{geometry
a4paper, top=1.0in, left=1.0in, bottom=1.0in, right=1.5in,

A REPORT ON

"Lung Cancer Detection Using CNN"

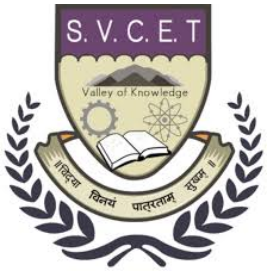
SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS

OF

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

SUBMITTED BY

Kalwaghe Akshay	**Exam No: 72009898L**
Nutan Varpe	**Exam No: 72015819C**
Pratibha Varpe	**Exam No: 72015820G**

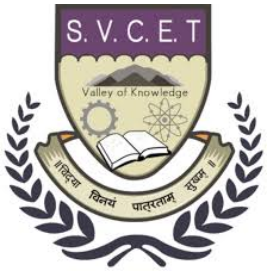


DEPARTMENT OF COMPUTER ENGINEERING

SAHYADRI VALLEY COLLEGE OF ENGINEERING AND TECHNOLOGY

RAJURI, PUNE 412411.

SAVITRIBAI PHULE PUNE UNIVERSITY



CERTIFICATE

This is to certify that the project report entitles

"Lung Cancer Detection Using CNN"

Submitted by

Kalwaghe Akshay	**Exam No: 72009898L**
Nutan Varpe	**Exam No: 72015819C**
Pratibha Varpe	**Exam No: 72015820G**

is a bonafide student of this institute and the work has been carried out by him/her under the supervision of **Prof. Pinjarkar N.R.** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of Bachelor of Engineering (Computer Engineering).

Prof. Pinjarkar N.R.
Guided By
Dept. of Computer Engg.

Prof. Dighe S.
HOD
Dept. of Computer Engg.

Prof. Pinjarkar N.R.
Project Co-ordinator
Dept. of Computer Engg.

Prof. P.Balaramudu
Vice-Principal
SVCET

Place :
Date:

ACKNOWLEDGEMENT

It gives us great pleasure in presenting the preliminary seminar report on ***'Lung Cancer Detection Using CNN****.*

I would like to take this opportunity to thank my internal guide for project ***Prof.Pinjarkar N.R.*** *for giving me all the help and guidance I needed. I am really grateful to them for their kind support. Their valuable suggestions were very helpful.*

I am also grateful to ***Prof.Dighe S.****, Head of Department and* ***Prof.P.Balaramudu****, Vice principal, SVCET for his indispensable support, suggestions.*

In the end our special thanks to All Staff for providing various resources such as laboratory with all needed software platforms, continuous Internet connection, for Our Project.

Kalwaghe Akshay
Nutan Varpe
Pratibha Varpe

ABSTRACT

Lung cancer is the cause of every sixth death around the world making it the second leading cause of death. Approximately 42 million people across the world suffer from cancer and this figure is continuously increasing. In India, approximately two and half million people are suffering from different types of cancer. If most of the cancers are detected in an early stage, then with the right remedy they can be cured. This paper offers details on a method in which CNN algorithm is used for the detection of illnesses like lung cancer. This paper also details the different machine learning techniques used to classify cancer into malignant and normal category. We get good accuracy on CNN which is 96.23% on 100 epochs.

Index Terms :Convolutional Neural Network, Deep Learning, Image processing

Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 Introduction of respective area

Despite numerous developments in the field of diagnosis of illnesses like cancer, still, the tumor is one of the risky and dangerous illness. Lung Cancer is the second most popular cause for death not only in India but across the world. Diagnosis of a tumor is a totally critical and important task. The detection and remedy of cancerous tumors are one of the most important research and study areas. If the cancer is identified at an early stage and if right remedy is given quickly after the detection of the disorder, the rate of survival for the patients can be improved. There are numerous strategies or techniques used to seize various types of cancers, like PET scan, CT scan, Mammograms, MRI, 3D Ultrasound, Single Photon Emission Computed Tomography (SPECT), etc. Mammograms are used for breast cancer detection analysis. CT scan, MRI and several other techniques are used to identify brain tumors, lung cancer, etc. The imaging method taken into consideration is mammogram and the type of classification strategies used are Feed forward back propagation, Extreme Learning Machine (ELM) ANN, backpropagation ANN, Particle Swarm Optimized Wavelet Neural Network, and CNN based on deep learning. For brain tumors, the imaging technique used is MRI and CT scan and the classification techniques considered are Level Set, K means Algorithm, SVM, Fuzzy C-means, Ad boost, Naïve Bayes classifier, and ANN classifier. For lung cancer, the medical imaging technique used is PET/CT. Also, classification techniques considered are FCM classifier, Feed Forward ANN, ANN, SVM binary classifier, and Entropy degradation method. Medical imaging techniques such as MRI and classification methods like ANN, SVM, and Multilayer perceptron neural network are considered for spine tumor detection. The two kinds of cancers are harmful and harmless growths. Standard MRI successions are for the most part used to separate various sorts of cerebrum cancers dependent on visual characteristics and different surface investigations of the delicate tissue. More than 120 classes of cerebrum cancers are known to be grouped in four levels as per the level of harm by the World Health Organization (WHO). A wide range of cerebrum cancers brings out certain indications dependent on the impacted district of the mind. The significant manifestations might incorporate migraines, seizures, vision issues, spewing, mental changes, memory slips, balance loss and so forth. Causes of cerebrum cancers are hereditary qualities, ionizing radiation cell phones, very low recurrence attractive fields, synthetic compounds, and head injury. Also, injury-resistant elements like infections and hypersensitivities may cause mild to severe cancer. The dangerous growths, otherwise called destructive

cancers, are of two sorts - essential growths, which start from the cerebrum, and optional growths, which begin someplace and spread to the mind. The danger factors for mind growth are openness to vinyl chloride, neurofibromatosis, ionizing radiations, etc.

1.2 Motivation

- Reducing death rate by wrong diagnosis using giving accurate diagnosis.
- To provide diagnosis system this helps to doctors.
- Motive behind proposed work is to achieve higher accuracy over existing work by using machine learning.
- The desire to provide a better and accurate diagnosis.

1.3 Problem Defination

• To reduce death rate by skin diseases in the world
To give unique solution for multiple diseases
• To provide higher accuracy over previous research
• To give most promising tool that can acceptable by all the doctors.
• Detection of multiple types of diseases Pneumonia, lung cancer and tumor.

CHAPTER 2

LITERATURE SURVEY

Wadood Abdul [1] used the architecture of CNN, a deep learning solution, in classifying the lung nodules as benign or malignant. LIDC-IDRI database was tested and the best results were obtained with 97.2% accuracy, 95.6% sensitivity, and 96.1% specificity, which outperforms the results obtained with other learning techniques. So, the ALCDC system performs better than the existing state-of-the-art systems.

Chao Ma, Gongning Luo, and Kuanquan Wang Waghmode et al. [2] stated that in this work, we introduce a new methodology that combines random forests and an active contour model for the automated segmentation of the glioma a type of tumor that occurs in the brain and spinal cord from multimodal volumetric MR images. Specifically, we employ a feature representation learning strategy to effectively explore both local and contextual information from multi-modal images for tissue segmentation by using modality-specific random forests as the feature learning kernels.

Onur Ozdemir et al. [3] proposed that the entirely 3D convolutional neural networks achieve state-of-the-art performance for both lung nodule detection and malignancy classification tasks on the publicly available LUNA16 and Kaggle Data Science Bowl challenges. It is important to have the coupling between detection and diagnosis components as nodule detection systems are typically designed and optimized on their own.

Anum Masood, Bin Sheng, Po Yang, and Ping Li, [4] proposed experimented enhanced multidimensional Region-based Fully Convolutional Network (mRFCN) based automated decision support system for lung nodule detection and classification. The mRFCN is used as an image classifier tool for feature extraction along with the novel multi-Layer fusion Region Proposal Network (mLRPN) with position-sensitive score maps (PSSM) being explored. They applied a median intensity projection to leverage three-dimensional information from CT scans and introduced a deconvolutional layer to adopt the proposed mLRPN in the architecture to automatically select potential regions of interest.

Khan Muhammad, Salman Khan [5] stated that an in-depth review of the surveys published so far and recent deep learning-based methods for BTC. Our survey covers the main steps of deep learning-based BTC methods, including preprocessing, features extraction, and classification, along with their achievements and limitations.

Chun-Mei Feng, Yong Xu [6] et al. Stated that discriminative information and sparsity in the PCA model. Specifically, in contrast to the traditional sparse PCA, which imposes sparsity on the loadings, here, sparse components are obtained to represent the data.

David N. Louis et al. [7] stated that notable changes include the addition of brain invasion as a criterion for atypical meningioma and the introduction of a soft tissue-type grading system for the new combined entity of solitary fibrous tumor human giopericytoma-a departure from how other CNS tumors are graded. Overall, this will facilitate clinical, experimental, and epidemiological studies that lead to improvements in the lives of patients with brain tumors.

Pär Salander et al. [8] proposed that most spouses witnessed months of global dysfunction preceding the symptom leading to physician consultation. The patient factors 'less alien symptoms', 'personality change' and 'avoidance'; the spouse factors 'spouse's passivity and 'spouse's successive adaptation'; and the physician factors 'reasonable alternative diagnosis', 'physician's inflexibility and 'physician's personal values' were identified as obstacles on the pathway to appropriate medical care.

Et al. [9] stated that the term brain tumor refers to a mixed group of neoplasms originating from intracranial tissues and the meninges with degrees of malignancy ranging from benign to aggressive. Each type of tumor has its biology, treatment, and prognosis and each is likely to be caused by different risk factors. Even benign tumors can be lethal due to their site in the brain, their ability to infiltrate locally, and their propensity to transform into malignancy. This creates problems in describing the epidemiology of these conditions and makes the classification of brain tumors a difficult science.

Jan J Heimans et al. [10] Proposed that a large number of Quality-of-Life instruments have been developed. The European Organization for Research and Treatment of Cancer Quality of Life Questionnaire (EORTC QLQ-C30) and the MOS Short-Form Health Survey are two frequently used general HRQL instruments. A specific brain tumor scale is the Brain Cancer Module, which is designed to be used in combination with general questionnaires. HRQL measurement and neuropsychological examination were used to investigate the impact of radiotherapy and surgery in low-grade glioma patients and the influence of tumor volume, tumor localization, performance status and age in both low-grade and high-grade glioma patients.

Malavika Suresh [11] stated that a noncognitive computer user interface has the endowment to perceive gestures and execute commands based on that. The design is implemented on a Linux system but can be implemented by installing modules for python on a windows system also. OpenCV and KERAS are the platforms used for identification. Gesture displayed on the screen is recognized by the vision-based algorithms. Using background removal technique, an assortment of skin color masks was trained by Lenet architecture in KERAS for the recognition.

M. Gurbina, M. Lascu, D. Lascu [12] stated that differentiate between a normal brain and a tumor brain (benign or malign). The study of some types of brain tumors such as metastatic bronchogenic carcinoma tumors, glioblastoma and sarcoma are performed using brain magnetic resonance imaging (MRI). The detection and classification of MRI brain tumors are implemented using different wavelet transforms and support vector machines. Accurate and automated classification of MRI brain images is extremely important for Medi-Cal analysis and interpretation.

S. Somasundaram, R. Gobinath [13] stated that focus on six features that are entropy, mean, correlation, contrast, energy and homogeneity. The performance metrics accuracy, sensitivity, and specificity are calculated to show that the proposed method is better compared to existing methods. The proposed technique uses MATLAB to detect the location and the size of a tumor in the brain through an MRI image.

Dhanasekaran Raghavan [14] proposed that the target with the aid of the following major steps, which include: Pre-processing of the brain images segmentation of pathological tissues Fluid (CSF)), extraction of the relevant features from each segmented tissue and classification of the tumor images with NN. As well, the experimental results and analysis are evaluated using Quality Rate (QR) with normal and abnormal Magnetic Resonance Imaging (MRI) images.

G. Hemanth; M. Janardhan [15] stated highly efficient and precise methods for brain tumor detection, classification and segmentation. To achieve this precise automatic or semi-automatic methods are needed. The research proposes an automatic segmentation method that relies upon CNN (Convolution Neural Networks), determining small 3×3 kernels. By incorporating this single technique, segmentation and classification are accomplished. CNN (an ML technique) from NN (Neural Networks) wherein it has layer-based for results classification. Various levels involved in the proposed mechanisms are data collection, pre-processing, average filtering, segmentation, feature extraction, CNN via classification and identification. With the use of Data Mining (DM) techniques, significant relations and patterns from the data can be extracted. The techniques of Machine Learning (ML) and DM are being effectively employed for brain tumor detection and prevention at an early stage of cancer.

S.K. Lakshmanaprabu [16] stated that Optimal Feature Level Fusion (OFLF) is considered to fuse low and high-level features of brain images; from this analysis, the images are classified as Benign or Malignant. From this implementation of medical images, the experiment results are evaluating performance metrics that are compared to existing classifiers. From the proposed MRI image classifica-

tion process the accuracy was 96.23%, sensitivity was 92.3% whereas specificity was 94.52%; compared to the existing classifier. This proposed methodology is implemented in the working platform of MATLAB.

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATION

3.1 Introduction

Any disease whether it is curable or not, it must be diagnosed properly with some time in hand to take the appropriate actions in time. As it is popularly said that early detection of any disease is half cured. Chest radiology is the most common method used for diagnosis of lung diseases, the term lung disease refers to the abnormalities that effect the lung organ, diseases are such as asthma, COPD, lung cancer, pneumonia and many other breathing problems. Medical data growth in healthcare communities, accurate analysis of medical data benefit early disease detection, patient care and community services. However, the analysis of patients is depends on accuracy of diagnosis and then treatment as well. The wrong diagnosed patients lead to deaths in diseases. So the high risk of diagnosis there is need of accurate diagnosis aid for Pneumonia, lung cancer and brain tumor diseases. So we are proposing diagnosis system based on machine learning for giving promising solution with high accuracy. The proposed system consists of many diseases such as Pneumonia, lung cancer and brain tumor disease detections and stages predictions. To reduce all the time being wasted, with the help of machine learning we intend to learn if we can predict if the patient is infected with a certain disease or not. We do that is by using deep learning model.

3.2 Assumptions and Dependencies

- To have understanding of the problem statement.
- To know what are the hardware and software requirements of proposed system.
- To have understanding of proposed system.
- To do planning various activities with the help of planner.
- Designing, programming, testing etc.

3.3 Functional Requirements

3.3.1 Login

The customer should be able to login in our system using valid email id and password which he/she has created at time of registration. If user enters email id or password that does not match in our database, he/she will see a message " Invalid User id or Password "

3.3.2 Secure Private Data

The user should be able to store their data securely.

3.3.3 Logout

The user must be logged into our System. This function is used when a logged in user finishes his/her job and wants to be logged out so that no one can misuse his/her username. The system will state the user has been logged out successfully

3.4 Nonfunctional Requirements

3.4.1 Performance Requirements

Error message should be displayed. .If there is no any location search then location not found message displayed.

3.4.2 Safety Requirements

If using ID Password mistake cannot be happen

3.4.3 Security Requirements

Refers to the technical innovations and procedures applied to the hardware and operation systems to protect against deliberate or accidental damage from a defined threat.

- Secure access of confidential data (users details). SSL can be used.

- 24 X 7 availability.
- Better component design to get better performance at peak time.
- Flexible service based architecture will be highly desirable for future extension.

3.4.4 Software Quality Attributes

Our software has many quality attribute that are given below:-

- Adaptability: This software is adaptable by all users.
- Availability: This software is freely available to all users. The availability of the software is easy for everyone.
- Maintainability: After the deployment of the project if any error occurs then it can be easily maintained by the software developer.
- Reliability: The performance of the software is better which will increase the reliability of the Software.
- User Friendly: Since, the software is a GUI application; the output generated is much user friendly in its behavior.
- Integrity: Integrity refers to the extent to which access to software or data by unauthorized persons can be controlled.
- Security: Users are authenticated using many security phases so reliable security is provided.
- Testability: The software will be tested considering all the aspects.

3.4.5 User Interfaces

Describe the logical characteristics of each user interface that the system needs. Some possible items to include are

- References to GUI standards style guides that are to be followed.
- Standards for fonts, icons, button labels, images, color schemes, field tabbing sequences, commonly used controls, and the like.

- Screen layout or resolution constraints.
- Standard buttons, functions, or navigation links that will appear on every screen, such as a help button.
- Shortcut keys.
- Message display conventions to be followed.
- Layout standards to facilitate software localization.

3.4.6 Communication Interfaces

State the requirements for any communication functions the product will use, including e-mail, Web browser, network communications protocols, and electronic forms.
1.Web Browser : Mozilla or Chrome
2.Protocol : Internet protocols used widely in the World Wide Web (Web)

3.5 System Requirements

3.5.1 Database Requirements

- Server: Google
- Database: Sqlite3

3.5.2 Software Requirement

- Operating system: Windows 7/8/10.
- Application Server: Apache Tomcat 7/8/9.
- Front End: HTML
- Scripts: Python
- Server side Script: Python
- IDE: Pycharm

3.5.3 Hardware Requirements

- Processor: Intel i3/i5/i7.
- Speed: 1.1 GHz..
- RAM: 2 GB(min).
- Hard Disk: 40 GB.
- Floppy Drive: 1.44 MB.
- Key Board: Standard Windows Keyboard.
- Mouse: Two or Three Button Mouse.
- Monitor: SVGA

3.6 Analysis Models: SDLC Model to applied

A process model for software engineering is selected based on the nature of the project and application, the methods and tools to be used and the controls and deliverable that are required. Incremental Model Incremental model is used as the process model in this project. The incremental model is a method of software development where the product is designed, implemented and tested incrementally (a little more is added each time) until the product is finished. It involves both development and maintenance.

1. Requirement analysis: In the first phase of the incremental model, the product analysis expertise identifies the requirements. And the system functional requirements are understood by the requirement analysis team. To develop the software under the incremental model, this phase performs a crucial role.
2. Design Development: In this phase of the Incremental model of SDLC, the design of the system functionality and the development method are finished with success. When software develops new practicality, the incremental model uses style and development phase.
3. Testing: In the incremental model, the testing phase checks the performance of each existing function as well as additional functionality. In the testing phase,

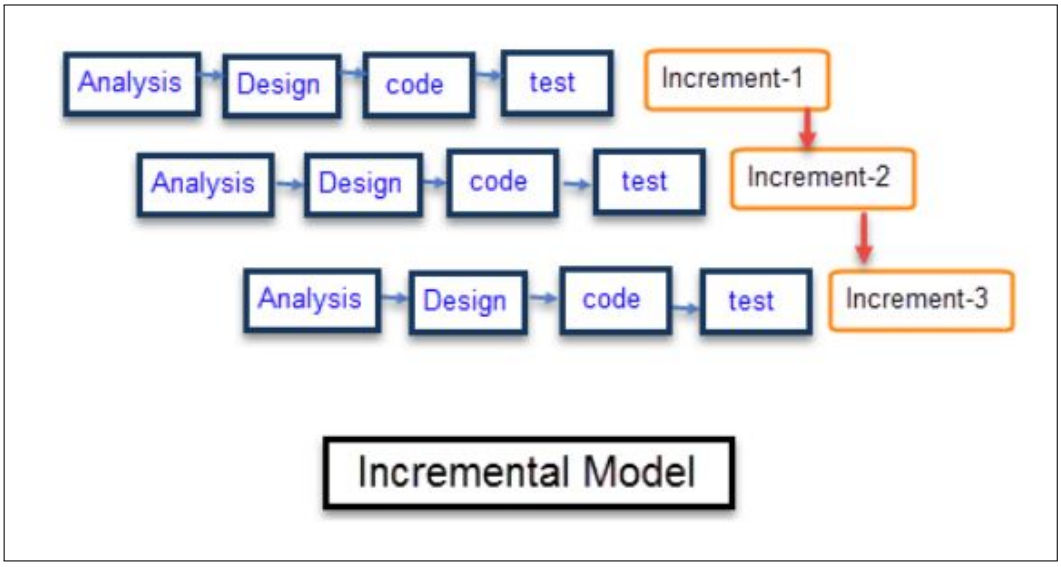


Figure 3.1: Incremental Model

the various methods are used to test the behavior of each task.

4. Implementation: Implementation phase enables the coding phase of the development system. It involves the final coding that design in the designing and development phase and tests the functionality in the testing phase. After completion of this phase, the number of the product working is enhanced and upgraded up to the final system product.

Advantage of Incremental Model:

- Errors are easy to be recognized.
- Easier to test and debug.
- More flexible.
- Simple to manage risk because it handled during its iteration.
- The Client gets important functionality early.

3.7 Mathematical Model

Set S={I,R,P,O}
Where,
I=Set of Input For Our System
R=Set Of Rules that are applied while processes are performed.
P=Set Of Processes
O=Set Of Outputs
I=I1,I2,I3,I4
I1=Add User Information
I2=Provide File Information
I3=Provide Key
R=R1,R2
R1=Get Proper Display
R2=Find Out Proper Information
P=P1,P2,P3
P1=Validation Of Required Details
P2=Upload Data Properly
P3=Distribute Data Properly
O=O1,O2
O1=Data Uploaded
O2=Data Accessing Properly

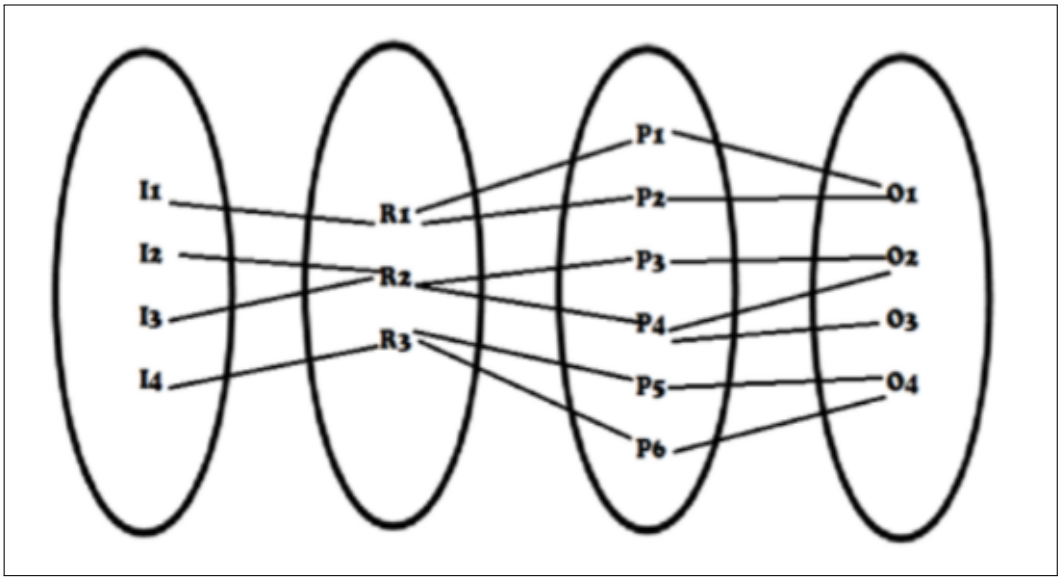


Figure 3.2: Venn Diagram

CHAPTER 4

SYSTEM DESIGN

4.1 System Architecture

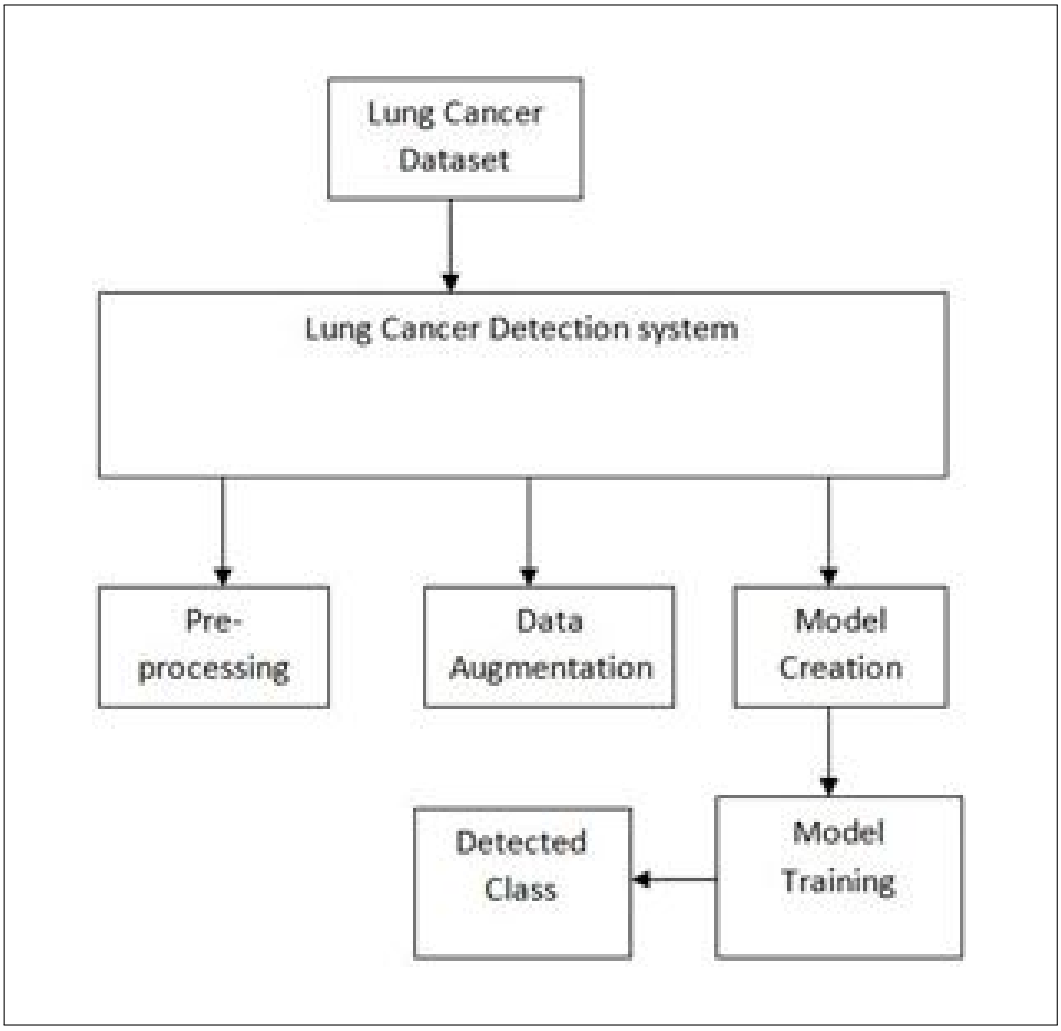


Figure 4.1: System Architecture

In a proposed system, we are proposing an experiment on lung cancer disease with a limited set of supervised data. We are proposing a combination of a Convolutional neural network-based multimodal disease risk prediction model

with higher accuracy. We are going to solve the accuracy issue in the diagnosis of lung cancer with accurate stage predictions.
B. Dataset We have collected the dataset from kaggle platform. We have split the dataset into two categories training and testing. For training 300 images and testing 60 mages we are used. Below is the following link
C. Pre-processing In pre-processing we are convert the every image into 224*224.
D. Data augmentation In data augmentation we are simply increase the dataset of training directory. We generate every image different format such as rotation, zoom and change the brightness of image.

4.2 UML Diagram

4.2.1 Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. In this Number of actors are work like Auctioneer, Auction Agent and Bidder. In this Ellipses are shows the use cases like Registration,Login,Security,Results etc.

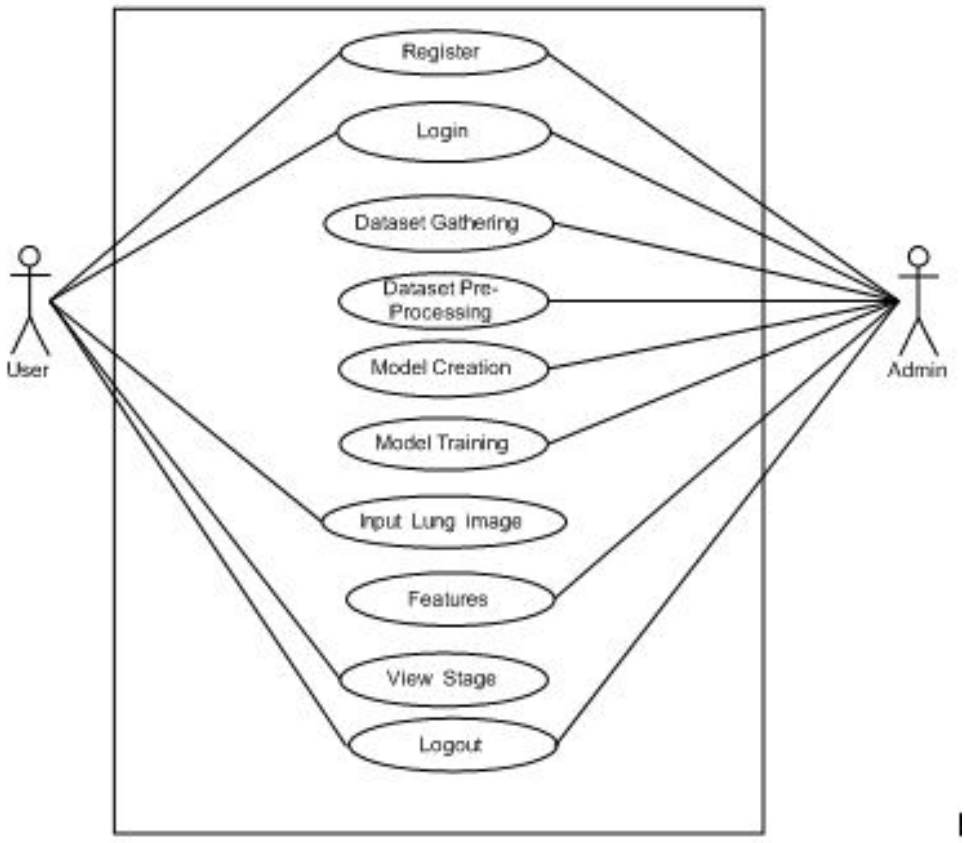


Figure 4.2: Usecase 1

4.2.2 Class Diagram

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.The class diagram is the main building block of object-oriented modelling. In the diagram, classes are represented with boxes that contain three compartments: The top compartment contains the name of the class. It is printed in bold

and centered, and the first letter is capitalized. The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase. The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

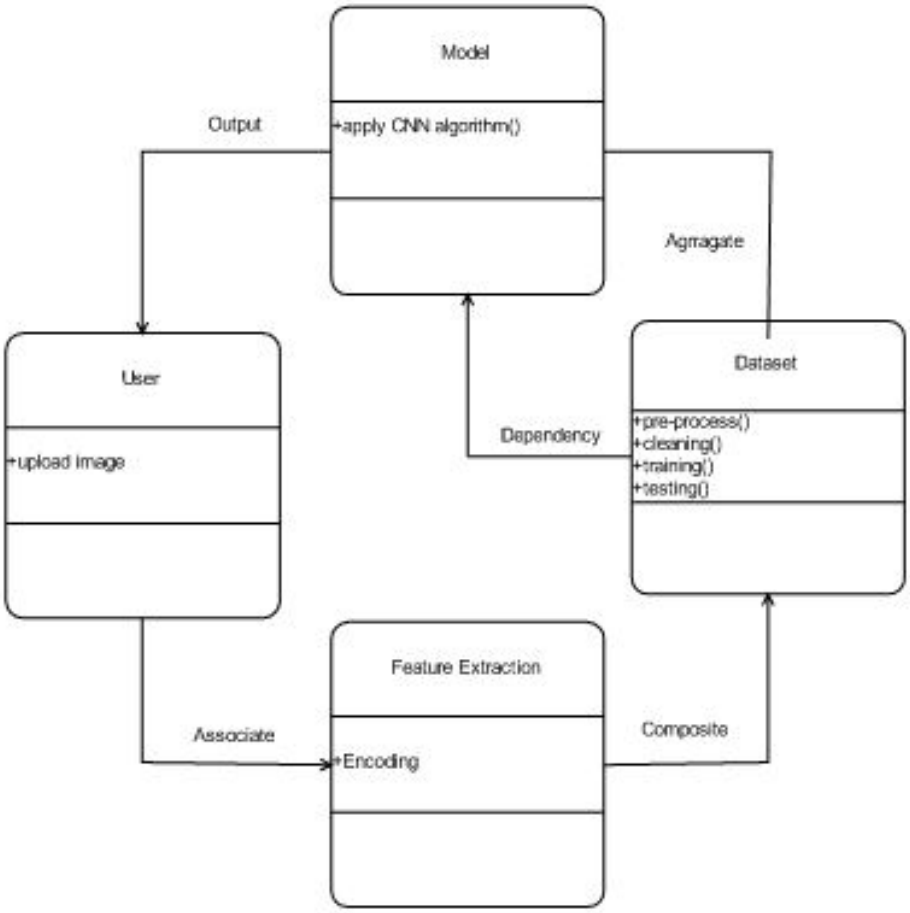


Figure 4.3: Class Diagram

CHAPTER 5

OTHER SPECIFICATION

5.1 Algorithms

CNN Algorithm

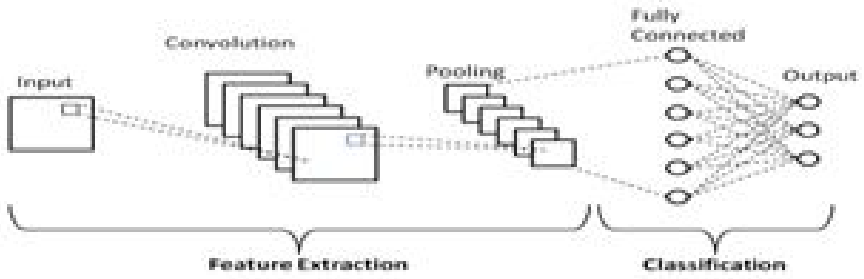


Figure 5.1: CNN Algorithm

5.2 Advantages

- Simple and easy to understand and use.
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood. • Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

5.3 Applications

- Security of Data.
  Image Classification, Medical sector

CHAPTER 6

PROJECT PLAN

Here the prediction is made about the size of total project. Effective software projectestimation is one of the most challenging and important activity in software development once you have on estimate size of your product you can desire the effort estimate. Software project estimation are having 4 basic steps:

- Estimate size of development product this generally ends up in either lines of code (loc) or function point (FP).
- Estimate effort in person-months or person hours.
- Estimate the schedule in calendar months.
- Estimate project cost.

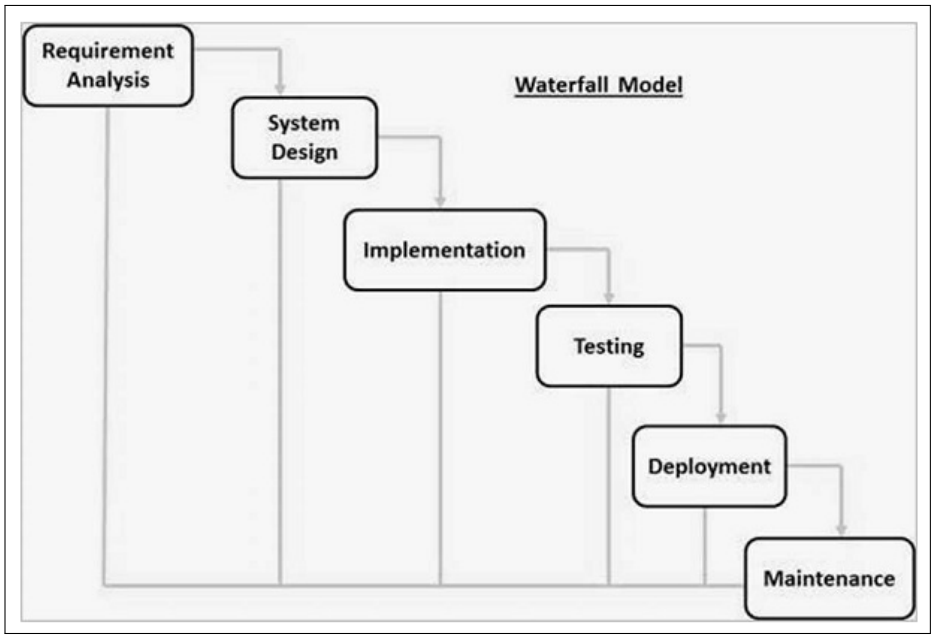


Figure 6.1: Waterfall Model

Waterfall Model - Application
Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

Waterfall Model - Advantages
The advantages of waterfall development are that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.
Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.
Some of the major advantages of the Waterfall Model are as follows

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

6.0.1 Reconciled Estimates

6.0.1.1 Cost Estimate

The Constructive Cost Model (COCOMO) is generally used estimation measures of cost, project duration, man power, etc.
Like all estimation model, the COCOMO model requires sizing information. This information can be specified in the form of
1. Object Point (OP)
2. Function Point (FP)
3. Lines of Source Code (KLOC)
For our project, we use the sizing information in the form of Lines of source code.
Equation for calculation of effort in person-month for the COCOMO model is:
E = a * (KLOC) bWhere;
a=3.2
b=1.05, for semi-detached projects
E=Effort in person-months
D = E/N
Where,
E=Effort in person-months
N=Number of persons required
D=Duration of project in months.
Estimation of KLOC:
KLOC according to module
Total number of code required to estimate to be 4.2 KLOC.

6.0.1.2 Time Estimates

Efforts: In which we are calculated efforts done by the each person in month. Efforts are calculated by using formula

E=3.2(KLOC)1:05E = 3:2(4:2)1:05
E=4*30 Person-month
Development time:
D=E/N
D=4*30/4

Sr. No.	Module Estimated	KLOC
1	Graphical User Interface	0.7
2	Get Tripadvisor Information	1.4
3	Add POI And Remove POI	0.3
4	Display Map	0.5
5	Notification	1.3
6	Total KLOC	4.2

Table 6.1: KLOC

D=3.82 month
Development time for Project
Requirements analysis require 3 months
Implementation and testing requires 3.82 months.
Total Duration for completion of project D= 6.82 months.

6.0.2 Project Resources

Hardware Resources Required

Sr.No.	Parameter	Minimum Requirement
1	Processor	Pentium IV /Intel I3 core
2	Speed	1.1 GHZ
3	Ram	512 MB
4	Hard Disk	20 GB
5	Keyboard	Standard Keyword
6	Mouse	Two Or Three Button Mouse
7	Monitor	LED Monitor

Software Resources Required
Platform :
1. Operating System : Windows XP / 7
2. Programming Language : Java/J2EE
3. Software Version : JDK 1.7 or above
4. Tools : Eclipse

5. Front End : JSP
6. Data Base : Mysql

6.1 Risk Management w.r.t. NP Hard analysis

In computational complexity theory, the complexity class NP-complete (abbreviated NP-C or NPC) is a class of decision problems. A decision problem L is NP-complete if it is in the set of NP problems and also in the set of NP-hard problems. The abbreviation NP refers to non deterministic polynomial time. NP-hard (Non-deterministic Polynomial-time hard), in computational complexity theory, is a class of problems that are, informally, at least as hard as the hardest problems in NP. A problem H is NP-hard if and only if there is an NP-complete problem L that is polynomial time Turing-reducible to H (i.e., LTH). The problem statement says that initially our problem was NP-hard but as we proceeded the desired output was gained and it was converted into NP-complete.

NP-Hard NP-Complete

NP-hard NP-Complet. Example is given below

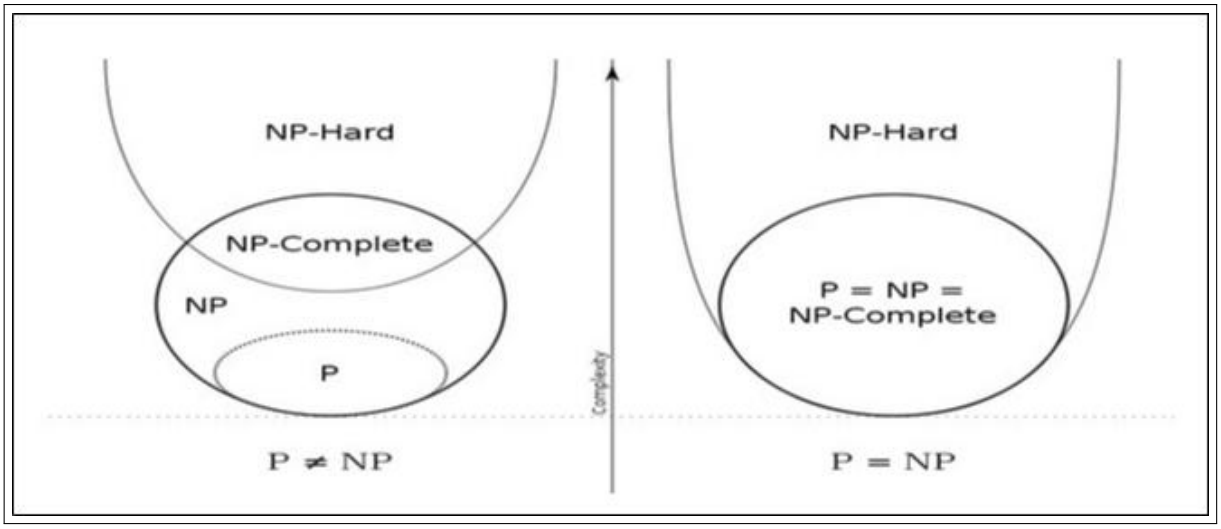


Figure 6.2: NP-hard and NP-complete

This problem is NP-Hard and we can make it NP-Compete by using this system.

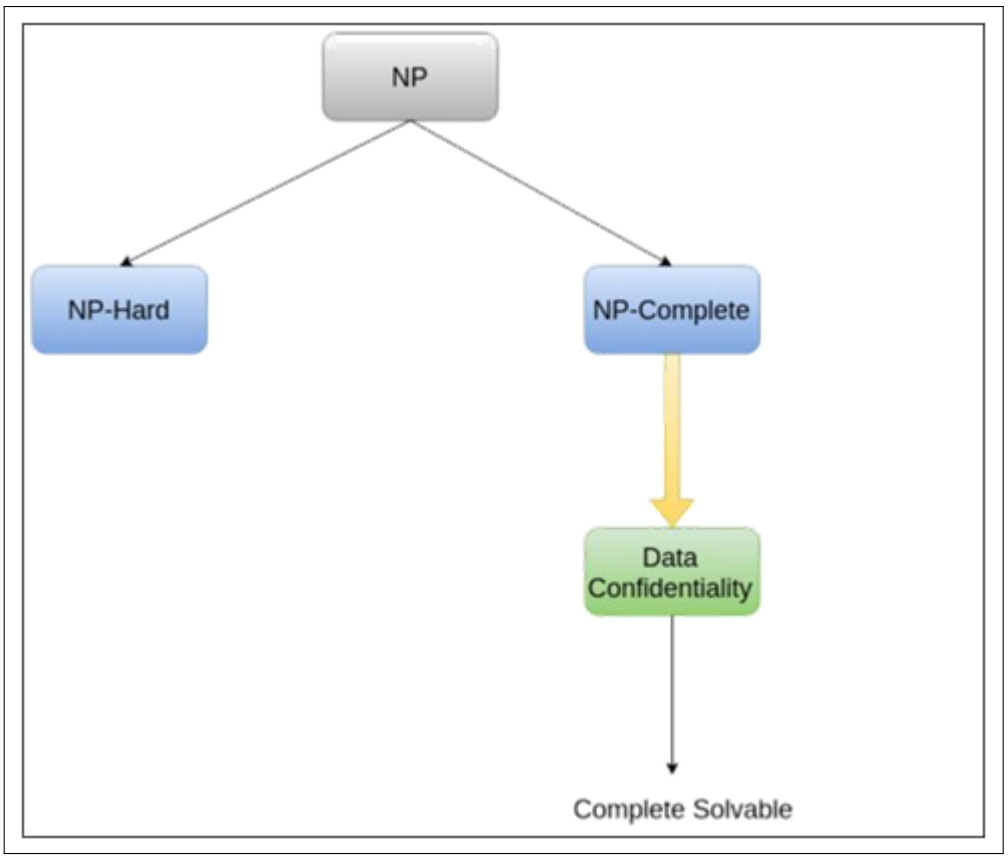


Figure 6.3: NP-hard and NP-complete

6.1.1 Risk Identification

Risk analysis and management are a series of steps that help a software team to understand and manage uncertainty. Risks affect the project very much and they can be a serious threat. Risk can even destroy the the whole setup but as the project is to be made we have to take risks.

For risks identification, review of scope document, requirements specifications and schedule is done. Answers to questionnaire revealed some risks. Each

risk is categorized as per the categories mentioned in. Please refer table 2.1 for all the risks. You can refer following risk identification questionnaire.

1 Have top software and customer managers formally committed to support the project?

2 Are end-users enthusiastically committed to the project and the system/product to be built?

3 Are requirements fully understood by the software engineering team and its customers?

4 Have customers been involved fully in the definition of requirements?

5 Do end-users have realistic expectations?

6 Does the software engineering team have the right mix of skills?

7 Are project requirements stable?

8 Is the number of people on the project team adequate to do the job?

9 Do all customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built?

6.1.2 Risk Analysis

The risks for the Project can be analyzed within the constraints of time and quality

6.1.3 Overview of Risk Mitigation, Monitoring, Management

Following are the details for each risk.

ID	Risk Description	Probability	Impact		
			Schedule	Quality	Overall
1	Will the system process 100-150 systems	Low	Low	High	High
2	Will the system able to handle multiple users at a time?	Low	Low	High	High
3	Will the system handle multiple errors?	Low	Low	High	High
4	Will the system degrade the performance of other subsystems	Low	Low	High	High
5	Will the software have defects after release?	Low	Low	High	High

Table 6.2: Risk Analysis

Probability	Value	Critical
High	Probability of occurrence is	> 75%
Medium	Probability of occurrence is	26 75%
Low	Probability of occurrence is	< 25%

Table 6.3: Risk Probability definitions

Impact	Value	Description
Very high	> 10%	Schedule impact or Unacceptable quality
High	5 10%	Schedule impact or Some parts of the project have low Quality
Medium	< 5%	Schedule impact or Barely noticeable degradation in quality LowImpact on schedule or Quality can be incorporated

Table 6.4: Risk Impact definitions

Risk ID	1
Risk Description	Unrealistic Schedule
Category	Time
Source	Failure to identify complex functionalities and time required to develop those functionalities
Probability	Mid
Impact	High
Response	Mitigate
Strategy	Strategy
Risk Status	Not Occurred

Table 6.5: Risk 1

Risk ID	2
Risk Description	Literature Servey and Communication
Category	Requirements
Source	Software Design Specification documentation review.
Probability	Mid
Impact	High
Response	Mitigate
Strategy	Better testing will resolve this issue.
Risk Status	Identified

Table 6.6: Risk 2

Risk ID	3
Risk Description	Use of wrong Technology
Category	Technology
Source	This was identified during early development and testing.
Probability	Low
Impact	Very High
Response	Accept
Strategy	Example Running Service Registry behind proxy balancer
Risk Status	Identified

Table 6.7: Risk 3

Risk ID	4
Risk Description	Unrealistic budget
Category	Cost
Source	Project scope expansion
Probability	Low
Impact	Very High
Response	Accept
Strategy	Detailed cost estimation
Risk Status	Not Occurred

Table 6.8: Risk 4

Risk ID	5
Risk Description	Difficult to implement
Category	Technology
Source	This was identified during early development and testing.
Probability	Mid
Impact	Very High
Response	Accept
Strategy	Better programming and testing will resolve this issue
Risk Status	Identified

Table 6.9: Risk 5

6.2 Project Schedule

6.2.1 Project task set

Major Tasks in the Project stages are:

- Task 1: Requirement Analysis (Base Paper Explanation).
- Task 2: Project Specification (Paper Work).
- Task 3: Technology Study and Design.
- Task 4: Coding and Implementation (Module Development).

6.2.2 Timeline Chart

A project timeline chart is presented. This may include a time line for the entire project. Above points should be covered in Project Planner as Annex C and you can mention here Please refer Annex C for the planner

Schedule		Date	Project Activity
October	1st Week	09/10/2020	Project Group Formation
	2nd Week	16/10/2020	Project Topic Searching
	4th Week	30/10/2020	Synopsis Submission
November	2nd Week	13/11/2020	Project Topic Selection
	4th Week	27/11/2020	Presentation of Project Idea
December	2nd Week	11/12/2020	Submission Of Literature Survey
	3rd Week	18/12/2020	Feasibility Assessment
January	1st Week	01/01/2021	Mid Sem Presentation
	3rd Week	15/01/2021	Design Of Mathematical Model
February	2nd Week	12/02/2021	End Sem Presentation
March	1st Week	05/03/2021	Preparation for Conference
	4th Week	26/03/2021	Paper Publish
April	2nd Week	09/04/2021	Report Preparation and submission
	3rd Week	23/04/2021	1 st module Presentation
May	2nd Week	07/05/2021	Discussion and implementation of 2nd module
	4th Week	28/05/2021	Work on User Interface
June	1st Week	04/06/2021	Integration of all Module
	4th Week	25/06/2021	Final report and presentation

Table 6.10: Timeline Chart

6.3 Team Organization

The manner in which staff is organized and the mechanisms for reporting are noted.

6.3.1 Team structure

The team structure for the project is identified. Roles are defined.

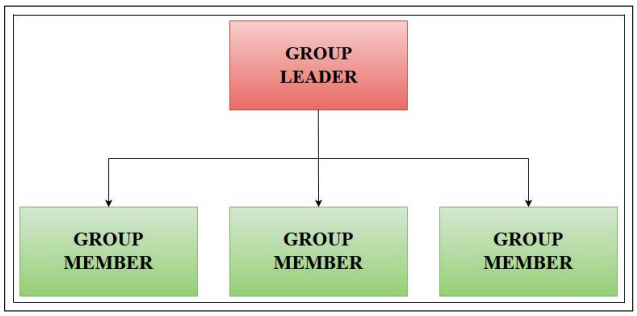


Table 6.11: Team Organization

CHAPTER 7

SOFTWARE IMPLEMENTATION

7.1 Introduction

7.1.1 Structured Programming

In the process of coding, the lines of code keep multiplying, thus, size of the software increases. Gradually, it becomes next to impossible to remember the flow of program. If one forgets how software and its underlying programs, files, procedures are constructed it then becomes very difficult to share, debug and modify the program. The solution to this is structured programming. It encourages the developer to use subroutines and loops instead of using simple jumps in the code, thereby bringing clarity in the code and improving its efficiency Structured programming also helps programmer to reduce coding time and organize code properly.
Structured programming states how the program shall be coded. Structured programming uses three main concepts: Top-down analysis -
A software is always made to perform some rational work. This rational work is known as problem in the software parlance. Thus it is very important that we understand how to solve the problem. Under top-down analysis, the problem is broken down into small pieces where each one has some significance. Each problem is individually solved and steps are clearly stated about how to solve the problem.
Modular Programming -
While programming, the code is broken down into smaller group of instructions. These groups are known as modules, sub-programs or subroutines. Modular programming based on the understanding of top-down analysis. It discourages jumps using 'goto' statements in the program, which often makes the program flow non-traceable. Jumps are prohibited and modular format is encouraged in structured programming.
Structured Coding -
In reference with top-down analysis, structured coding sub-divides the modules into further smaller units of code in the order of their execution. Structured programming uses control structure, which controls the flow of the program, whereas structured coding uses control structure to organize its instructions in definable patterns.

7.1.2 Functional Programming

Functional programming is style of programming language, which uses the con-

cepts of mathematical functions. A function in mathematics should always produce the same result on receiving the same argument. In procedural languages, the flow of the program runs through procedures, i.e. the control of program is transferred to the called procedure. While control flow is transferring from one procedure to another, the program changes its state.
In procedural programming, it is possible for a procedure to produce different results when it is called with the same argument, as the program itself can be in different state while calling it. This is a property as well as a drawback of procedural programming, in which the sequence or timing of the procedure execution becomes important.
Functional programming provides means of computation as mathematical functions, which produces results irrespective of program state. This makes it possible to predict the behavior of the program. Functional programming uses the following concepts:
First class and High-order functions -
These functions have capability to accept another function as argument or they return other functions as results.
Pure functions -
These functions do not include destructive updates, that is, they do not affect any I/O or memory and if they are not in use, they can easily be removed without hampering the rest of the program.
Recursion -
Recursion is a programming technique where a function calls itself and repeats the program code in it unless some pre-defined condition matches. Recursion is the way of creating loops in functional programming.
Strict evaluation -:
It is a method of evaluating the expression passed to a function as an argument. Functional programming has two types of evaluation methods, strict (eager) or non-strict (lazy). Strict evaluation always evaluates the expression before invoking the function. Non-strict evaluation does not evaluate the expression unless it is needed. -calculus -
Most functional programming languages use $-$calculus as their type systems. $-$expressions are executed by evaluating them as they occur.

7.1.3 Programming style

Programming style is set of coding rules followed by all the programmers to write the code. When multiple programmers work on the same software project,

they frequently need to work with the program code written by some other developer. This becomes tedious or at times impossible, if all developers do not follow some standard programming style to code the program.
An appropriate programming style includes using function and variable names relevant to the intended task, using well-placed indentation, commenting code for the convenience of reader and overall presentation of code. This makes the program code readable and understandable by all, which in turn makes debugging and error solving easier. Also, proper coding style helps ease the documentation and updation.

7.1.4 Coding Guidelines

Practice of coding style varies with organizations, operating systems and language of coding itself.
The following coding elements may be defined under coding guidelines of an organization:
Naming conventions − This section defines how to name functions, variables, constants and global variables.
Indenting −This is the space left at the beginning of line, usually 2−8 whitespace or single tab.
White space −It is generally omitted at the end of line. Operators −Defines the rules of writing mathematical, assignment and logical operators. For example, assignment operator '=' should have space before and after it, as in "x = 2".
Control Structures − The rules of writing if-then-else, case-switch, while-until and for control flow statements solely and in nested fashion.
Line length and wrapping− Defines how many characters should be there in one line, mostly a line is 80 characters long. Wrapping defines how a line should be wrapped, if is too long.
Functions − This defines how functions should be declared and invoked, with and without parameters.
Variables − This mentions how variables of different data types are declared and defined.
Comments −This is one of the important coding components, as the comments included in the code describe what the code actually does and all other associated descriptions. This section also helps creating help documentations for other developers.

7.1.5 Software Documentation

Software documentation is an important part of software process. A well written document provides a great tool and means of information repository necessary to know about software process. Software documentation also provides information about how to use the product.

A well-maintained documentation should involve the following documents:

Requirement documentation - This documentation works as key tool for software designer, developer and the test team to carry out their respective tasks. This document contains all the functional, non-functional and behavioral description of the intended software.

Source of this document can be previously stored data about the software, already running software at the client's end, client's interview, questionnaires and research. Generally it is stored in the form of spreadsheet or word processing document with the high-end software management team.

This documentation works as foundation for the software to be developed and is majorly used in verification and validation phases. Most test-cases are built directly from requirement documentation.

Software Design documentation - These documentations contain all the necessary information, which are needed to build the software. It contains: (a) High-level software architecture, (b) Software design details, (c) Data flow diagrams, (d) Database design

These documents work as repository for developers to implement the software. Though these documents do not give any details on how to code the program, they give all necessary information that is required for coding and implementation. Technical documentation - These documentations are maintained by the developers and actual coders. These documents, as a whole, represent information about the code. While writing the code, the programmers also mention objective of the code, who wrote it, where will it be required, what it does and how it does, what other resources the code uses, etc.

The technical documentation increases the understanding between various programmers working on the same code. It enhances re-use capability of the code. It makes debugging easy and traceable.

There are various automated tools available and some comes with the programming language itself. For example java comes JavaDoc tool to generate technical documentation of code.

User documentation - This documentation is different from all the above explained. All previous documentations are maintained to provide information about

the software and its development process. But user documentation explains how the software product should work and how it should be used to get the desired results.
These documentations may include, software installation procedures, how-to guides, user-guides, un-installation method and special references to get more information like license updation etc.738777

7.1.6 Software Implementation Challenges

There are some challenges faced by the development team while implementing the software. Some of them are mentioned below:
Code-reuse - Programming interfaces of present-day languages are very sophisticated and are equipped huge library functions. Still, to bring the cost down of end product, the organization management prefers to re-use the code, which was created earlier for some other software. There are huge issues faced by programmers for compatibility checks and deciding how much code to re-use.
Version Management - Every time a new software is issued to the customer, developers have to maintain version and configuration related documentation. This documentation needs to be highly accurate and available on time.
Target-Host - The software program, which is being developed in the organization, needs to be designed for host machines at the customers end. But at times, it is impossible to design a software that works on the target machines.

CHAPTER 8

PROJECT IMPLEMENTATION

8.1 Introduction

In a proposed system, we are proposing an experiment on lung cancer disease with a limited set of supervised data. We are proposing a combination of a Convolutional neural network-based multimodal disease risk prediction model with higher accuracy. We are going to solve the accuracy issue in the diagnosis of lung cancer with accurate stage predictions.

8.2 Tools and Technologies Used

8.2.1 Hardware Resources Required

1. Processor - Pentium IV/Intel I3 core
2. Speed - 1.1 GHZ
3. RAM - 512 MB(min)
4. Hard disk - 20 GB
5. Keyboard - Standard Keyboard
6. Mouse - Two Or Three Button Mouse
7. Monitor - LED Monitor

8.2.2 Software Resources Required

Platform :

1. Operating System : Windows XP / 7
2. Programming Language : Python
3. Software Version : Pycharm above
4. Tools : pycharm
5. Front End : HTML, CSS
6. Data Base : Sqlite3

Tools:

1)Documentation:Texmaker 4.4.1
2)Digram:StarUML 2.5
3)Testing:Manual Testing

8.2.3 Technologies

- Python

Advantages of python:

- Lightweight
- Interpreter based
- Less lines of code
- Static HTML

8.2.4 Tools

- Pycharm

  Python support including language enhancements, search and refactoring, Quick Assist and Clean Up to migrate anonymous classes to lambda expressions and back, and new formatter options for lambdas.

  The python workbench provides a new dark theme which includes syntax highlighter settings for several programming languages.

Advantages

- Best Support for Latest python Technologies
- Fast and Smart Code Editing
- Easy and Efficient Project Management
- Rapid User Interface Development
- Support for Multiple Languages
- Cross Platform Support

8.3 Verification and Validation for Acceptance

V Model: V- model means Verification and Validation model. Just like the waterfall model, the V-Shaped life cycle is a sequential path of execution of processes. Each phase must be completed before the next phase begins. Testing of the product is planned in parallel with a corresponding phase of development. Under V-Model, the corresponding testing phase of the development phase is planned in parallel. So there are Verification phases on one side of the .V. and Validation phases on the other side. Coding phase joins the two sides of the VModel.

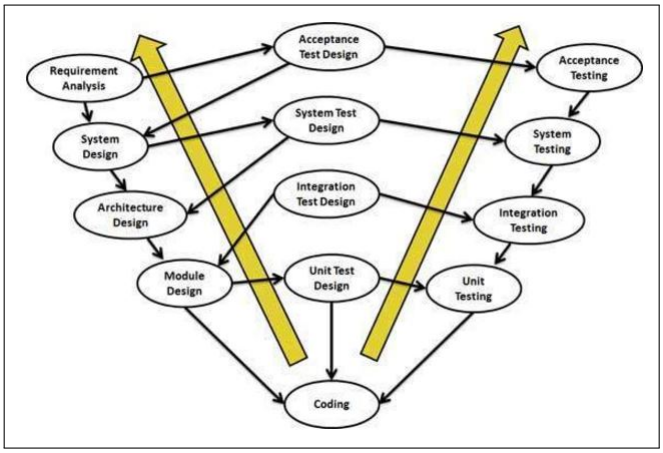


Figure 8.1: V-Model

CHAPTER 9

SOFTWARE TESTING

9.1 Introduction

Software testing is an activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. It is more than just running a program with the intention of finding faults. Every project is new with different parameters. No single yardstick maybe applicable in all circumstances. This is a unique and critical area with altogether different problems. Although critical to software quality and widely deployed by programs and testers. Software testing steel remains an art, due to limited understanding of principles of software. The difficulty stems from complexity of software. The purpose of software testing can be quality assurance, verification and validation or reliability estimation. Testing can be used as a generic metric as well. Software testing is a trade-off between budget,time and quality.

9.2 TYPES OF SOFTWARE TESTING

- Manual Testing
  Manual and Automated test are the types of software testing. We are doing a manual test for testing our system that is without using any automated tool or any script. In this type tester takes over the role of an end user and test the software to identify any unexpected behavior or bug. There are different stages for manual testing like unit testing, integration testing, system testing and user acceptance testing. Testers use test plan, test cases or test scenario to test the software to ensure the completeness of a testing. Manual testing also includes exploratory testing as a testers explore the software to identify the errors in it.

- Automated Testing
  Automation testing which is also known as Test Automation is when the tester writes scripts and uses software to test the software. This process involves automation of a manual process. Automation testing is used to re-run the test scenarios that were performed manually, quickly and repeatedly.

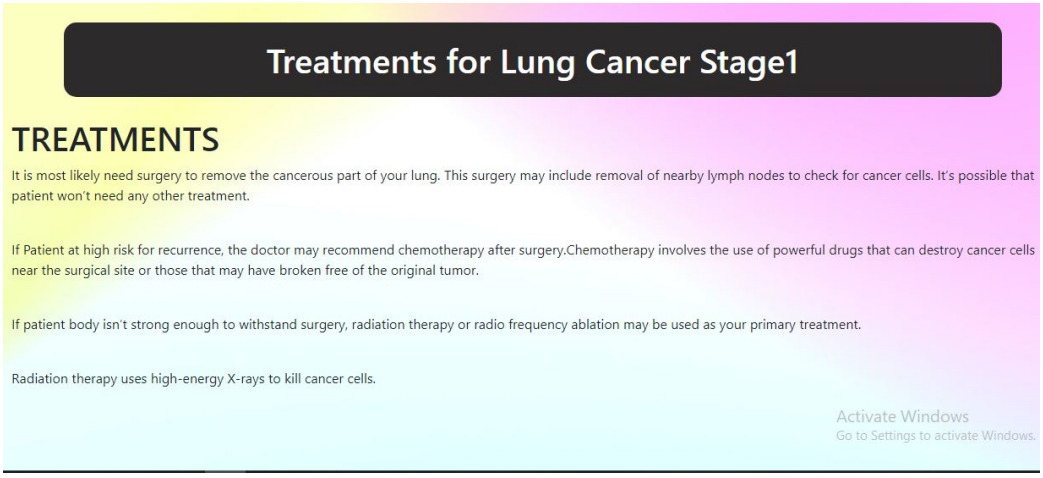
CHAPTER 10

RESULTS

10.1 Screen shots


LOGIN
Username...
Password...
Login
Don't have an account? Sign Up


Dashboard
LUNG CANCER DETECTION
Choose File
NO FILE CHOSEN
UPLOAD


Treatments for Lung Cancer Stage1
TREATMENTS
It is most likely need surgery to remove the cancerous part of your lung. This surgery may include removal of nearby lymph nodes to check for cancer cells. It's possible that patient won't need any other treatment.
If Patient at high risk for recurrence, the doctor may recommend chemotherapy after surgery.Chemotherapy involves the use of powerful drugs that can destroy cancer cells near the surgical site or those that may have broken free of the original tumor.
If patient body isn't strong enough to withstand surgery, radiation therapy or radio frequency ablation may be used as your primary treatment.
Radiation therapy uses high-energy X-rays to kill cancer cells.
Activate Windows
Go to Settings to activate Windows.

CHAPTER 11

DEPLOYMENT AND MAINTENANCE

11.1 Installation and Un-installation

In installation process it is a process of assembling the hardware components to your toll so that the automation can take place smoothly. The software installation part is so easy that user have to install pycharm(python Development Tool) and sqlite3 for database.

11.2 User help

Running the System
Step One
Open Pycharm
Open sqlite3
Step Two
Run the Project
Open in Browser

CHAPTER 12

CONCLUSIONS & FUTURE WORK

The proposed system proposes a hybrid approach for lung cancer detection systems over machine learning and CNN techniques. This system is a use of CNN algorithm which resolves the accuracy problem. The proposed system tries to improve accuracy and reduces the death rate. We get the 96.23% accuracy on 100 epochs. In future work we can implement on more diseases.

ANNEXURE A

LABORATORY ASSIGNMENTS ON PROJECT ANALYSIS OF ALGORITHMIC DESIGN

To develop the problem under consideration and justify feasibilty using concepts of knowledge canvas and IDEA Matrix. for IDEA Matrix and Knowledge canvas model. Case studies are given in this book. IDEA Matrix is represented in the following form. Knowledge canvas represents about identification of opportunity for product. Feasibility is represented w.r.t. business perspective.

A.1 THEORY

A.1.1 FEASIBILITY STUDY

A feasibility study is carried out to select the best system that meets performance requirements. The main aim of the feasibility study activity is to determine whether it would be financially and technically feasible to develop the product. The feasibility study activity involves the analysis of the problem and collection of all relevant information relating to the product such as the different data items which would be input to the system, the processing required to be carried out on these data, the out- put data required to be produced by the system as well as various constraints on the behavior of the system.
Technical Feasibility This is concerned with specifying equipment and software that will successfully satisfy the user requirement. The technical needs of the system may vary.
Economic Feasibility Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More commonly known as Cost / Benefit analysis, the procedure is to determine the benefits and savings that are expected from a proposed system and compare them with costs.
Operational Feasibility This is mainly related to human organizational and political aspects.

A.1.2 KNOWLEDGE CANVAS AND KNOWLEDGE FORCES INNOVATION

THROUGH ASSOCIATION
Knowledge canvas is one that depicts the knowledge forces and knowledge flow across the organization and extended organizations. It captures the current knowledge state and knowledge forces in the environment. It tries to build the bigger and broader knowledge scenario for you and your environment. It helps you to identify the knowledge opportunities, prospective knowledge partners, and the

knowledge losses. It is a simple representation of knowledge opportunities, with reference to the environment.

ANNEXURE B

LABORATORY ASSIGNMENTS ON PROJECT QUALITY AND RELIABILITY TESTING OF PROJECT DESIGN

It should include assignments such as

- Use of divide and conquer strategies to exploit distributed/parallel/concurrent processing of the above to identify object, morphisms, overloading in functions (if any), and functional relations and any other dependencies (as per requirements). It can include Venn diagram, state diagram, function relations, i/o relations; use this to derive objects, morphism, overloading
- Use of above to draw functional dependency graphs and relevant Software modeling methods, techniques including UML diagrams or other necessities using appropriate tools.
- Testing of project problem statement using generated test data (using mathematical models, GUI, Function testing principles, if any) selection and appropriate use of testing tools, testing of UML diagram's reliability. Write also test cases [Black box testing] for each identified functions. You can use Mathematica or equivalent open source tool for generating test data.
- Additional assignments by the guide. If project type as Entreprenaur, Refer [**?**],[**?**],[**?**], [**?**]

ANNEXURE C

PROJECT PLANNER

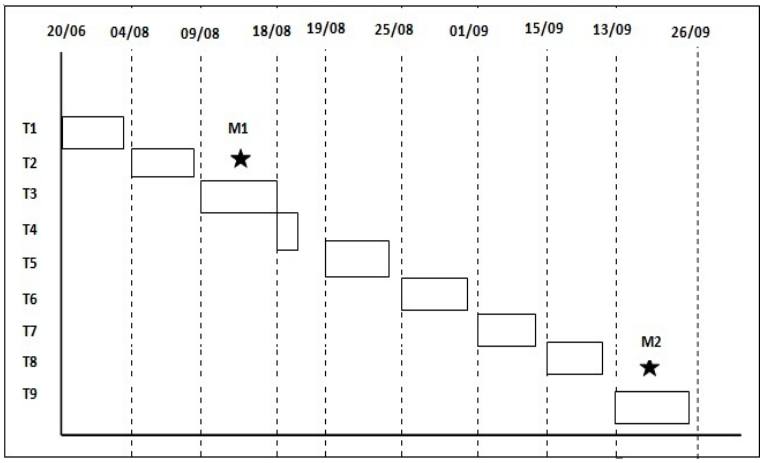


Figure C.1: Timeline Chart (I) (Actual Days)

Indicates Milestone
M1: Requirement Gathering
M2: UML diagrams

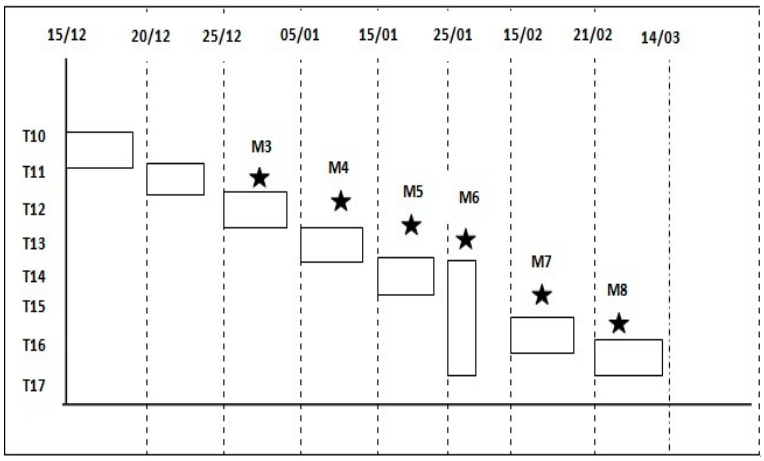


Figure C.2: Timeline Chart (II) (Expected Days)

Indicates Milestone
M3: .
M4:
M5:

M6:
M7:
M8:

ANNEXURE D

REVIEWERS COMMENTS OF PAPER SUBMITTED

(At-least one technical paper must be submitted in Term-I on the project design in the conferences/workshops in IITs, Central Universities or UoP Conferences or equivalent International Conferences Sponsored by IEEE/ACM)

1. Paper Title: Lung cancer detection using CNN
2. Name of the Conference/Journal where paper submitted : International Conference on Information, Networks and Communications (ICINC 2021)
3. Paper accepted/rejected : Accepted
4. Review comments by reviewer : Ok
5. Corrective actions if any : No

CERTIFICATE
OF PUBLICATION

IJIRCCE

International Journal of Innovative Research in Computer and Communication Engineering

Website: www.ijircce.com Email: ijircce@gmail.com

This is hereby Awarding this Certificate to

AKSHAY KALWAGHE

Department of Computer, Sahyadri Valley College of Engineering & Technology, Pune, India

Published a paper entitled

Lung Cancer Detection by CNN

in IJIRCCE, Volume 10, Issue 4, April 2022

Impact Factor 8.165

e-ISSN: 2320-9801
p-ISSN: 2320-9798

ISSN INTERNATIONAL STANDARD SERIAL NUMBER

INNO SPACE SJIF Scientific Journal Impact Factor

P. Kumar

Editor-in-Chief

CERTIFICATE
OF PUBLICATION

IJIRCCE

International Journal of Innovative Research in Computer and Communication Engineering

Website: www.ijircce.com Email: ijircce@gmail.com

This is hereby Awarding this Certificate to

PRATIBHA CARPE

Department of Computer, Sahyadri Valley College of Engineering & Technology, Pune, India

Published a paper entitled

Lung Cancer Detection by CNN

in IJIRCCE, Volume 10, Issue 4, April 2022

Impact Factor 8.165

e-ISSN: 2320-9801
p-ISSN: 2320-9798

ISSN INTERNATIONAL STANDARD SERIAL NUMBER

INNO SPACE SJIF Scientific Journal Impact Factor

P. Kumar

Editor-in-Chief

CERTIFICATE
OF PUBLICATION

IJIRCCE

International Journal of Innovative Research in Computer and Communication Engineering

Website: www.ijircce.com Email: ijircce@gmail.com

This is hereby Awarding this Certificate to

NUTAN VARPE

Department of Computer, Sahyadri Valley College of Engineering & Technology, Pune, India

Published a paper entitled

Lung Cancer Detection by CNN

in IJIRCCE, Volume 10, Issue 4, April 2022

Impact Factor 8.165

e-ISSN: 2320-9801
p-ISSN: 2320-9798

ISSN INTERNATIONAL STANDARD SERIAL NUMBER

INNO SPACE SJIF Scientific Journal Impact Factor

P. Kumar

Editor-in-Chief

CHAPTER 13

PLAGIARISM REPORT

13.1 ABSTRACT PLAGIARISM REPORT

Plagiarism Scan Report

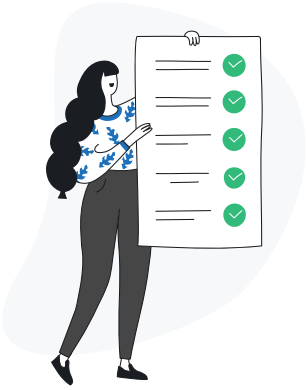
Report Generated on: May 23,2022

0%	100%	
Plagiarised	Unique	

Total Words:	**122**
Total Characters:	**721**
Plagiarized Sentences:	**0**
Unique Sentences:	**7 (100%)**

Content Checked for Plagiarism

Lung cancer is the cause of every sixth death around the world making it the second leading cause of death. Approximately 42 million people across the world suffer from cancer and this figure is continuously increasing. In India, approximately two and half million people are suffering from different types of cancer. If most of the cancers are detected in an early stage, then with the right remedy they can be cured. This paper offers details on a method in which CNN algorithm is used for the detection of illnesses like lung cancer. This paper also details the different machine learning techniques used to classify cancer into malignant and normal category. We get good accuracy on CNN which is 96.23% on 100 epochs.



No Plagiarism Found