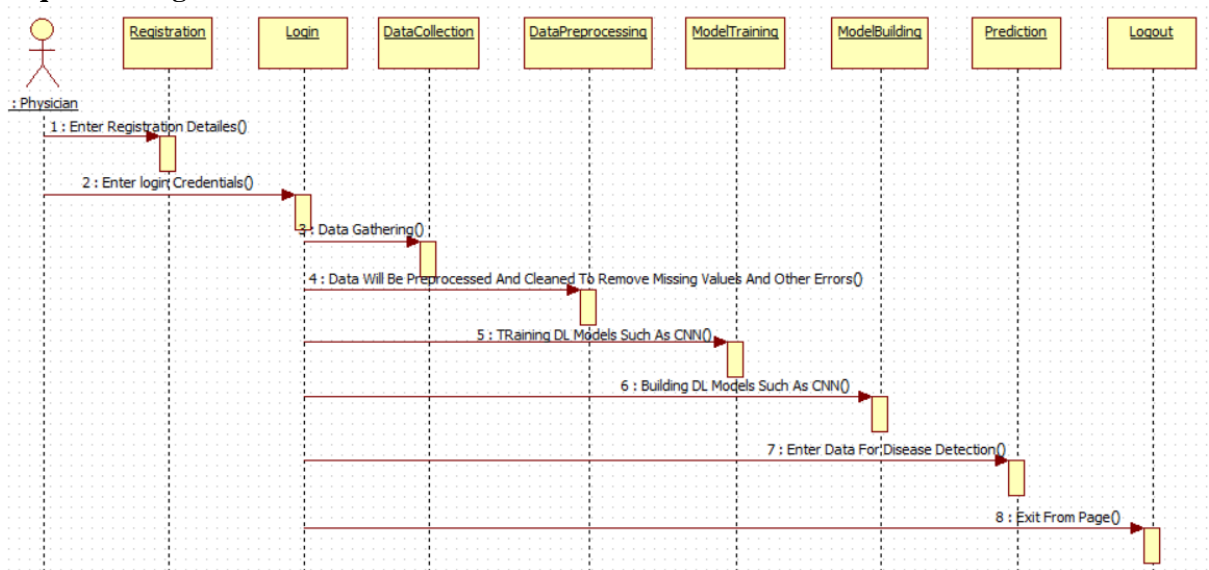
Collaboration:
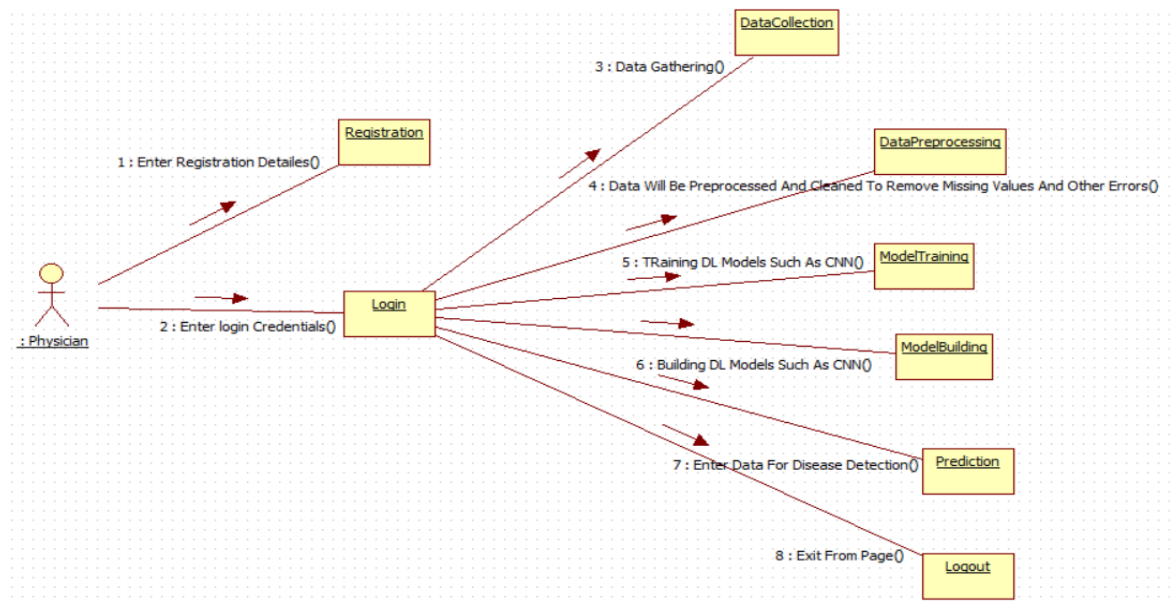
State Chart:
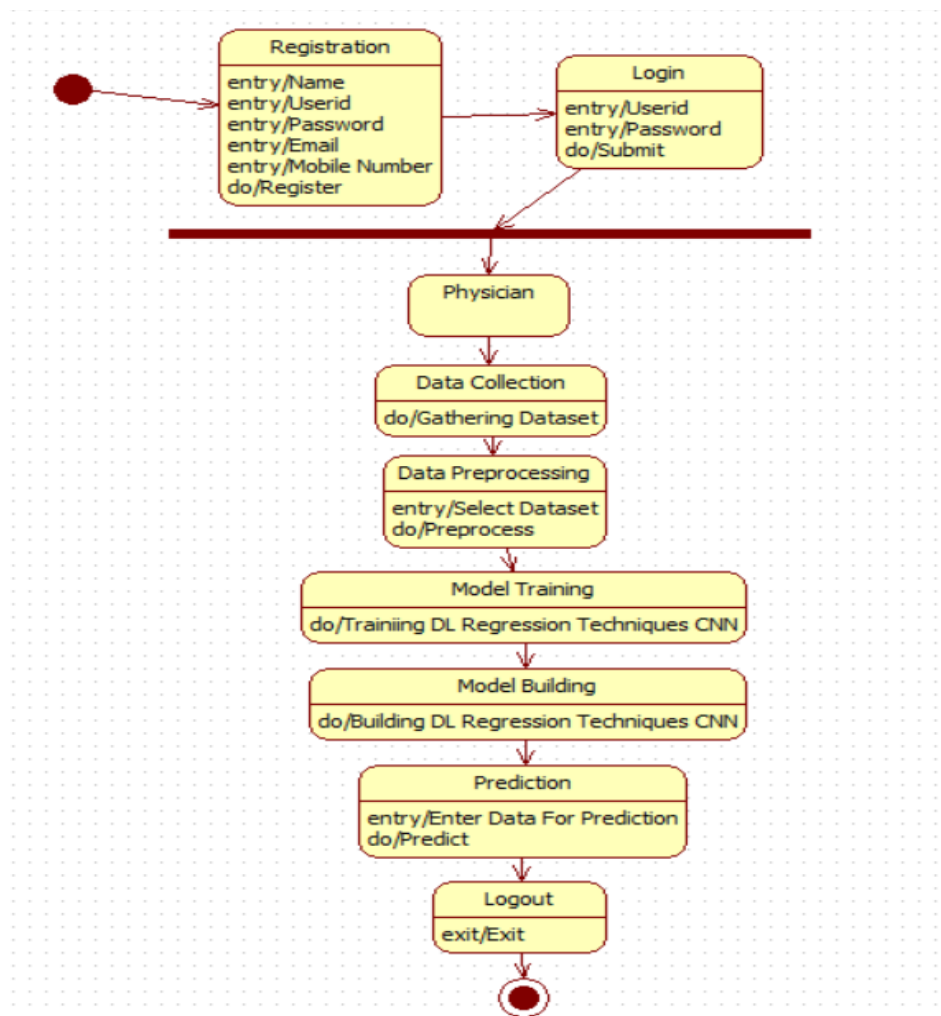
Component Diagram:
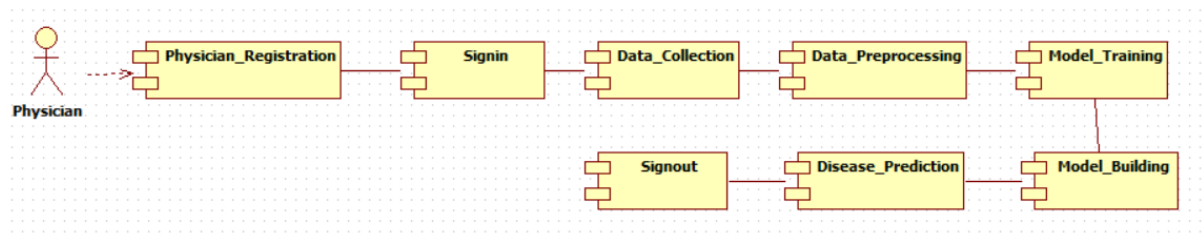
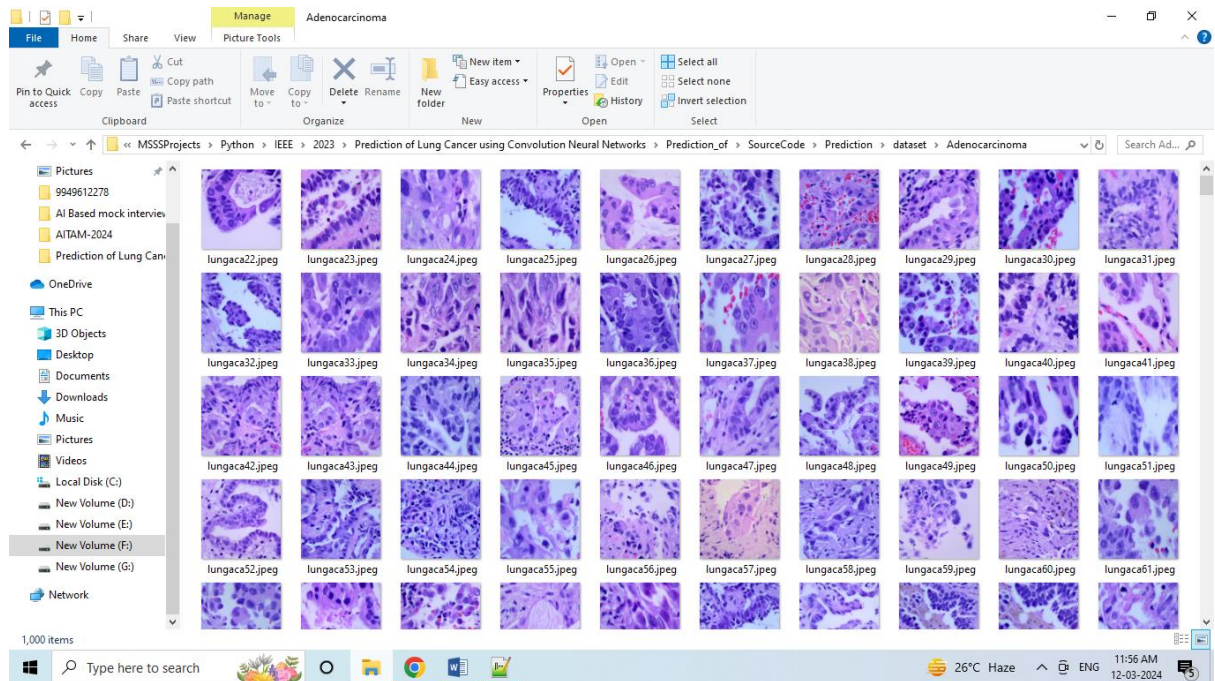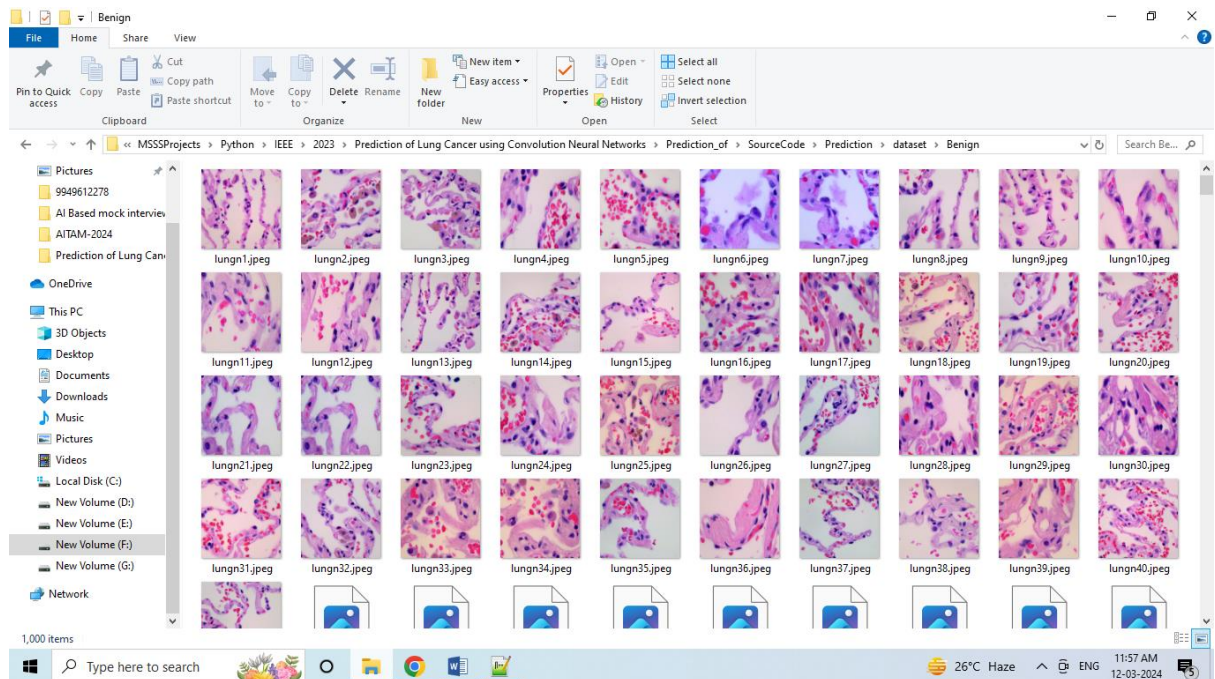**Deployment**
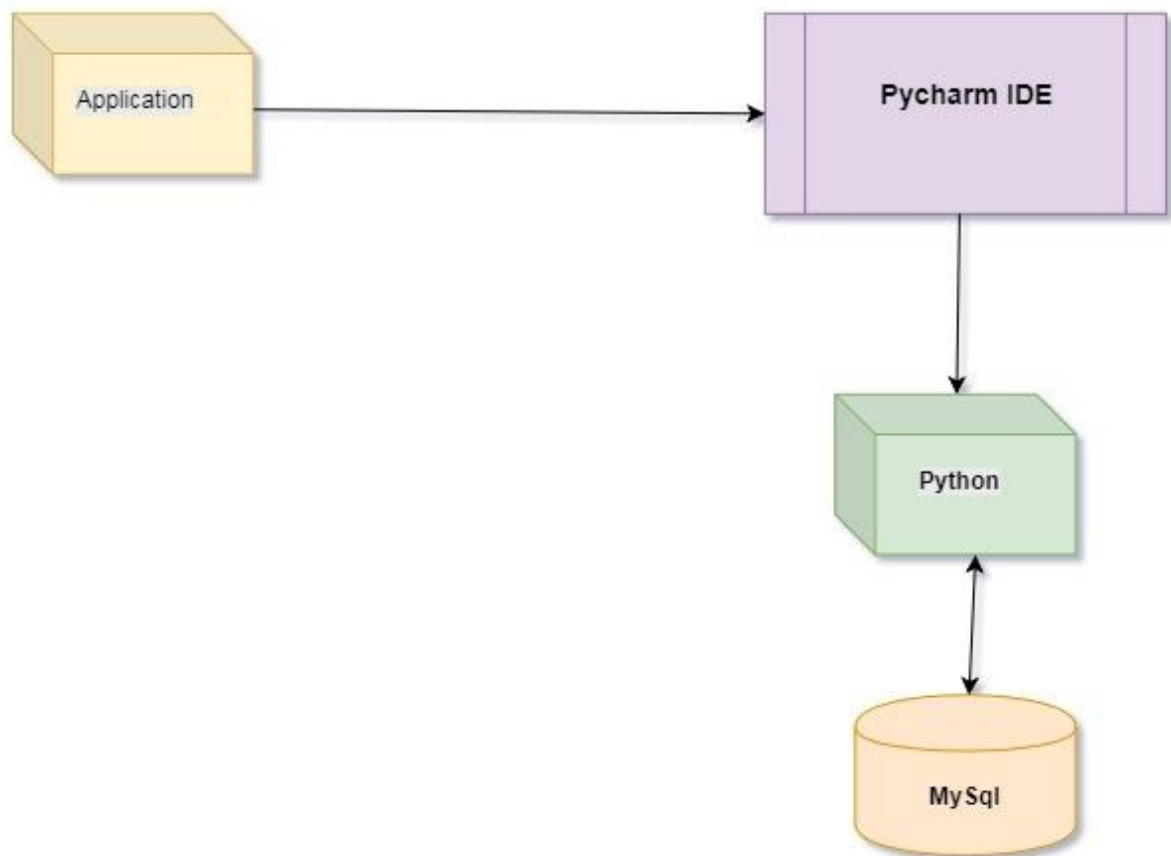
CHAPTER 5

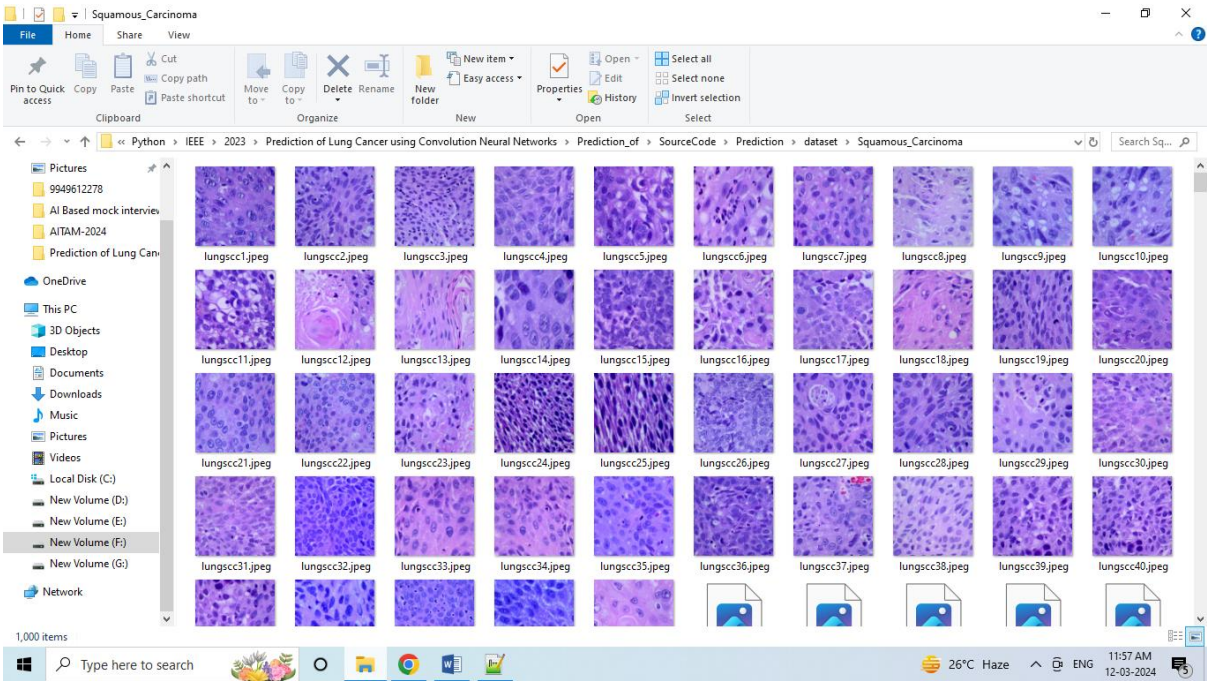# IMPLEMENTATION

Database & Dataset

Adenocarcinoma

Benign

Squamous_Carcinoma

About MySQL:

MySQL is a relational database management system (RDBMS)[1] that runs as a server providing multi-user access to a number of databases. The SQL phrase stands for Structured Query Language. Free-software- open-source projects that require a full-featured database management system often use MySQL. For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, Joomla, WordPress, phpBB, Drupal and other software built on the LAMP software stack. MySQL is also used in many high-profile, large-scale World Wide Web products, including Wikipedia, Google, Facebook, and Twitter.

MySQL is the world's most popular open-source database software, with over 100 million copies of its software downloaded or distributed throughout its history. With its superior speed, reliability, and ease of use, MySQL has become the preferred choice for Web, Web 2.0, SaaS, ISV, Telecom companies and forward-thinking corporate IT Managers because it eliminates the major problems associated with downtime, maintenance and administration for modern, online applications.

Many of the world's largest and fastest-growing organizations use MySQL to save time and money powering their high-volume Web sites, critical business systems, and packaged software including industry leaders such as Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube, Wikipedia, and Booking.com.

The flagship MySQL offering is MySQL Enterprise, a comprehensive set of production-tested software, proactive monitoring tools, and premium support services available in an affordable annual subscription.

MySQL is a key part of LAMP (Linux, Apache, MySQL, PHP / Perl / Python), the fast-growing open-source enterprise software stack. More and more companies are using LAMP as an alternative to expensive proprietary software stacks because of its lower cost and freedom from platform lock-in.

MySQL was originally founded and developed in Sweden by two Swedes and a Finn: David Axmark, Allan Larsson and Michael "Monty" Widenius, who had worked together since the 1980's. More historical information on MySQL is

```sql
create database if not exists `lungcancer_detection`;
USE `lungcancer_detection`;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*Table structure for table `evaluations` */
DROP TABLE IF EXISTS `evaluations`;
CREATE TABLE `evaluations` (
  `accuracy` varchar(100) DEFAULT NULL,
  `loss` varchar(500) DEFAULT NULL,
  `precision` varchar(500) DEFAULT NULL,
  `recall` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*Data for the table `evaluations` */
insert  into `evaluations`(`accuracy`,`loss`,`precision`,`recall`) values
('0.9383333325386047','0.17991961538791656','0.9379194378852844','0.9316666722297
668');
/*Table structure for table `physician` */
DROP TABLE IF EXISTS `physician`;
CREATE TABLE `physician` (
  `name` varchar(100) DEFAULT NULL,
  `username` varchar(100) DEFAULT NULL,
  `passwrd` varchar(100) DEFAULT NULL,
  `email` varchar(100) DEFAULT NULL,
  `mno` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*Data for the table `physician` */
insert  into `physician`(`name`,`username`,`passwrd`,`email`,`mno`) values
('ali','ali','ali','ali@gmail.com','8121953811');
```

SOFTWARE TESTING

Software testing is one of the main stages of project development life cycle to provide our cessation utilizer with information about the quality of the application and ours, in our Project we have under gone some stages of testing like unit testing where it's done in development stage of the project when we are in implementation of the application after the Project is yare we have done manual testing with different Case of all the different modules in the application we have even done browser compatibility testing in different web browsers in market, even we have done Client side validation testing on our application

Unit testing

The unit testing is done in the stage of implementation of the project only the error is solved in development stage some of the error we come across in development are given below

TESTING

Testing is the debugging program is one of the most critical aspects of the computer programming triggers, without programming that works, the system would never produce an output of which it was designed. Testing is best performed when user development is asked to assist in identifying all errors and bugs. The sample data are used for testing. It is not quantity but quality of the data used the matters of testing. Testing is aimed at ensuring that the system was accurately an efficiently before live operation commands.

Testing objectives:

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say, testing is a process of executing a program with intent of finding an error.

Code testing:

This examines the logic of the program. For example, the logic for updating various sample data and with the sample files and directories were tested and verified.

Specification Testing:

Executing this specification starting what the program should do and how it should perform under various conditions. Test cases for various situation and combination of conditions in all the modules are tested.

Unit testing:

In the unit testing we test each module individually and integrate with the overall system. Unit testing focuses verification efforts on the smallest unit of software design in the module. This is also known as module testing. The module of the system is tested separately. This testing is carried out during programming stage itself. In the testing step each module is found to work satisfactorily as regard to expected output from the module. There are some validation checks for fields also. For example, the validation check is done for varying the user input given by the user which validity of the data entered. It is very easy to find error debut the system.

Each Module can be tested using the following two Strategies:

1. Black Box Testing
2. White Box Testing

BLACK BOX TESTING

What is Black Box Testing?

Black box testing is a software testing technique in which **functionality of the software under test (SUT) is tested without looking at the internal code structure**, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on the software requirements and specifications.

In Black Box Testing we just focus on inputs and output of the software system without bothering about internal knowledge of the software program.

The above Black Box can be any software system you want to test. For example: an operating system like Windows, a website like Google, a database like Oracle or even your own custom application.

30

Black box testing - Steps

Here are the generic steps followed to carry out any type of Black Box Testing.

- Initially requirements and specifications of the system are examined.
- Tester chooses valid inputs (positive test scenario) to check whether SUT processes them correctly. Also, some invalid inputs (negative test scenario) are chosen to verify that the SUT is able to detect them.
- Tester determines expected outputs for all those inputs.
- Software tester constructs test cases with the selected inputs.
- The test cases are executed.
- Software tester compares the actual outputs with the expected outputs.
- Defects if any are fixed and re-tested.

Types of Black Box Testing

There are many types of Black Box Testing but following are the prominent ones -

- **Functional testing** – This black box testing type is related to functional requirements of a system; it is done by software testers.
- **Non-functional testing** – This type of black box testing is not related to testing of a specific functionality, but non-functional requirements such as performance, scalability, usability.
- **Regression testing** – Regression testing is done after code fixes, upgrades or any other system maintenance to check the new code has not affected the existing code.

WHITE BOX TESTING

White Box Testing is the testing of a software solution's internal coding and infrastructure. It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability. White box testing is also known as **clear, open, structural, and glass box testing**.

It is one of two parts of the **"box testing" approach** of software testing. Its counterpart, Blackbox testing, involves testing from an external or end-user type perspective. On the other hand, Whitebox testing is based on the inner workings of an application and revolves around internal testing.

What do you verify in White Box Testing?

White box testing involves the testing of the software code for the following:

- Internal security holes
- Broken or poorly structured paths in the coding processes
- The flow of specific inputs through the code
- Expected output
- The functionality of conditional loops
- Testing of each statement, object and function on an individual basis

The testing can be done at system, integration and unit levels of software development. One of the basic goals of Whitebox testing is to verify a working flow for an application. It involves testing a series of predefined inputs against expected or desired outputs so that when a specific input does not result in the expected output, you have encountered a bug.

How do you perform White Box Testing?

To give you a simplified explanation of white box testing, we have divided it into two basic steps. This is what testers do when testing an application using the white box testing technique:

STEP 1) UNDERSTAND THE SOURCE CODE

The first thing a tester will often do is learn and understand the source code of the application. Since white box testing involves the testing of the inner workings of an application, the tester must be very knowledgeable in the programming languages used in the applications they are testing. Also, the testing person must be highly aware of secure coding practices. Security is often one of the primary objectives of testing software. The tester should be able to find security issues and prevent attacks from hackers and naive users who might inject malicious code into the application either knowingly or unknowingly.

Step 2) CREATE TEST CASES AND EXECUTE

The second basic step to white box testing involves testing the application's source code for proper flow and structure. One way is by writing more code to test the application's source code.

TemplateNotFound

```
jinja2.exceptions.TemplateNotFound: admin.html
```

Traceback (most recent call last)

File "C:\Users\ALI\AppData\Local\Programs\Python\Python38\lib\site-packages\flask\app.py", line *2551*, in __call__
```
return self.wsgi_app(environ, start_response)
```

File "C:\Users\ALI\AppData\Local\Programs\Python\Python38\lib\site-packages\flask\app.py", line *2531*, in wsgi_app
```
response = self.handle_exception(e)
```

File "C:\Users\ALI\AppData\Local\Programs\Python\Python38\lib\site-packages\flask\app.py", line *2528*, in wsgi_app
```
response = self.full_dispatch_request()
```

File "C:\Users\ALI\AppData\Local\Programs\Python\Python38\lib\site-packages\flask\app.py", line *1825*, in full_dispatch_request
```
rv = self.handle_user_exception(e)
```

File "C:\Users\ALI\AppData\Local\Programs\Python\Python38\lib\site-packages\flask\app.py", line *1823*, in full_dispatch_request
```
rv = self.dispatch_request()
```

File "C:\Users\ALI\AppData\Local\Programs\Python\Python38\lib\site-packages\flask\app.py", line *1799*, in dispatch_request
```
return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args)
```

File "C:\Users\ALI\Documents\Python\BigMart_SalesPrediction\venv\BM_SalesPrediction\index.py", line *36*, in admin
```
return render_template("admin.html")
```

File "C:\Users\ALI\AppData\Local\Programs\Python\Python38\lib\site-packages\flask\templating.py", line *146*, in render_template
```
template = app.jinja_env.get_or_select_template(template_name_or_list)
```

File "C:\Users\ALI\AppData\Local\Programs\Python\Python38\lib\site-packages\jinja2\environment.py", line *1081*, in get_or_select_template
```
return self.get_template(template_name_or_list, parent, globals)
```

File "C:\Users\ALI\AppData\Local\Programs\Python\Python38\lib\site-packages\jinja2\environment.py", line *1010*, in get_template
```
return self._load_template(name, globals)
```

BuildError

```
werkzeug.routing.exceptions.BuildError: Could not build url for endpoint ''. Did you mean 'static' instead?
```

Traceback (most recent call last)

File "C:\Users\ALI\AppData\Local\Programs\Python\Python38\lib\site-packages\flask\app.py", line *2551*, in __call__
```
return self.wsgi_app(environ, start_response)
```

File "C:\Users\ALI\AppData\Local\Programs\Python\Python38\lib\site-packages\flask\app.py", line *2531*, in wsgi_app
```
response = self.handle_exception(e)
```

File "C:\Users\ALI\AppData\Local\Programs\Python\Python38\lib\site-packages\flask\app.py", line *2528*, in wsgi_app
```
response = self.full_dispatch_request()
```

File "C:\Users\ALI\AppData\Local\Programs\Python\Python38\lib\site-packages\flask\app.py", line *1825*, in full_dispatch_request
```
rv = self.handle_user_exception(e)
```

File "C:\Users\ALI\AppData\Local\Programs\Python\Python38\lib\site-packages\flask\app.py", line *1823*, in full_dispatch_request
```
rv = self.dispatch_request()
```

File "C:\Users\ALI\AppData\Local\Programs\Python\Python38\lib\site-packages\flask\app.py", line *1799*, in dispatch_request
```
return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args)
```

File "C:\Users\ALI\Documents\Python\BigMart_SalesPrediction\venv\BM_SalesPrediction\index.py", line *72*, in data_preprocessing
```
return render_template("data_preprocessing.html",msg="Data Preprocessing Completed..!")
```

File "C:\Users\ALI\AppData\Local\Programs\Python\Python38\lib\site-packages\flask\templating.py", line *147*, in render_template
```
return _render(app, template, context)
```

File "C:\Users\ALI\AppData\Local\Programs\Python\Python38\lib\site-packages\flask\templating.py", line *130*, in _render
```
rv = template.render(context)
```

File "C:\Users\ALI\AppData\Local\Programs\Python\Python38\lib\site-packages\jinja2\environment.py", line *1301*, in render
```
self.environment.handle_exception()
```

File "C:\Users\ALI\AppData\Local\Programs\Python\Python38\lib\site-packages\jinja2\environment.py", line *936*, in handle_exception

TypeError Testing:
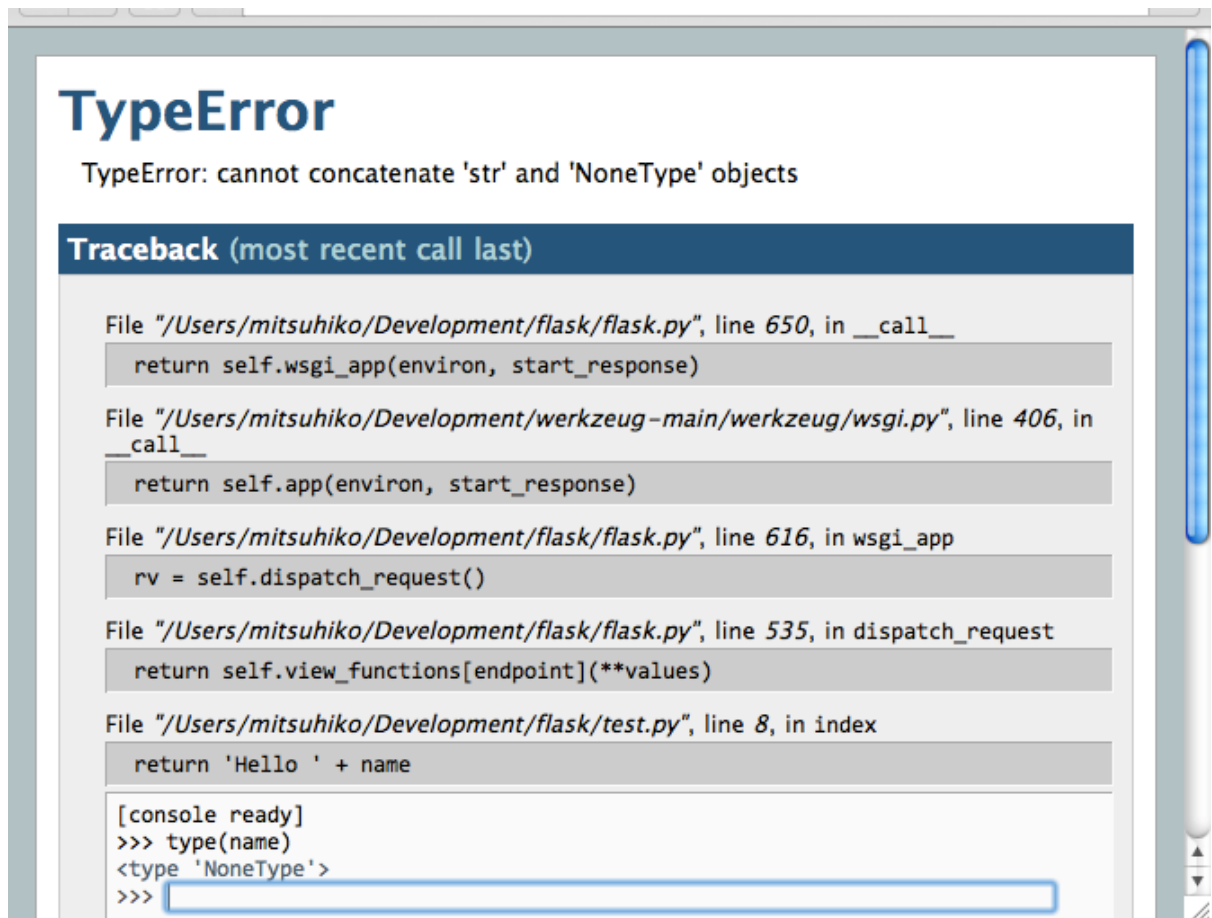
Name Error Testing:

Exception:

TemplateNotFound:

SCREEN SHORTS

Home Screen
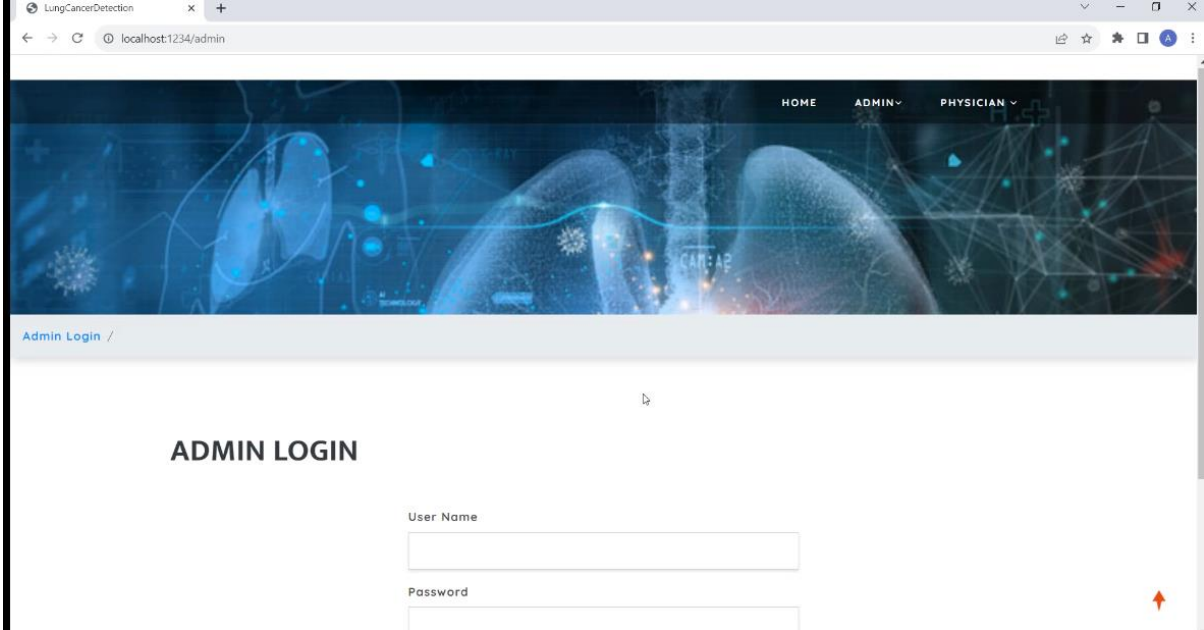
Admin Login
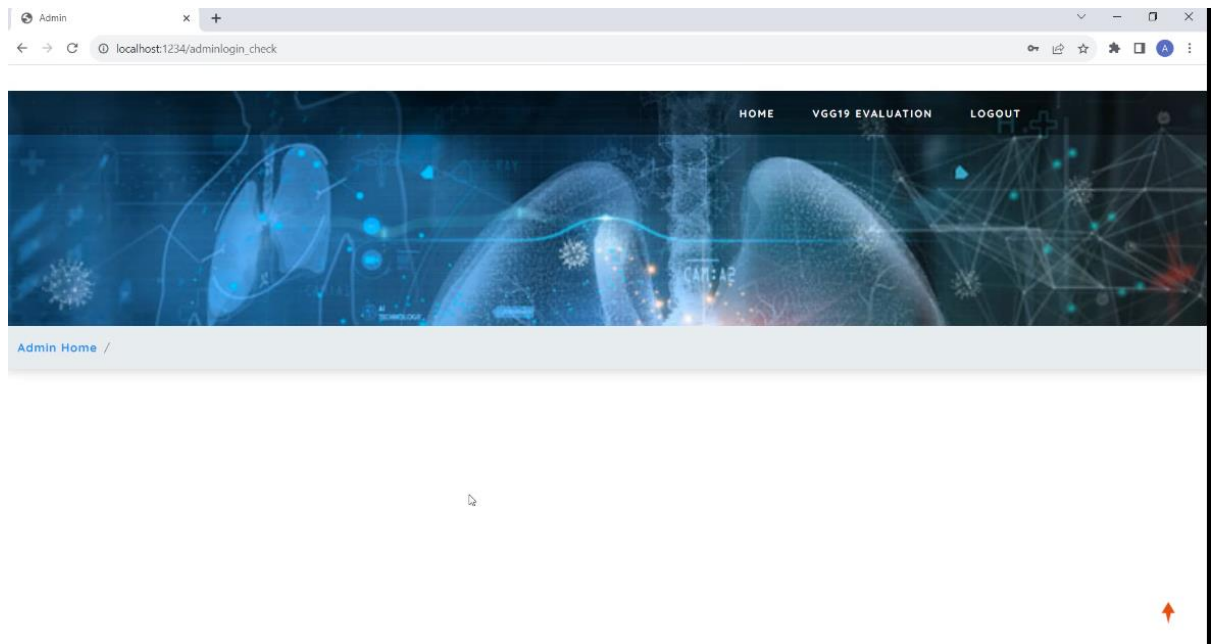
Admin Home
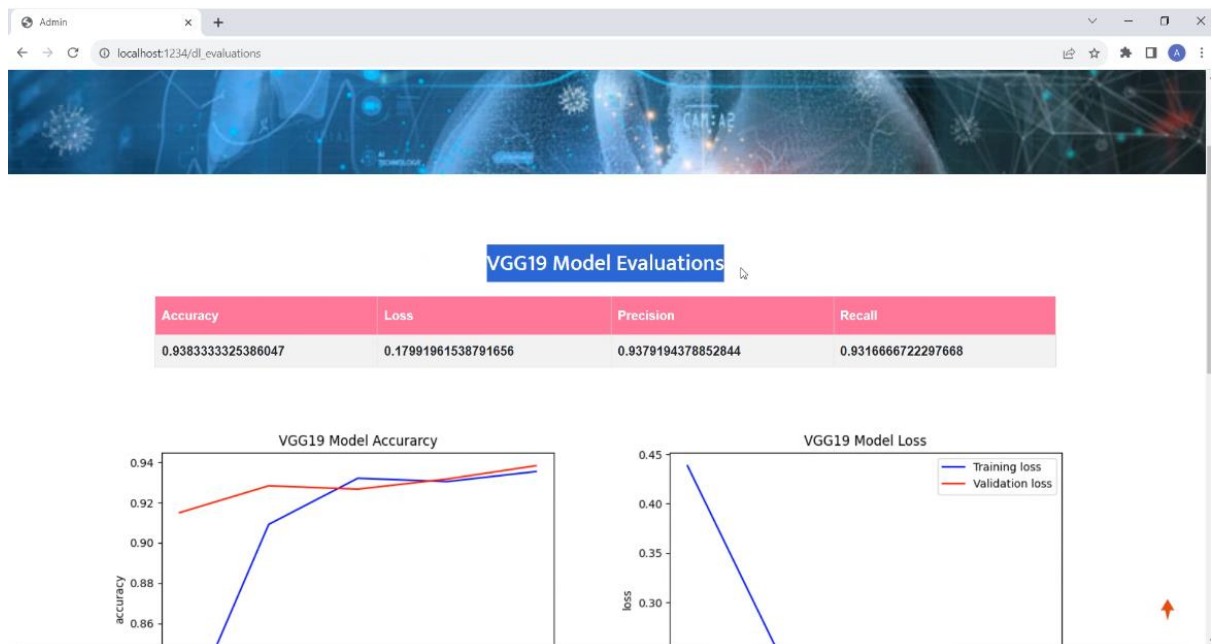
VGG-19 Model Evaluations

User registration

User Login

Prediction page
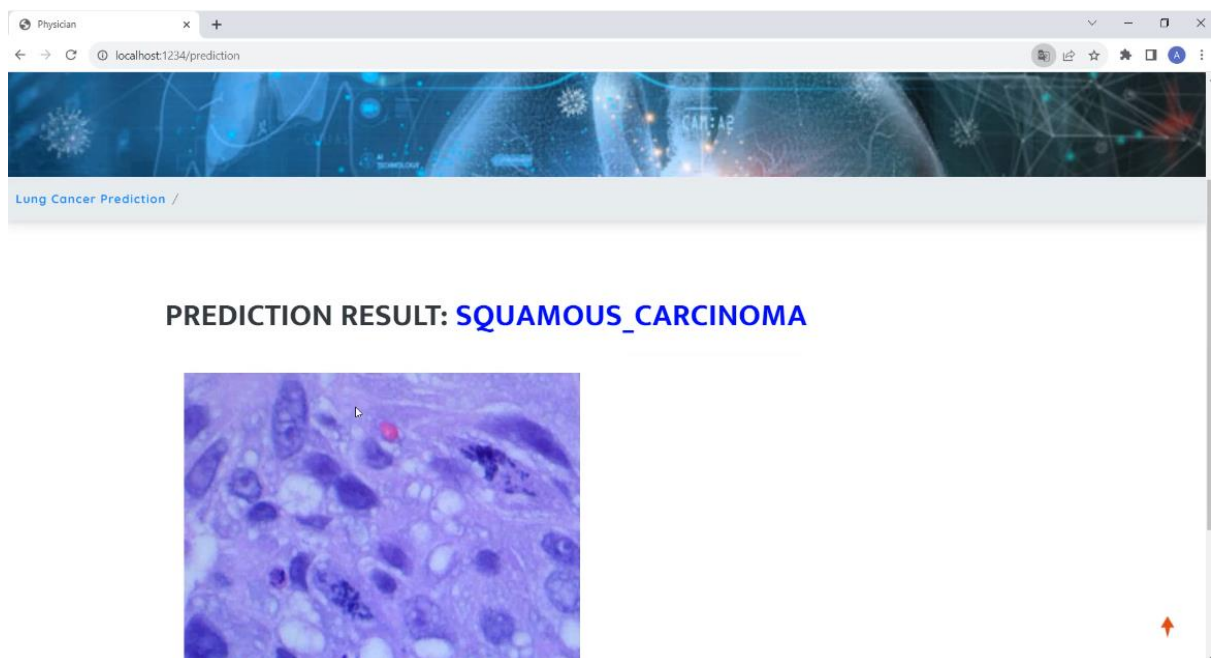
5.1 SOURCE CODE

Sample Code
DBConfig.py

```python
import mysql.connector
class DBConnection:
    @staticmethod
    def getConnection():
        database = mysql.connector.connect(host="localhost", user="root", passwd="root",
db='lungcancer_detection')
        return database


if __name__=="__main__":
    print(DBConnection.getConnection())
```

LungCancer_Prediction.py

```python
import os
import sys
import numpy as np
import operator
import pickle
from keras.models import Sequential, load_model
import cv2
from tensorflow.keras.utils import img_to_array
from keras.preprocessing import image as image_utils
import numpy
from keras.preprocessing import image
from tensorflow.keras.utils import load_img

def prediction_image(test_image):
    try:

        data = []
        img_path = test_image

        testing_img=cv2.imread(img_path)
        cv2.imwrite("../Prediction/static/lr_detection.jpg", testing_img)

        model_path = 'vgg19_model.h5'
        model = load_model(model_path)

        test_image = load_img(test_image, target_size=(128, 128))
        test_image = img_to_array(test_image)
        test_image = np.expand_dims(test_image, axis=0)
        test_image /= 255
        prediction = model.predict(test_image)
        lb = pickle.load(open('label_transform.pkl', 'rb'))
        prediction_result=lb.inverse_transform(prediction)[0]

        print(prediction_result)
```

```python
            return prediction_result

        except Exception as e:
            print("Error=", e)
            tb = sys.exc_info()[2]
            print("LINE NO: ", tb.tb_lineno)

'''testimage="lungaca27.jpeg"
prediction_image(testimage)'''
```

Training_VGG19.py
```python
import os
import numpy as np
import numpy as np
import pickle
import cv2
from keras.applications.vgg19 import VGG19
import keras
from os import listdir
from sklearn.preprocessing import LabelBinarizer
from keras.models import Sequential
#from keras.layers.normalization import BatchNormalization
from tensorflow.keras.layers import BatchNormalization
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
from keras.layers.core import Activation, Flatten, Dropout, Dense
from keras import backend as K
from keras.preprocessing.image import ImageDataGenerator
from keras.optimizers import Adam
from keras.preprocessing import image
#from keras.preprocessing.image import img_to_array
from keras.metrics import Recall,Precision
from tensorflow.keras.utils import img_to_array,load_img
from sklearn.preprocessing import MultiLabelBinarizer
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt2
import matplotlib.pyplot as plt3
import matplotlib.pyplot as plt4
from sklearn.metrics import f1_score, precision_score, accuracy_score,
recall_score,confusion_matrix
from DBConfig import DBConnection
from keras.preprocessing import image
import sys
def build_vgg19():
    try:
        database = DBConnection.getConnection()
        cursor = database.cursor()
        EPOCHS = 5
        BS = 32
```

```
print("[INFO] Loading Training dataset images...")
DIRECTORY = "..\\Prediction\\dataset"




CATEGORIES=['Adenocarcinoma','Benign','Squamous_Carcinoma']
image_data = []
target_class = []

for category in CATEGORIES:
    print(category)
    path = os.path.join(DIRECTORY, category)
    print(path)
    for img in os.listdir(path):
        img_path = os.path.join(path, img)
        img = load_img(img_path, target_size=(128,128))
        img = img_to_array(img)
        #img = img / 255
        image_data.append(img)
        target_class.append(category)

label_binarizer = LabelBinarizer()
image_labels = label_binarizer.fit_transform(target_class)
pickle.dump(label_binarizer, open('label_transform.pkl', 'wb'))
n_classes = len(label_binarizer.classes_)
print(n_classes)
np_image_list = np.array(image_data, dtype=np.float16) / 225.0

x_train, x_test, y_train, y_test = train_test_split(np_image_list, image_labels,
test_size=0.2, random_state=42)

# Model Initialization
base_model=VGG19(include_top=False,input_shape=(128,128,3))
base_model.trainable=False

classifier=keras.models.Sequential()
classifier.add(base_model)
classifier.add(Flatten())
classifier.add(Dense(3,activation='softmax'))


classifier.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy',Re
call(),Precision()])

print("[INFO] training network...")

aug = ImageDataGenerator(
    rotation_range=25, width_shift_range=0.1,
    height_shift_range=0.1, shear_range=0.2,
    zoom_range=0.2, horizontal_flip=True,
    fill_mode="nearest")
```

```python
history = classifier.fit_generator(
    aug.flow(x_train, y_train, batch_size=BS),
    validation_data=(x_test, y_test),

  steps_per_epoch=len(x_train) // BS,
    epochs=EPOCHS, verbose=1
)

acc = history.history['accuracy']
'''val_acc = history.history['val_accuracy']
loss = history.history['loss']
val_loss = history.history['val_loss']

precision=history.history['precision']
val_precision = history.history['val_precision']

recall = history.history['recall']
val_recall = history.history['val_recall']'''

epochs = range(1, len(acc) + 1)
# Train and validation accuracy
plt.plot(epochs, history.history['accuracy'], 'b', label='Training accurarcy')
plt.plot(epochs, history.history['val_accuracy'], 'r', label='Validation accurarcy')
plt.title('VGG19 Model Accurarcy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(loc='lower right')
plt.savefig('static/accuracy.png')
plt.show()


# Train and validation loss
plt2.plot(epochs,  history.history['loss'], 'b', label='Training loss')
plt2.plot(epochs, history.history['val_loss'], 'r', label='Validation loss')
plt2.title('VGG19 Model Loss')
plt2.ylabel('loss')
plt2.xlabel('epoch')
plt2.legend()
plt2.savefig('static/loss.png')
plt2.show()

plt3.plot(epochs, history.history['precision'],'b', label='Training loss')
plt3.plot(epochs, history.history['val_precision'], 'r', label='Validation loss')
plt3.title('VGG19 Model Precision')
plt3.ylabel('precision')
plt3.xlabel('epoch')
plt3.legend(['Training', 'Validation'], loc='lower right')
plt3.savefig('static/precision.png')
plt3.show()
```

```python
        plt4.plot(epochs,history.history['recall'],'b')
        plt4.plot(epochs,history.history['val_recall'],'r')
        plt4.title('VGG19 Model Recall')
        plt4.ylabel('recall')
        plt4.xlabel('epoch')

        plt4.legend(['Training', 'Validation'], loc='lower right')
        plt4.savefig('static/recall.png')
        plt4.show()
        print("[INFO] Calculating VGG19 model accuracy")
        scores = classifier.evaluate(x_test, y_test)
        print("Test Accuracy:",scores)
        #vgg19_accuracy=scores[1]*100
        #print(vgg19_accuracy)'''
        print("Training Completed..!")
        loss = scores[0]
        acc = scores[1]
        recall =scores[2]
        precision=scores[3]

        # save the model to disk
        #print("[INFO] Saving model...")
        classifier.save('vgg19_model.h5')
        cursor.execute("delete from evaluations")

        sql = "insert into evaluations
values('"+str(acc)+"','"+str(loss)+"','"+str(precision)+"','"+str(recall)+"')"
        cursor.execute(sql)
        database.commit()



    except Exception as e:
        print("Error=" , e)
        tb = sys.exc_info()[2]
        print(tb.tb_lineno)

build_vgg19()
```
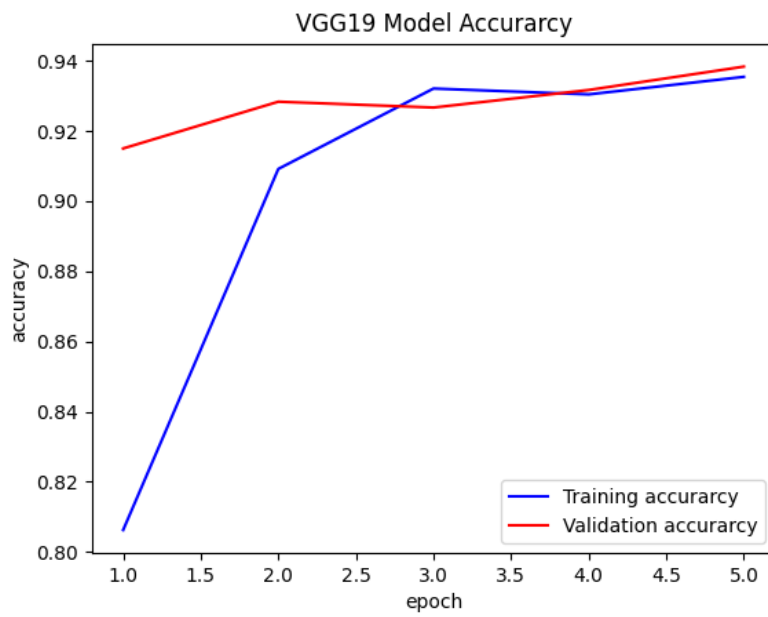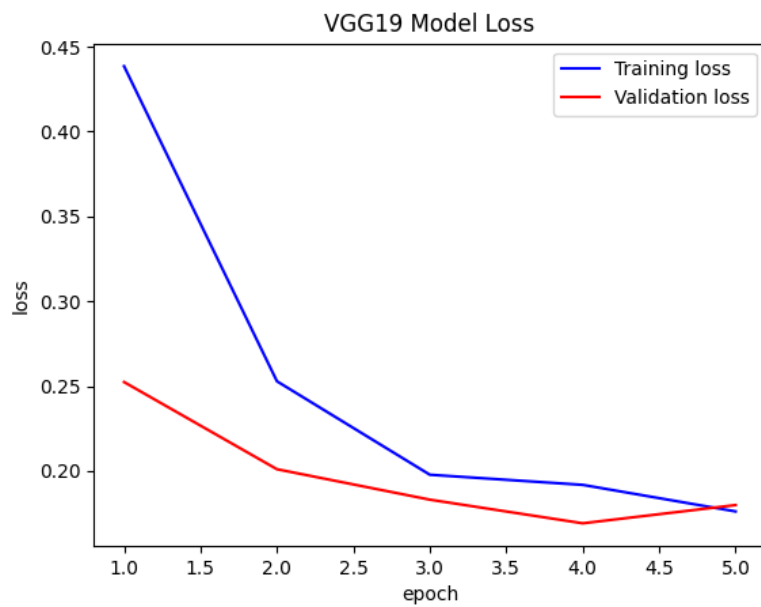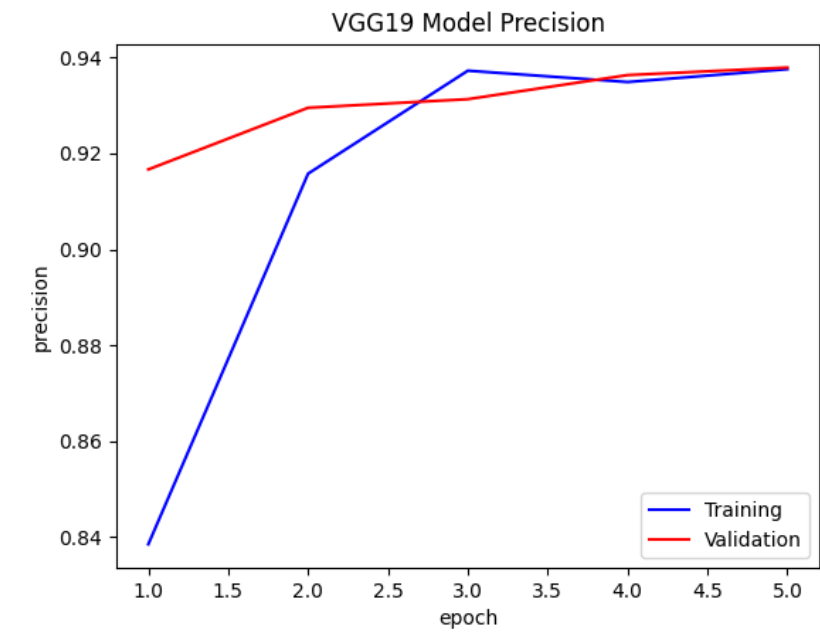
CHAPTER 6

RESULTS

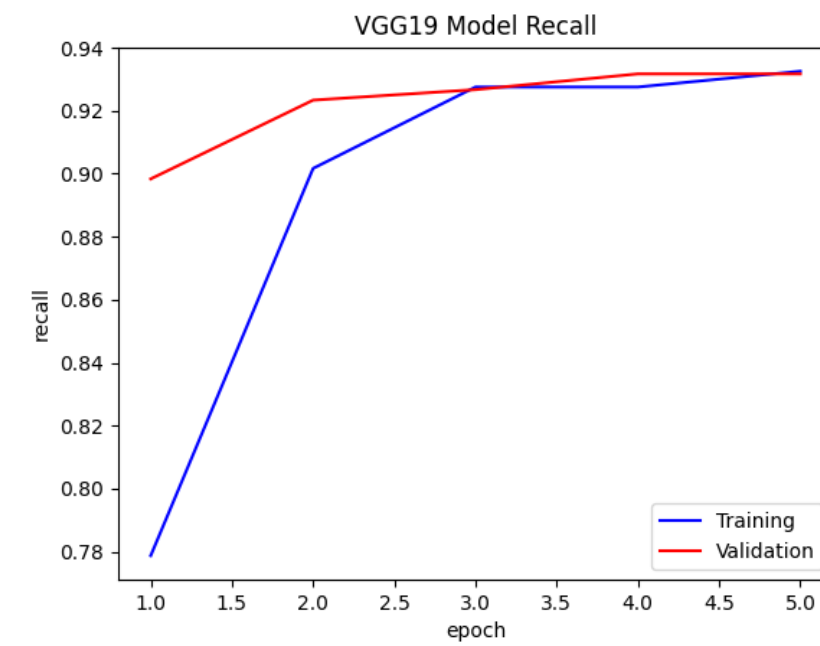**Test Results**

Accuracy

Loss

Precision

Recall

CHAPTER 7

CONCLUSION

We used a pre-trained system (VGG19) to analyse lung scans and predict cancer. Our system achieved an accuracy of (91%) in identifying lung cancer. This method is helpful because it saves time and works well. There's room for improvement, like fine-tuning the system and using more data. Overall, this approach shows promise for helping doctors find lung cancer.

The data-set when applied VGG-16 algorithm to check the accuracy and other metrics, resulted in the provided outcomes: The best result predicted with the model is provided. The figure 5 predicts accuracy and loss, figure 6 predicts AUC and precision and figure 7 predicts precision and F1 score. The uphill and downfall has been shown in the figures provided.

REFERENCES

[1] M. Norouzi and P. Hardy, "Clinical applications of nanomedicines in lung cancer treatment," Acta Biomaterialia, vol. 121, pp. 134–142, 2021.

[2] P. H. Viale, "The American cancer society's facts & figures: 2020 edition," Journal of the Advanced Practitioner in Oncology, vol. 11, no. 2, p. 135, 2020.

[3] D. G. Beer, S. L. Kardia, C.-C. Huang, T. J. Giordano, A. M. Levin,
D. E. Misek, L. Lin, G. Chen, T. G. Gharib, D. G. Thomas, et al., "Gene-expression profiles predict survival of patients with lung adeno- carcinoma," Nature medicine, vol. 8, no. 8, pp. 816–824, 2002.

[4] C.-R. Guo, Y. Mao, F. Jiang, C.-X. Juan, G.-P. Zhou, and N. Li, "Com-putational detection of a genome instability-derived lncrna signature for predicting the clinical outcome of lung adenocarcinoma," Cancer Medicine, vol. 11, no. 3, pp. 864–879, 2022.

[5] M. A. Gillette, S. Satpathy, S. Cao, S. M. Dhanasekaran, S. V. Vasaikar, K. Krug, F. Petralia, Y. Li, W.-W. Liang, B. Reva, et al., "Proteogenomic characterization reveals therapeutic vulnerabilities in lung adenocarcinoma," Cell, vol. 182, no. 1, pp. 200–225, 2020.

[6] A. Agaimy, O. Daum, M. Michal, M. W. Schmidt, R. Stoehr, A. Hartmann, and G. Y. Lauwers, "Undifferentiated large cell/rhabdoid carcinoma presenting in the intestines of patients with concurrent or recent non-small cell lung cancer (nsclc): clinicopathologic and molecular analysis of 14 cases indicates an unusual pattern of dedifferentiated metastases," Virchows Archiv, pp. 1–11, 2021.

[7] K. I. Tosios, V. Papanikolaou, D. Vlachodimitropoulos, and N. Goutas, "Primary large cell neuroendocrine carcinoma of the parotid gland. report of a rare case," Head and Neck Pathology, pp. 1–8, 2021.

[8] B.-Y. Wang, J.-Y. Huang, H.-C. Chen, C.-H. Lin, S.-H. Lin, W.-H. Hung, and Y.-F. Cheng, "The comparison between adenocarcinoma and squamous cell carcinoma in lung cancer patients," Journal of cancer research and clinical oncology, vol. 146, no. 1, pp. 43–52,
2020.

[9] S. Li, P. Xu, B. Li, L. Chen, Z. Zhou, H. Hao, Y. Duan, M. Folkert, J. Ma, S. Huang, et al., "Predicting lung nodule malignancies by combining deep convolutional neural network and handcrafted features," Physics in Medicine & Biology, vol. 64, no. 17, p. 175012, 2019.

[10] images from biorender, "www.biorender.com," 2022.