A

MINI PROJECT REPORT

ON

EARLY DETECTION OF LUNG CANCER USING ARTIFICIAL INTELLIGENCE

A project report submitted to the
Jawaharlal Nehru Technological University
In partial fulfillment for the award

Bachelor of Technology
In
COMPUTER SCIENCE AND ENGINEERING

Submitted by

ANIKETH KULKARNI	**–**	**19RJ1A0507**
B.VAHANTH PHALGUNA	**–**	**19RJ1A0528**
AKULA MAHESH	**–**	**19RJ1A0502**
J.GANESH CHOWDARY	**–**	**18RJ1A0520**

Under the esteemed guidance of

Dr P. LALITHA KUMARI

MALLA REDDY INSTITUTE OF TECHNOLOGY

(Affiliated to JNTU, Hyderabad | Approved by AICTE, New Delhi)
Accredited by NBA, Certificated by ISO 9001:2015
Maisammaguda, Dhullapally, Via: Kompally, Hyderabad - 500100
2019 - 2023

MALLA REDDY INSTITUTE OF TECHNOLOGY

(Sponsored by Malla Reddy Educational Society)
Accredited by NBA, Certificated by ISO 9001:2015
Approved by AICTE & Affiliated to JNTU, Hyderabad
Maisammaguda, Dhullapally Post, (Via: Kompally), Secunderabad - 500100.

CERTIFICATE

This is to certify that minor project work entitled **"EARLY DETECTION OF LUNG CANCER USING ARTIFICIAL INTELLIGENCE"** is a bonafide work carried by **ANIKETH KULKARNI (19RJ1A0507), B. VAHANTH PHALGUNA (19RJ1A0528), AKULA MAHESH (19RJ1A0502), J.GANESH CHOWDARY (18RJ1A0520)** of **COMPUTER SCIENCE AND ENGINEERING** in **MALLA REDDY INSTITUTE OF TECHNOLOGY** and submitted to **JNTU UNIVERSITY, Hyderabad** in the partial fulfillment of the requirement for the award of **BACHELOR OF TECHNOLOGY**.

Project Guide	**Project Coordinator**	**Head of Department**
Dr P. LALITHA KUMARI	**Ms. K. BHAVANI**	**Dr K.R.N KIRAN KUMAR**

External Examiner

DECLARATION

We hereby declare that the project entitle **"EARLY DETECTION OF LUNG CANCER USING ARTIFICIAL INTELLIGENCE"** submitted to **Malla Reddy Institute of Technology**, affiliated to Jawaharlal Nehru Technological University Hyderabad (**JNTUH**) for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** is a result of work done by us.

It is further declared that the project report or any part thereof has not been previously submitted to any University or Institute for the award of degree.

ANIKETH KULKARNI - 19RJ1A0507
B. VAHANTH PHALGUNA - 19RJ1A0528
AKULA MAHESH - 19RJ1A0502
J.GANESH CHOWDARY - 18RJ1A0520

ACKNOWLEGEMENT

We are happy to express our deep sense of gratitude to the **Principal** of the college **Dr. K. Srinivasa Rao, Professor, Malla Reddy Institute of Technology** for having provided us the adequate facilities to pursue our project.

We would like to thank **Dr.K.R.N Kiran Kumar, Professor and Head, Department of Computer Science and Engineering, Malla Reddy Institute of Technology,** for having provided the freedom to use all the facilities available in the department, especially the laboratories and library.

We are also grateful to our project coordinator, **Ms. K. Bhavani**, **Assistant Professor**, **Department of Computer Science and Engineering**, **Malla Reddy Institute of Technology**, for extending her support and assisting us throughout our project work.

We are very grateful to our project guide, **Dr. P. Lalitha Kumari, Associate Professor, Department of Computer science and engineering**, **Malla Reddy Institute of Technology**, for her extensive patience and guidance throughout our project work.

We sincerely thank all the **Teaching and Non-teaching staff** of the department of **Computer Science and Engineering**, also like to thank our **Classmates** for their timely suggestions, healthy criticism and motivation during the course of our project work. We thank our **parents** who were the backbone behind our deeds.

Finally, we express our immense gratitude with pleasure to all **individuals** who have either directly or indirectly contributed to our need at right time for the development and success of our project work.

ABSTRACT

In this computer era we are totally going with the automation of everything, in the same way the medical industry is also automated with the help of image processing and data analytics. The best way to control the death cause by cancer is early detection. The medical image or a CT scan image is pre-processed. The contrast of the image is increased with the CLAHE Equalization technique. Then it is segmented with the help of random walk segmentation method.

In segmentation the three processes will happen the ROI of image is segmented and then the border correction is done. As third part the continuous pixel change is segmented. The classification is the major portion where the cancerous and non-cancerous is identified with the pre trained model. All the methods used above deals with the traditional way of image processing and data analytics. In Future this accuracy will be boosted with the modern XGboost algorithm where less data is used to get high accuracy.

The identification of lung cancer at the early stage is very demanding and difficult task due to construction of the cell. The cancer grows in the body when cancerous cells start to develop uncontrollably. The image processing plays vital role in the prediction of lung cancer at early stage which is also helpful in treatment to avoid the lung cancer. This proposed system is developed to detect lung cancer at early stage with the help of image processing techniques and artificial neural network classifier to design computer-based diagnosis system. In this system, during the preprocessing step, several image enhancing techniques, masks are applied using morphological operations and thresholding technique, which eliminates background and surrounding tissue. Region of interest (ROI) is calculated using region-based segmentation algorithm. Circle fit algorithm is used to extract the desired nodule. Radius, Mean Intensity, Area, Euler Number and ECD features are extracted in feature extracting step. Finally, Back propagation algorithm is used to train Artificial Neural Network (ANN) in categorization stage.v.

INDEX

TABLE OF CONTENTS **PAGE NO**

LIST OF FIGURES

LIST OF SCREENSHOTS

1. INTRODUCTION

Lung cancer is a type of cancer that begins in the lungs. Your lungs are two spongy organs in your chest that take in oxygen when you inhale and release carbon dioxide when you exhale. Lung cancer is the leading cause of cancer deaths worldwide. People who smoke have the greatest risk of lung cancer, though lung cancer can also occur in people who have never smoked. The risk of lung cancer increases with the length of time and number of cigarettes you've smoked. If you quit smoking, even after smoking for many years, you can significantly reduce your chances of developing lung cancer.

Lung cancer growth has turned out to be a standout amongst the most widely recognized reasons for disease in the two people. Countless bite the dust each year because of lung malignancy. The illness has diverse stages where by it begins from the little tissue and spreads all through the distinctive territories of the lungs by a procedure called metastasis. It is the uncontrolled development of undesirable cells in the lungs. It is assessed that around 12,203 people had lung disease in 2016, 7130 guys and 5073 females; passing from lung malignant growth in 2016 were 8839. . Biomedical image handling is the most recent rising apparatus in medicinal research utilized for the early recognition of malignancies. Biomedical image handling strategies can be utilized in the restorative field to analysis maladies at the beginning time. It utilizes biomedical images, for example, X-beams, Computed innovation and MRIs. The principle commitment of image handling in the restorative field is to analysis the malignant growth at the beginning time, expanding survival rates.

The main cause of lung cancer is growth of cells in lung tissue which is irregular and out of control. One of the reasons is smoking. If it is detected earlier, then there will be a good chance of curing. Screening is the one of the important step for lung cancer detection. Screening is the process used to detect and identify the nodule. A nodule appear as round and white in co lour on a Computed Tomography scans images or an chest X-ray. There are two types of nodules one is a benign and second one is a malignant. A nodule with diameter 3 cm or less is called a Pulmonary or non-cancerous nodule. These nodules are also called as benign. A nodule whose diameter is larger than 3 cm is poisonous and called as malignant nodule. Malignant nodule should be identified as early possible because it is likely to be cancerous nodule. To check whether these nodules are expanding, they are needed to be observed over the time. If there is a change in the size of nodule and it is growing then there is a probability of getting cancer. So, a nodule should be observed. As compared with other types of cancer, the long-term endurance rate of lung cancer patient is very lo w. So, the identification of lung cancer at early stage is very important and it provides vital research platform in medical image processing field.

The time factor is basic for tumors of the mind, the lungs, and bosoms. image handling can identify these malignant growths in the early periods of the maladies encouraging an early treatment process. The image preparing procedure comprises of four essential stages, pre-handling, division, including extraction and grouping. This paper presents image preparing procedures whereby the CT examine image is utilized as information image, is handled and beginning period lung disease is distinguished utilizing an SVM (bolster vector machine) calculation as a classifier in the grouping stage to improve exactness, affectability, and explicitness. First the image is pre-handled and divided. After that Features are removed from the sectioned image lastly the image is delegated ordinary or destructive. Advanced image handling is the utilization of PC calculations toper form image preparing on computerized images. As a subfield of advanced flag 2 preparing, computerized image handling has numerous points of interest over simple image preparing. It permits a lot more extensive scope of calculations to be connected to the information data—the point of advanced image handling is to improve the image information (Features) by stifling undesirable mutilations as well as upgrade of some vital image includes with the goal that our AI Computer Vision models can profit by this improved information to take a shot at. Feature extraction begins from an underlying arrangement of estimated information and assembles determined qualities (Features) proposed to be useful and non-excess, encouraging the resulting learning and speculation steps, and at times prompting better human elucidations. Feature extraction is a dimensionality decrease process, where an underlying arrangement of crude factors is diminished to progressively sensible gatherings (Features) for handling, while still precisely and totally portraying the first informational collection.

The chosen Features are relied upon to contain the pertinent data from the information, with the goal that the ideal undertaking can be performed by utilizing this decreased portrayal rather than the total introductory information. Feature extraction includes lessening the measure of assets required to depict a substantial arrangement of information. When performing an examination of complex information one of the serious issues originates from the quantity of factors included. Examination with countless for the most part requires a lot of memory and calculation control likewise it might make an arrangement calculation overfit to preparing tests and sum up in effectively to new examples. Feature extraction is a general term for strategies for building mixes of the factors to get around these issues while as yet portraying 3 the information with adequate exactness. Many AI specialists trust that appropriately streamlined component extraction is the way to successful model development.

2. LITERATURE SURVEY

2.1 LUNG NODULE DETECTION IN CT IMAGES USING IMAGE SEGMENTATION METHODS.

Author: Nanusha

Abstract:.

The paper presents a complete computer-aided detection (CAD) system for the detection of lung nodules in computed tomography images. A new mixed feature selection and classification methodology is applied for the first time on a difficult medical image analysis problem. The CAD system was trained and tested on images from the publicly available Lung Image Database Consortium (LIDC) on the National Cancer Institute website. The detection stage of the system consists of a nodule segmentation method based on nodule and vessel enhancement filters and a computed divergence feature to locate the center of the nodule clusters.

The detection and segmentation of lung nodules based on computer tomography images (CT) is a basic and significant step to achieve the robotic needle biopsy. In this paper, we reviewed some typical segmentation algorithms, including thresholding, active contour, differential operator, region growing and watershed. To analyse their performance on lung nodule detection, we applied them to four CT images of different kinds of lung nodules. The results show that thresholding, active contour and differential operator do well in the segmentation of solitary nodules, while region growing has an advantage over the others on segmenting nodules adhere to vessels. For segmentation of semi-transparent nodules, differential operator is an especially suitable choice. Watershed can segment nodules adhere to vessels and semi-transparent nodules well, but it has low sensitivity in solitary nodule.

In the subsequent classification stage, invariant features, defined on a gauge coordinates system, are used to differentiate between real nodules and some forms of blood vessels that are easily generating false positive detections. The performance of the novel feature-selective classifier based on genetic algorithms and artificial neural networks (ANNs) is compared with that of two other established classifiers, namely, support vector machines (SVMs) and fixed topology neural networks. A set of 235 randomly selected cases from the LIDC database was used to train the CAD system. The system has been tested on 125 independent cases from the LIDC database.

2.2 Segmentation and Image Analysis of Abnormal Lungs at CT: Current Approaches, Challenges, and Future Trend

AUTHOR: Awais Mansoor Ph.D et al,

Abstract:

Our aim is to review and explain the capabilities and performance of currently available approaches for segmentation of lungs with pathologic conditions on chest CT images, with illustrations to give radiologists a better understanding of potential choices for decision support in everyday practice. The computer-based process of identifying the boundaries of lung from surrounding thoracic tissue on computed tomographic (CT) images, which is called segmentation, is a vital first step in radiologic pulmonary image analysis. Many algorithms and software platforms provide image segmentation routines for quantification of lung abnormalities; however, nearly all of the current image segmentation approaches apply well only if the lungs exhibit minimal or no pathologic conditions. When moderate to high amounts of disease or abnormalities with a challenging shape or appearance exist in the lungs, computer-aided detection systems may be highly likely to fail to depict those abnormal regions because of inaccurate segmentation methods. In particular, abnormalities such as pleural effusions, consolidations, and masses often cause inaccurate lung segmentation, which greatly limits the use of image processing methods in clinical and research contexts. In this review, a critical summary of the current methods for lung segmentation on CT images is provided, with special emphasis on the accuracy and performance of the methods in cases with abnormalities and cases with exemplary pathologic findings. The currently available segmentation methods can be divided into five major classes: (a) thresholding-based, (b) region-based, (c) shape-based, (d) neighboring anatomy–guided, and (e) machine learning–based methods. The feasibility of each class and its shortcomings are explained and illustrated with the most common lung abnormalities observed on CT images. In an overview, practical applications and evolving technologies combining the presented approaches for the practicing radiologist are detailed.

2.3 Automatic detection of a tiny lung nodule on CT utilized in a local density maximum algorithms

Author: Binsheng Zhao, Gordon Gamsu

Due to Increase CT offer higher resolution and faster required time. This has to be result in a chance to distinguish a little lung knob which might be speak to a lung diseases at an early and potential more fix stage. Anyway, in a current clinical practice hundred of a such meager segment of CT picture are create for every patient and is assess by a radiology from a conventional perspective of taking a gander at each picture in a pivotal mode. This outcome in a possibility to miss little knob and potential miss a disease stage. In a paper they present a PC technique for robotization recognize of little lung knob on a multi cut CT picture. The technique comprise of three stages for example

1. Division of a lungs from the other anatomic structures
2. Identification of a knob up-and-comer in an extricated lung and
3. Decrease of a bogus positive among an identify knob applicant. A 3D lung picture can be extricated by distinguish a thickness histogram of a volume lung picture follow by an intricacy activity. Max thickness build incorporate a knob sprinkle all over a lung can be recognize by utilizes a nearby thickness greatest calculation.

Data of a knob, for example, a size and strong shape into a calculation to lessen an identify knob applicant. The strategy was applied to an identification of a PC reproduce little lung knob and accomplish an affectability of 84% with overall, five bogus positive outcome for every output.

2.4 Quantification of the Nodule Detection in Chest CT

Author: Farag, Shireen Y. Elhabian, Salwa A. Elshazly

This paper examines a detection step in a automatic detection and classified of lung nodule from low dose CT scan. They give an approach to estimated a gray level intensity distribution and a figure of merit of a size of appropriate template. A data driven approach is used to be design the template. The paper represent broad study of a sensitive and specify of a nodule detected step in which is a quality of a nodule model is a great factor. Now validity of a detection approach on label clinical dataset from a

Early Lung Cancer Action Project screen study is conduct. This paper show a relationship between a spatial support of a nodule template and a resolution of the lung cancer CT image which can be use to automatically select a template size. The paper also show that isotropic template which do not provide adequate detection rate of a clear nodule. The nodule models in a paper can be use in various machine learning approach for automatic nodule detect and classification.

2.5 SOFTWARE ENVIRONMENT

2.5.1 PYTHON OVERVIEW

Python is a high-level, interpreted scripting language developed in the late 1980s by Guido van Rossum at the National Research Institute for Mathematics and Computer Science in the Netherlands. The initial version was published at the alt. Sources newsgroup in 1991, and version 1.0 was released in 1994.

Python 2.0 was released in 2000, and the 2.x versions were the prevalent releases until December 2008. At that time, the development team made the decision to release version 3.0, which contained a few relatively small but significant changes that were not backward compatible with the 2.x versions. Python 2 and 3 are very similar, and some features of Python 3 have been backported to Python 2. But in general, they remain not quite compatible.

Both Python 2 and 3 have continued to be maintained and developed, with periodic release updates for both. As of this writing, the most recent versions available are 2.7.15 and 3.6.5. However, an official End of Life date of January 1, 2020 has been established for Python 2, after which time it will no longer be maintained. If you are a newcomer to Python, it is recommended that you focus on Python 3, as this tutorial will do.

Python is still maintained by a core development team at the Institute, and Guido is still in charge, having been given the title of BDFL (Benevolent Dictator For Life) by the Python community. The name Python, by the way, derives not from the snake, but from the British comedy troupe Monty Python’s Flying Circus, of which Guido was, and presumably still is, a fan. It is common to find references to Monty Python sketches and movies scattered throughout the Python documentation.

2.5.2 WHY TO CHOOSE PYTHON

If you’re going to write programs, there are literally dozens of commonly used languages to choose from. Why choose Python? Here are some of the features that make Python an appealing choice.

2.5.2.1 PYTHON IS POPULAR

Python has been growing in popularity over the last few years. The 2018 Stack Overflow Developer Survey ranked Python as the 7th most popular and the number one most wanted technology of the year. World-class software development countries around the globe use Python every single day.

According to research by Dice Python is also one of the hottest skills to have and the most popular programming language in the world based on the Popularity of Programming Language Index.

Due to the popularity and widespread use of Python as a programming language, Python developers are sought after and paid well. If you’d like to dig deeper into Python salary statistics and job opportunities, you can do so here.

2.5.2.2 PYTHON IS INTERPRETED

Many languages are compiled, meaning the source code you create needs to be translated into machine code, the language of your computer’s processor, before it can be run. Programs written in an interpreted language are passed straight to an interpreter that runs them directly.

This makes for a quicker development cycle because you just type in your code and run it, without the intermediate compilation step.

One potential downside to interpreted languages is execution speed. Programs that are compiled into the native language of the computer processor tend to run more quickly than interpreted programs. For some applications that are particularly computationally intensive, like graphics processing or intense number crunching, this can be limiting.

In practice, however, for most programs, the difference in execution speed is measured in milliseconds, or seconds at most, and not appreciably noticeable to a human user. The expediency of coding in an interpreted language is typically worth it for most applications.

2.5.2.3 PYTHON IS FREE

The Python interpreter is developed under an OSI-approved open-source license, making it free to install, use, and distribute, even for commercial purposes.

A version of the interpreter is available for virtually any platform there is, including all flavors of Unix, Windows, macOS, smartphones and tablets, and probably anything else you ever heard of. A version even exists for the half dozen people remaining who use OS/2.

2.5.2.4 PYTHON IS PORTABLE

Because Python code is interpreted and not compiled into native machine instructions, code written for one platform will work on any other platform that has the Python interpreter installed. (This is true of any interpreted language, not just Python.)

2.5.2.5 PYTHON IS SIMPLE

As programming languages go, Python is relatively uncluttered, and the developers have deliberately kept it that way.

A rough estimate of the complexity of a language can be gleaned from the number of keywords or reserved words in the language. These are words that are reserved for special meaning by the compiler or interpreter because they designate specific built-in functionality of the language.

Python 3 has 33 keywords, and Python 2 has 31. By contrast, C++ has 62, Java has 53, and Visual Basic has more than 120, though these latter examples probably vary somewhat by implementation or dialect.

Python code has a simple and clean structure that is easy to learn and easy to read. In fact, as you will see, the language definition enforces code structure that is easy to read.

But it’s not that simple for all its syntactical simplicity, Python supports most constructs that would be expected in a very high-level language, including complex dynamic data types, structured and functional programming, and object-oriented programming.

Additionally, a very extensive library of classes and functions is available that provides capability well beyond what is built into the language, such as database manipulation or GUI programming.

Python accomplishes what many programming languages don’t: the language itself is simply designed, but it is very versatile in terms of what you can accomplish with it.

2.5.2.6 PYTHON USED FOR

Python is used by hundreds of thousands of programmers and is used in many places. Sometimes only Python code is used for a program, but most of the time it is used to do simple jobs while another programming language is used to do more complicated tasks.

Its standard library is made up of many functions that come with Python when it is installed. On the Internet there are many other libraries available that make it possible for the Python language to do more things. These libraries make it a powerful language; it can do many different things.

Some things that Python is often used for are:

- Web development
- Scientific programming
- Desktop GUIs
- Network programming
- Game programming
- App development
- Data engineering

2.6 ADVANTAGES OF PYTHON OVER OTHER PROGRAMMING LANGUAGES.

1. Less Coding

Almost all of the tasks done in Python requires less coding when the same taskis done in other languages. Python also has an awesome standard library support, so you don't have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

3. Python is for everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and machine learning, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

2.7 DISADVANTAGES OF PYTHON

So far, we've seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let's now see the downsides of choosing Python over another language.

1. Speed Limitations

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in slow execution. This, however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the client-side. Besides that, it is rarely ever used to implement smartphone- based applications. One such application is called Carbonnelle. The reason it is not so famous despite the existence of Brython is that it isn't that secure

. 3. Design Restrictions

As you know, Python is dynamically-typed. This means that you don't need to declare the type of variable while writing the code. It uses duck-typing. But wait, what's that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can raise run-time errors.

4. Underdeveloped Database Access Layers

Compared to more widely used technologies like JDBC (Java DataBase Connectivity) and ODBC (Open DataBase Connectivity), Python's database 12 access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

2.8 PYTHON INSTALLATION PROCESS

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

2.8.1 INSTALL PYTHON STEP-BY-STEP IN WINDOWS OR OTHER OPERATING SYSTEM.

Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

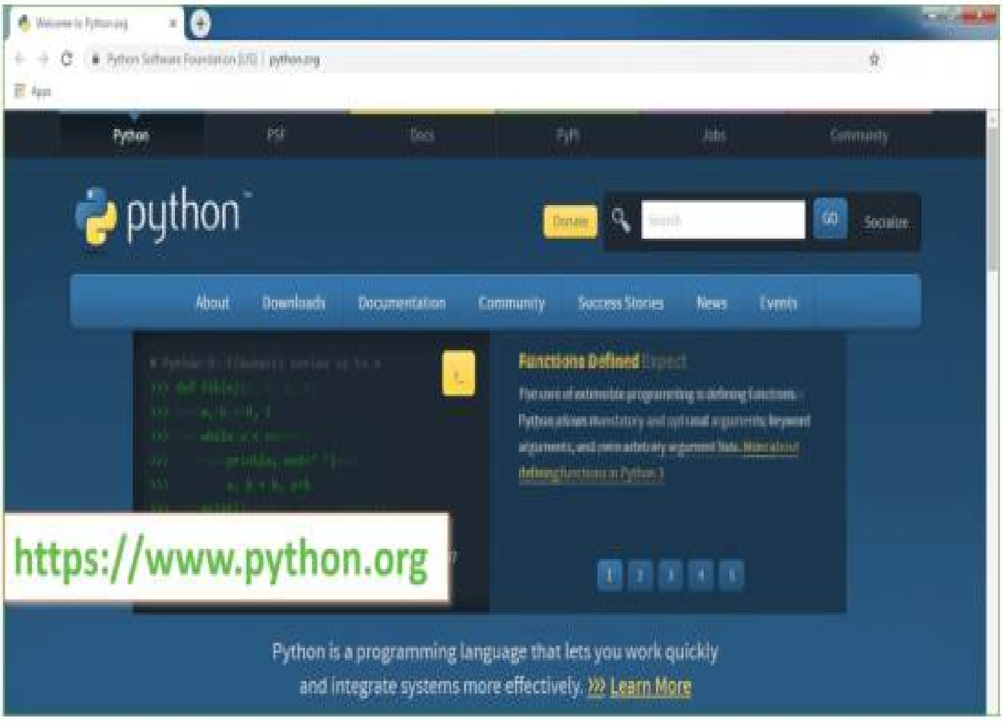
Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a **Windows 64-bit operating system**. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheatsheet here.The steps on how to install Python on Windows 10, 8 and 7 are **divided into 4 parts** to help understand better.

2.8.1.1 DOWNLOAD THE CORRECT VERSION INTO THE SYSTEM

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: https://www.python.org

Now, check for the latest and the correct version for your operating system

.



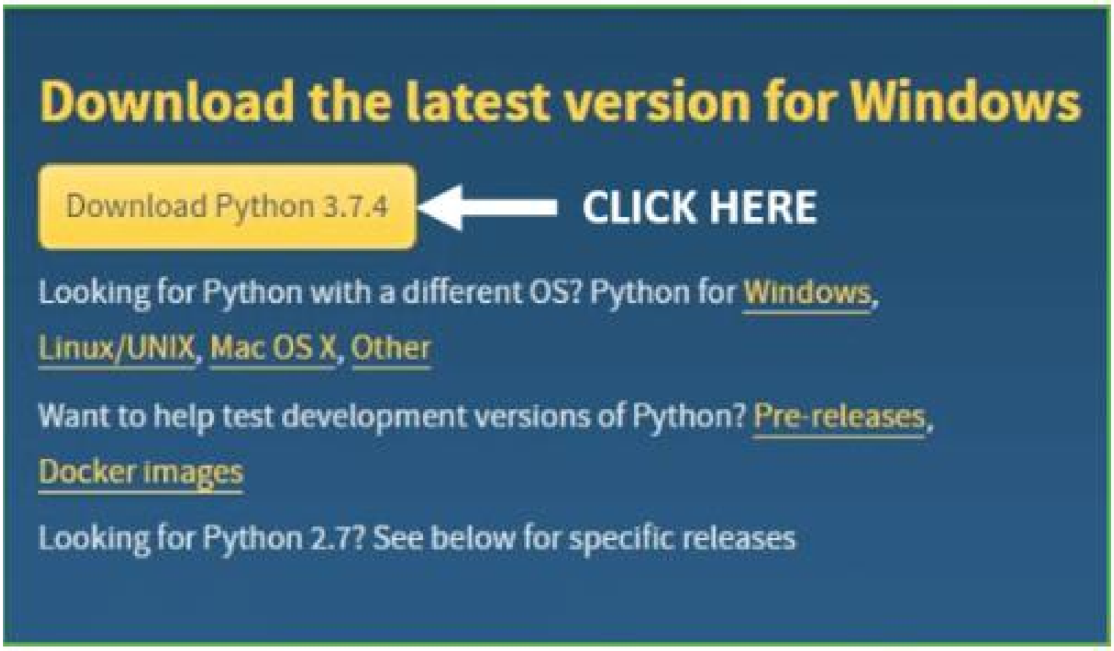
Functions Defined

The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. More about defining functions in Python 3

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

Figure-2.8.1.1.1 To select the require Python version.

Step 2: Click on the Download Tab.



Download the latest version for Windows

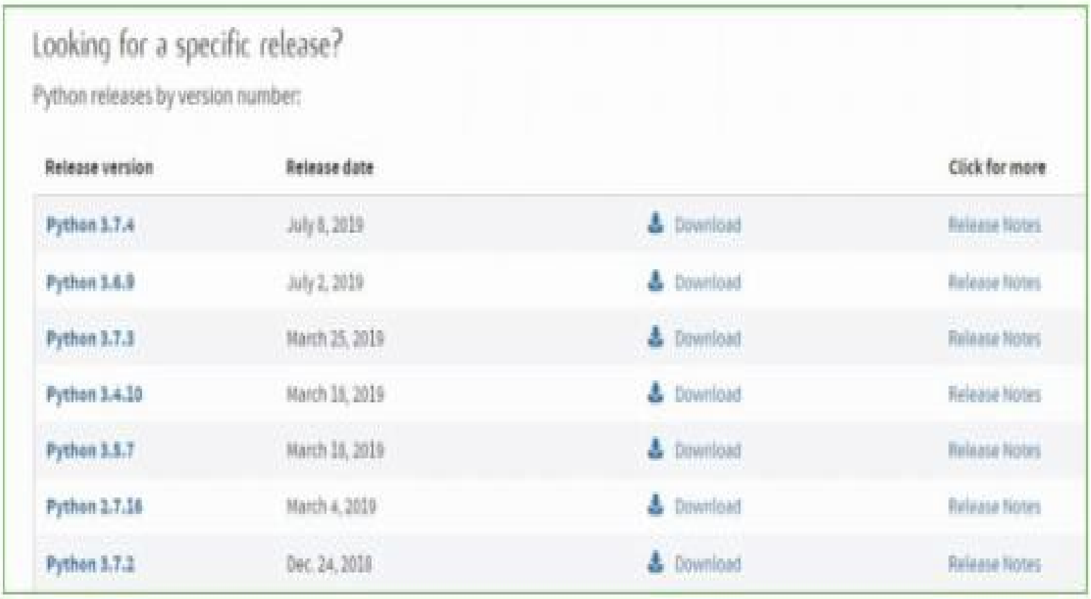
Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Pre-releases, Docker images

Looking for Python 2.7? See below for specific releases

Figure-2.8.1.1.2 To download the python version

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4



Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.7.4	July 8, 2019	Download	Release Notes
Python 3.6.9	July 2, 2019	Download	Release Notes
Python 3.7.3	March 25, 2019	Download	Release Notes
Python 3.4.10	March 18, 2019	Download	Release Notes
Python 3.5.7	March 18, 2019	Download	Release Notes
Python 2.7.16	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec. 24, 2018	Download	Release Notes

Figure-2.8.1.1.3 To select python file based on configuration of system

Step 4: Scroll down the page until you find the Files option and download the file.

Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.

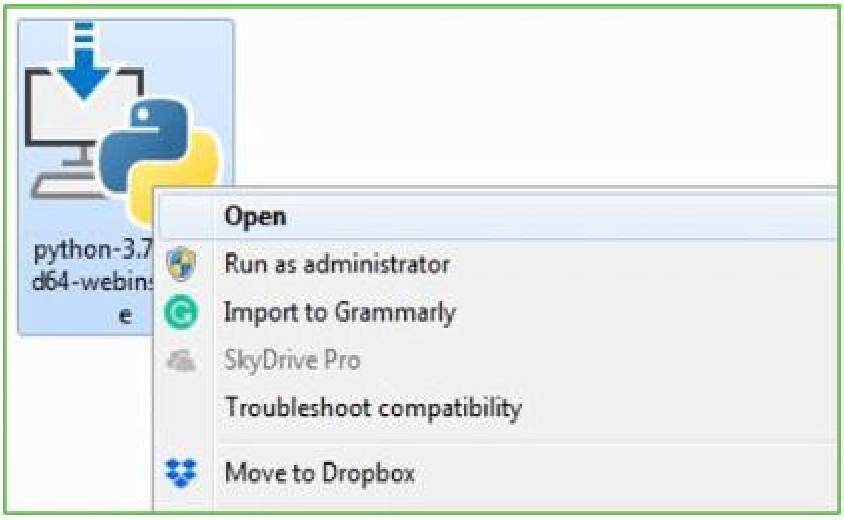
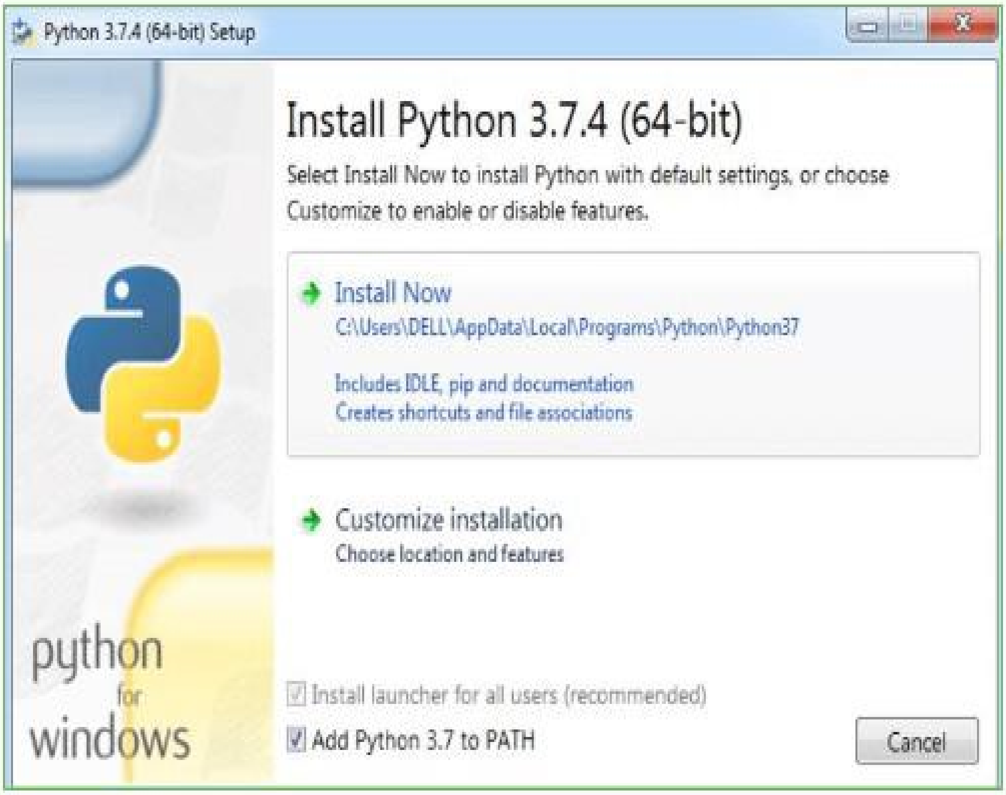


Figure-2.8.1.1.4 run program

Step 2: Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.


Python 3.7.4 (64-bit) Setup

Install Python 3.7.4 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\DELL\AppData\Local\Programs\Python\Python37

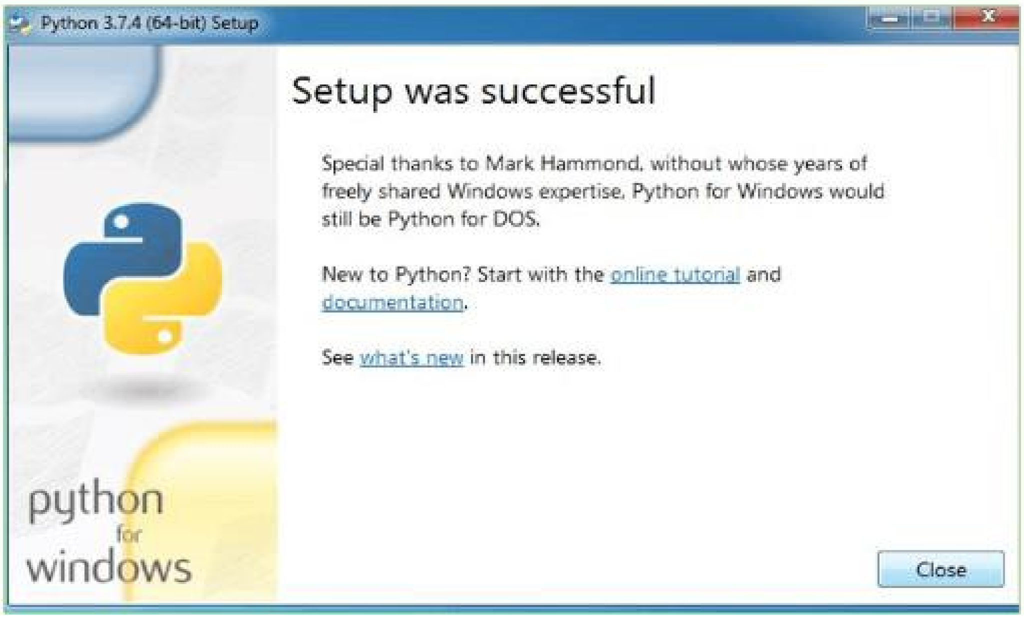
Includes IDLE, pip and documentation
Creates shortcuts and file associations

Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☑ Add Python 3.7 to PATH

Figure-2.8.1.1.5 To set path of python

Step 3: Click on Install NOW After the installation is successful. Click on Close.


Python 3.7.4 (64-bit) Setup

Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the online tutorial and documentation.

See what's new in this release.

Figure-2.8.1.6 Install sucess

With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

2.8.1.2 VERIFY THE PYTHON INSTALLATION

Step 1: Click on Start

Step 2: In the Windows Run Command, type “cmd”.

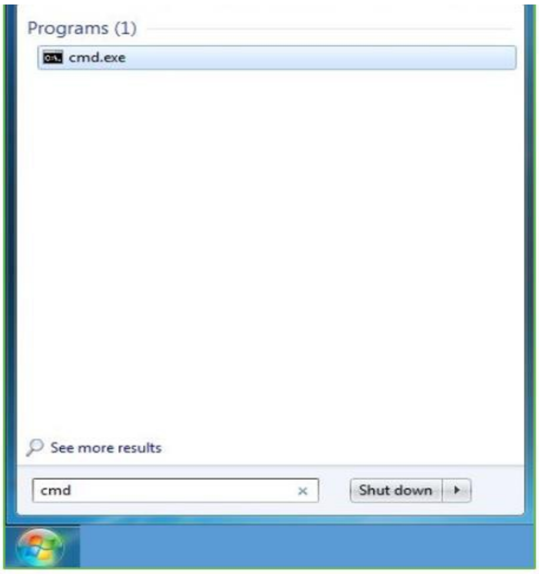
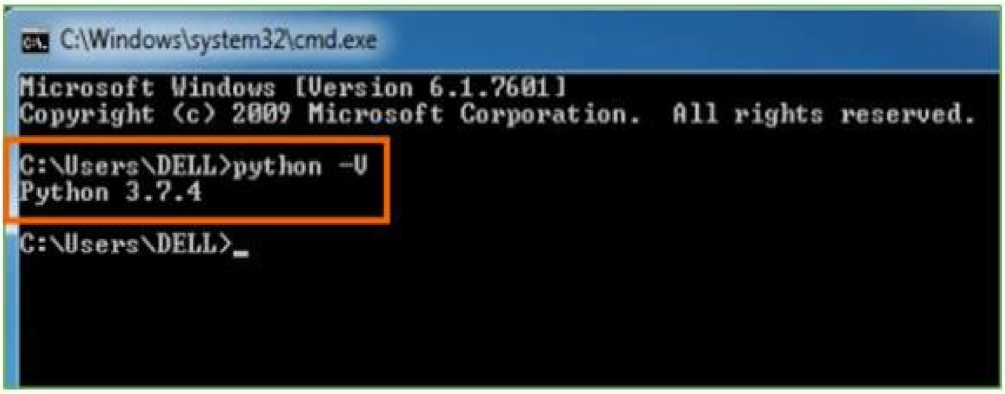


Figure-2.8.2.2.1 To run CMD

Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type **python –V** and press Enter.



C:\Windows\system32\cmd.exe

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\DELL>python -V
Python 3.7.4

C:\Users\DELL>_
```

Figure-2.8.2.2.2 Check the python version

Step 5: You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

2.9 WHAT IS MACHINE LEARNING

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models *tunable parameters* that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain.Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

2.9.1 CATEGORIES OF MACHINE LEARNING

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into *classification* tasks and *regression* tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as *clustering* and *dimensionality reduction.* Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

2.9.2 NEED FOR MACHINE LEARNING

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven't surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, "to make decisions, based on data, with efficiency and scale".

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can't do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

2.9.3 CHALLENGES IN MACHINES LEARNING

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

Quality of data − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

Time-Consuming task − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

Lack of specialist persons − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

No clear objective for formulating business problems − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

Issue of overfitting & underfitting − If the model is overfitting or underfitting, it cannot be represented well for the problem.

Curse of dimensionality − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

Difficulty in deployment − Complexity of the ML model makes it quite difficult to be deployed in real life.

2.9.4 APPLICATIONS OF MACHINES LEARNING

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML −

- Emotion analysis
- Sentiment analysis
- Error detection and prevention
- Weather forecasting and prediction
- Stock market analysis and forecasting
- Speech synthesis
- Speech recognition
- Customer segmentation
- Object recognition
- Fraud detection
- Fraud prevention

- Recommendation of products to customer in online shopping

(a) Terminologies of Machine Learning

- **Model –** A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
- **Feature –** A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.
- **Target (Label) –** A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
- **Training –** The idea is to give a set of inputs(features) and it's expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
- **Prediction –** Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

(b) Types of Machine Learning

- **Supervised Learning –** This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.
- **Unsupervised Learning –** This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
- **Semi-supervised Learning** – This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
- **Reinforcement Learning** – This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

2.9.5 ADVANTAGES OF MACHINE LEARNING

1. Easily identifies trends and patterns -

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

2. No human intervention needed (automation)

With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

3. Continuous Improvement

As **ML algorithms** gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

4. Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

5. Wide Applications

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

2.9.6 DISADVANTAGES OF MACHINE LEARNING

1. Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

2. Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

3. Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

4. High error-susceptibility

Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

2.10 WHAT IS DEEP LEARNING?

Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain—albeit far from matching its ability—allowing it to “learn” from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy.

Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention. Deep learning technology lies behind everyday products and services (such as digital assistants, voice-enabled TV remotes, and credit card fraud detection) as well as emerging technologies (such as self-driving cars).

2.10.1 HOW DEEP LEARNING WORKS

Deep learning neural networks, or artificial neural networks, attempts to mimic the human brain through a combination of data inputs, weights, and bias. These elements work together to accurately recognize, classify, and describe objects within the data.

Deep neural networks consist of multiple layers of interconnected nodes, each building upon the previous layer to refine and optimize the prediction or categorization. This progression of computations through the network is called forward propagation. The input and output layers of a deep neural network are called *visible* layers. The input layer is where the deep learning model ingests the data for processing, and the output layer is where the final prediction or classification is made.

Another process called backpropagation uses algorithms, like gradient descent, to calculate errors in predictions and then adjusts the weights and biases of the function by moving backwards through the layers in an effort to train the model. Together, forward propagation and backpropagation allow a neural network to make predictions and correct for any errors accordingly. Over time, the algorithm becomes gradually more accurate.

The above describes the simplest type of deep neural network in the simplest terms. However, deep learning algorithms are incredibly complex, and there are different types of neural networks to address specific problems or datasets. For example,

- Convolutional neural networks (CNNs), used primarily in computer vision and image classification applications, can detect features and patterns within an image, enabling tasks, like object detection or recognition. In 2015, a CNN bested a human in an object recognition challenge for the first time.
- Recurrent neural network (RNNs) are typically used in natural language and speech recognition applications as it leverages sequential or times series data.

2.10.2 DEEP LEARNING APPLICATIONS

Real-world deep learning applications are a part of our daily lives, but in most cases, they are so well-integrated into products and services that users are unaware of the complex data processing that is taking place in the background. Some of these examples include the following:

Law Enforcement

Deep learning algorithms can analyze and learn from transactional data to identify dangerous patterns that indicate possible fraudulent or criminal activity. Speech recognition, computer vision, and other deep learning applications can improve the efficiency and effectiveness of investigative analysis by extracting patterns and evidence from sound and video recordings, images, and documents, which helps law enforcement analyze large amounts of data more quickly and accurately.

Financial Services

Financial institutions regularly use predictive analytics to drive algorithmic trading of stocks, assess business risks for loan approvals, detect fraud, and help manage credit and investment portfolios for clients.

Customer Services

Many organizations incorporate deep learning technology into their customer service processes. Chatbots—used in a variety of applications, services, and customer service portals—are a straightforward form of AI. Traditional chatbots use natural language and even visual recognition, commonly found in call center-like menus. However, more sophisticated chatbot solutions attempt to determine, through learning, if there are multiple responses to ambiguous questions. Based on the responses it receives, the chatbot then tries to answer these questions directly or route the conversation to a human user.

Virtual assistants like Apple's Siri, Amazon Alexa, or Google Assistant extends the idea of a chatbot by enabling speech recognition functionality. This creates a new method to engage users in a personalized way.

Healthcare

The healthcare industry has benefited greatly from deep learning capabilities ever since the digitization of hospital records and images. Image recognition applications can support medical imaging specialists and radiologists, helping them analyze and assess more images in less time.

2.11 DEEP LEARNING VS. MACHINE LEARNING

If deep learning is a subset of machine learning, how do they differ? Deep learning distinguishes itself from classical machine learning by the type of data that it works with and the methods in which it learns.

Machine learning algorithms leverage structured, labeled data to make predictions—meaning that specific features are defined from the input data for the model and organized into tables. This doesn’t necessarily mean that it doesn’t use unstructured data; it just means that if it does, it generally goes through some pre-processing to organize it into a structured format.

Deep learning eliminates some of data pre-processing that is typically involved with machine learning. These algorithms can ingest and process unstructured data, like text and images, and it automates feature extraction, removing some of the dependency on human experts. For example, let’s say that we had a set of photos of different pets, and we wanted to categorize by “cat”, “dog”, “hamster”, et cetera. Deep learning algorithms can determine which features (e.g. ears) are most important to distinguish each animal from another. In machine learning, this hierarchy of features is established manually by a human expert.

Then, through the processes of gradient descent and backpropagation, the deep learning algorithm adjusts and fits itself for accuracy, allowing it to make predictions about a new photo of an animal with increased precision.

Machine learning and deep learning models are capable of different types of learning as well, which are usually categorized as supervised learning, unsupervised learning, and reinforcement learning. Supervised learning utilizes labeled datasets to categorize or make predictions; this requires some kind of human intervention to label input data correctly. In contrast, unsupervised learning doesn't require labeled datasets, and instead, it detects patterns in the data, clustering them by any distinguishing characteristics. Reinforcement learning is a process in which a model learns to become more accurate for performing an action in an environment based on feedback in order to maximize the reward.

2.12 Convolutional Neural Networks (CNN)

2.12.1 What is CNN?

Within Deep Learning, a Convolutional Neural Network or CNN is a type of artificial neural network, which is widely used for image/object recognition and classification. Deep Learning thus recognizes objects in an image by using a CNN. CNNs are playing a major role in diverse tasks/functions like image processing 28 problems, computer vision tasks like localization and segmentation, video analysis, to recognize obstacles in self-driving cars, as well as speech recognition in natural language processing. As CNNs are playing a significant role in these fast-growing and emerging areas, they are very popular in Deep Learning.

A typical neural network will have an input layer, hidden layers, and an output layer. CNNs are inspired by the architecture of the brain. Just like a neuron in the brain processes and transmits information throughout the body, artificial neurons or nodes in CNNs take inputs, processes them and sends the result as output. The image is fed as input. The input layer accepts the image pixels as input in the form of arrays. In CNNs, there could be multiple hidden layers, which perform feature extraction from the image by doing calculations. This could include convolution, pooling, rectified linear units, and fully connected layers. Convolution is the first layer that does feature extraction from an input image. The fully connected layer classifies the object and identifies it in the output layer. CNNs are feed-forward networks in that information flow takes place in one direction only, from their inputs to their outputs. Just as artificial neural networks (ANN) are biologically inspired, so are CNNs. The visual cortex in the brain, which consists of alternating layers of simple and complex cells, motivates their architecture. CNN architectures come in several variations; however, in general, they consist of convolutional and pooling (or subsampling) layers, which are grouped into modules. Either one or more fully connected layers, as in a standard feed-forward neural network, follow these modules.

CNNs have fundamentally changed our approach towards image recognition as they can detect patterns and make sense of them. They are considered the most effective architecture for image classification, retrieval and detection tasks as the accuracy of their results is very high. They have broad applications in real-world tests, where they produce high-quality results and can do a good job of localizing and identifying where in an image a person/car/bird, etc., are. This aspect has made them the go-to method for predictions involving any image as an input. Acritical feature of CNNs is their ability to achieve 'spatial invariance', which implies that they can learn to recognize and extract image features anywhere in the image. There 29 is no need for manual extraction as CNNs learn features by themselves from the image/data and perform extraction directly from images. This makes CNNs a potent tool within Deep Learning for getting accurate results.

2.12.2 CNN Layers

A deep learning CNN consists of three layers: a convolutional layer, a pooling layer and a fully connected (FC) layer. The convolutional layer is the first layer while the FC layer is the last.

From the convolutional layer to the FC layer, the complexity of the CNN increases. It is this increasing complexity that allows the CNN to successively identify larger portions and more complex features of an image until it finally identifies the object in its entirety.

1. Convolutional layer

The majority of computations happen in the convolutional layer, which is the core building block of a CNN. A second convolutional layer can follow the initial convolutional layer. The process of convolution involves a kernel or filter inside this layer moving across the receptive fields of the image, checking if a feature is present in the image. Over multiple iterations, the kernel sweeps over the entire image. After each iteration a dot product is calculated between the input pixels and the filter. Thefinal output from the series of dots is known as a feature map or convolved feature. Ultimately, the image is converted into numerical values in this layer, which allows the CNN to interpret the image and extract relevant patterns from it.

2. Pooling layer

Like the convolutional layer, the pooling layer also sweeps a kernel or filter across the input image. But unlike the convolutional layer, the pooling layer reduces the 30 number of parameters in the input and also results in some information loss. On thepositive side, this layer reduces complexity and improves the efficiency of the CNN.

3. Fully connected layer

The FC layer is where image classification happens in the CNN based on thefeatures extracted in the previous layers. Here, fully connected means that all theinputs or nodes from one layer are connected to every activation unit or node of thenext layer. All the layers in the CNN are not fully connected because it would result in an unnecessarily dense network. It also would increase losses and affect the output quality, and it would be computationally expensive.

2.12.3 How CNN works?

A CNN can have multiple layers, each of which learns to detect the different features of an input image. A filter or kernel is applied to each image to produce an output that gets progressively better and more detailed after each layer. In the lower layers, the filters can start as simple features. At each successive layer, the filters increase in complexity to check and identify features that uniquely represent the input object. Thus, the output of each convolved image -- the partially recognized image after each layer -- becomes the input for the next layer. In the last layer, which is an FC layer, the CNN recognizes the image or the object it represents. With convolution, the input image goes through a set of these filters. As each filter activates certain features from the image, it does its work and passes on its output to the filter in the next layer. Each layer learns to identify different features and the operations end up being repeated for dozens, hundreds or even thousands of layers. Finally, all the image data progressing through the CNN's multiple layers allow the CNN to identify the entire object.

2.12.4 Steps involved in CNN algorithm

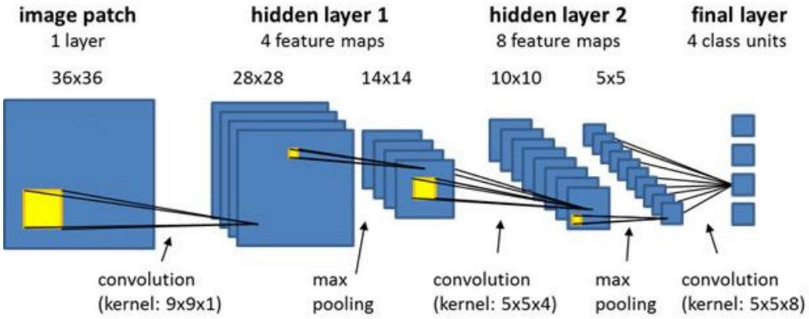
1) Feature Extraction: CNN compose of multiple layers and first layer define for feature extraction and this features will be extracted from given input image dataset or any other multidimensional dataset.

2) Feature Selection: Using this layer features will be selected by applyingalayer called pooling or max polling.

3) Activation module: using this module RELU will be applied on input features to remove out unimportant features and hold only relevant important features

4) Flatten: This layer will be define to convert multidimensional input features into single dimensional input array

5) Dense: This layer can be used to connect one layer to other layer to receiveinput features from previous layer to new layer to further filter input features in next layer to get most important features from dataset to have best prediction result.



3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The important step in the identification of lung cancer is detection of nodule. Image enhancement pre-processing is done again before extracting desired nodules. The image boundary connected objects are cleared. The gray thresholding for binarization, image background techniques are used for image pre-processing. Region based algorithm is used to segment the nodules from lung. Nodule with area between 75 pixels and 1000 pixels is identified and segmented for further process

DISADVANTAGES OF EXISTING SYSTEM

- Less accuracy
- Low Efficiency
- The CT filter picture is pre-prepared pursued by division of the ROI of the lung.
- Discrete waveform Transform is connected for picture pressure and highlights are extricated utilizing a GLCM.

3.2 PROPOSED SYSTEM

The median filter is generally used to diminish noise in an image. In the image, the median filter checks its nearby pixel to decide whether that neighbouring pixel is similar or not. In this filter it replaces pixel value with its neighbouring median pixel values. Histogram equalization technique is used to adjust image intensity to enhance contrast. It is the graphical interpretation of the image's pixel intensity values. It can be interpreted as the data structure that stores the frequencies of all the pixel intensity levels in the image.

ADVANTAGES OF PROPOSED SYSTEM

- High accuracy
- High efficiency
- The classification is the major portion where the cancerous and non-cancerous is identified with the pre trained model.

- The major method of prevention is the avoidance of risk factors, includingsmoking and air pollution.
- Treatment and long-term outcomes depend on the type of cancer, the stage(degree of spread), and the person's overall health.
- NSCLC is sometimes treated with surgery, whereas SCLC usually responds better to chemotherapy and radiotherapy.
- Lung carcinomas derive from transformed, malignant cells that originate as epithelial cells, or from tissues composed of epithelial cells. Other lung cancers, such as the rare sarcomas of the lung, are generated by the malignant transformation of connective tissues (i.e. nerve, fat, muscle, bone), which arise from mesenchymal cells. Lymphomas and melanomas (from lymphoid and melanocyte cell lineages) can also rarely result in lung cancer

4. FEASIBILITY STUDY

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

Three key considerations involved in the feasibility analysis are:

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

4.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

4.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

5. SYSTEM REQUIREMENTS

5.1 HARDWARE REQUIREMENTS

System	:	i3 or above.
Ram	:	4 GB.
Hard Disk	:	40 GB.

5.2 SOFTWARE REQUIRMENTS

Operating System	:	Windows8 or Above.
Coding Language	:	python
Front End	:	Python
Back End	:	MySQL

6. SYSTEM DESIGN

6.1 SYSTEM ARCHITECTURE

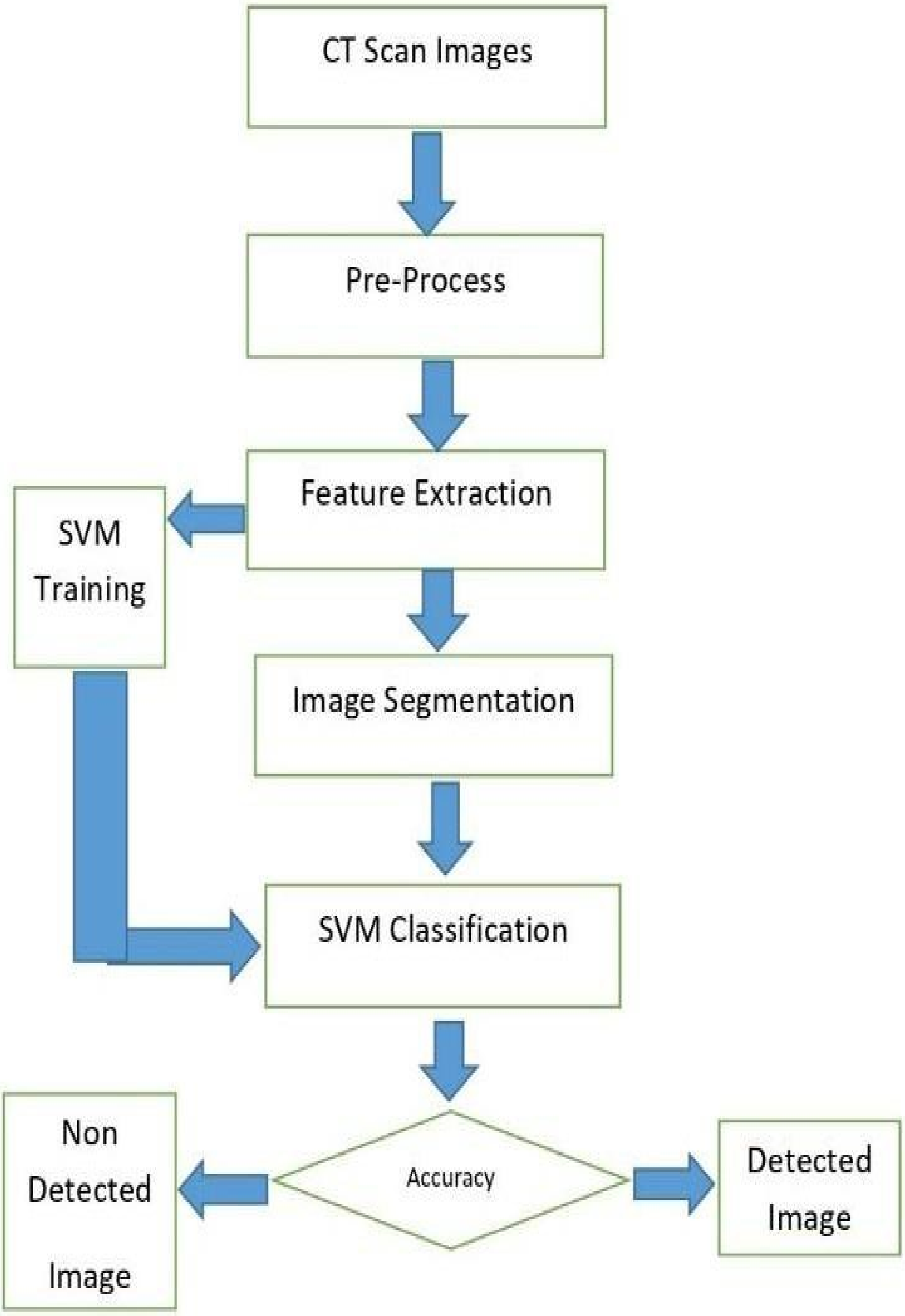


Figure 6.1.1 - Support Vector Machine Implementation in System

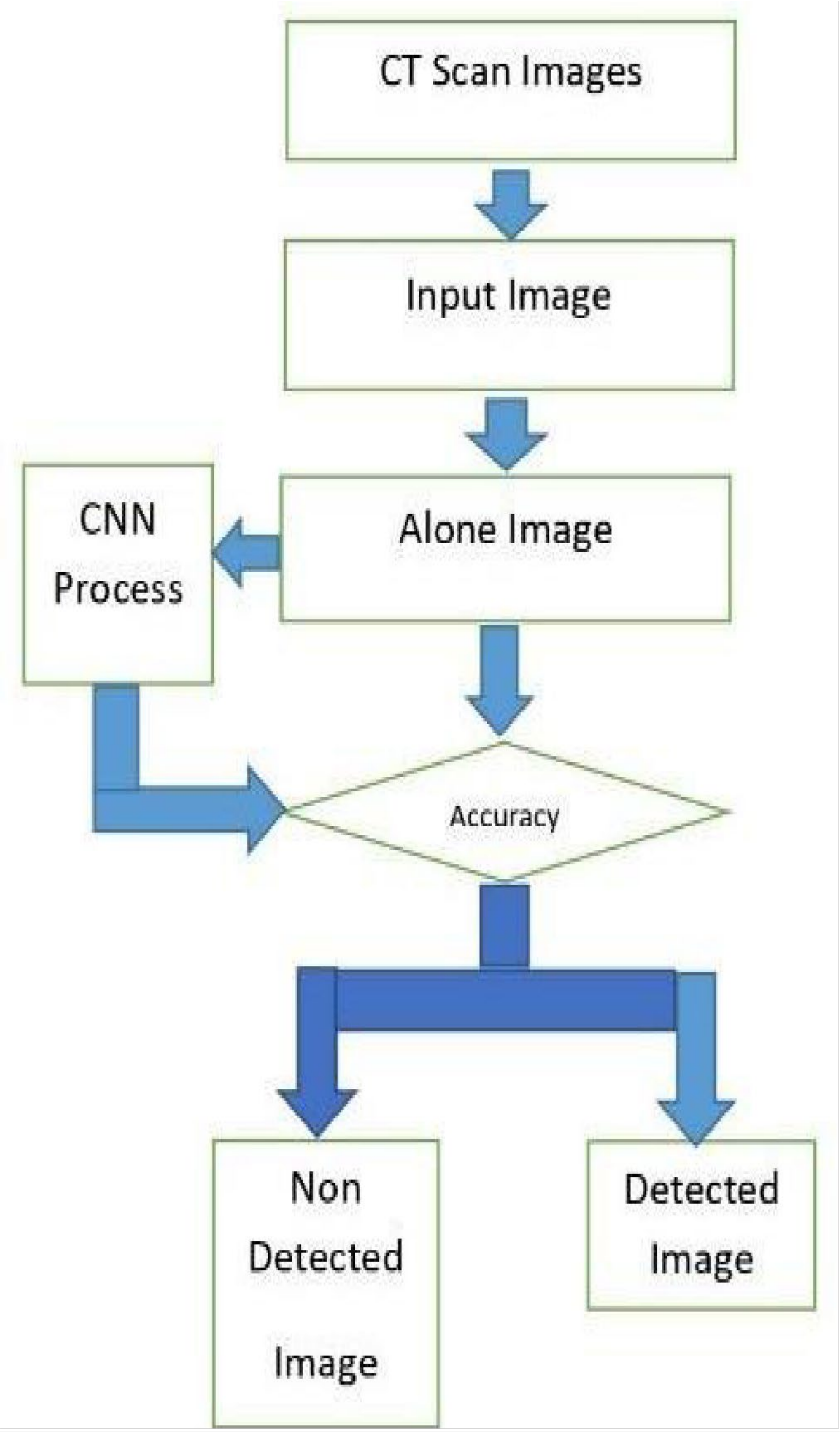


Figure 6.1.2- Convolutional Neural Network Implementation in System

6.2 DATA FLOW DIAGRAM

- The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
- The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
- DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
- DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

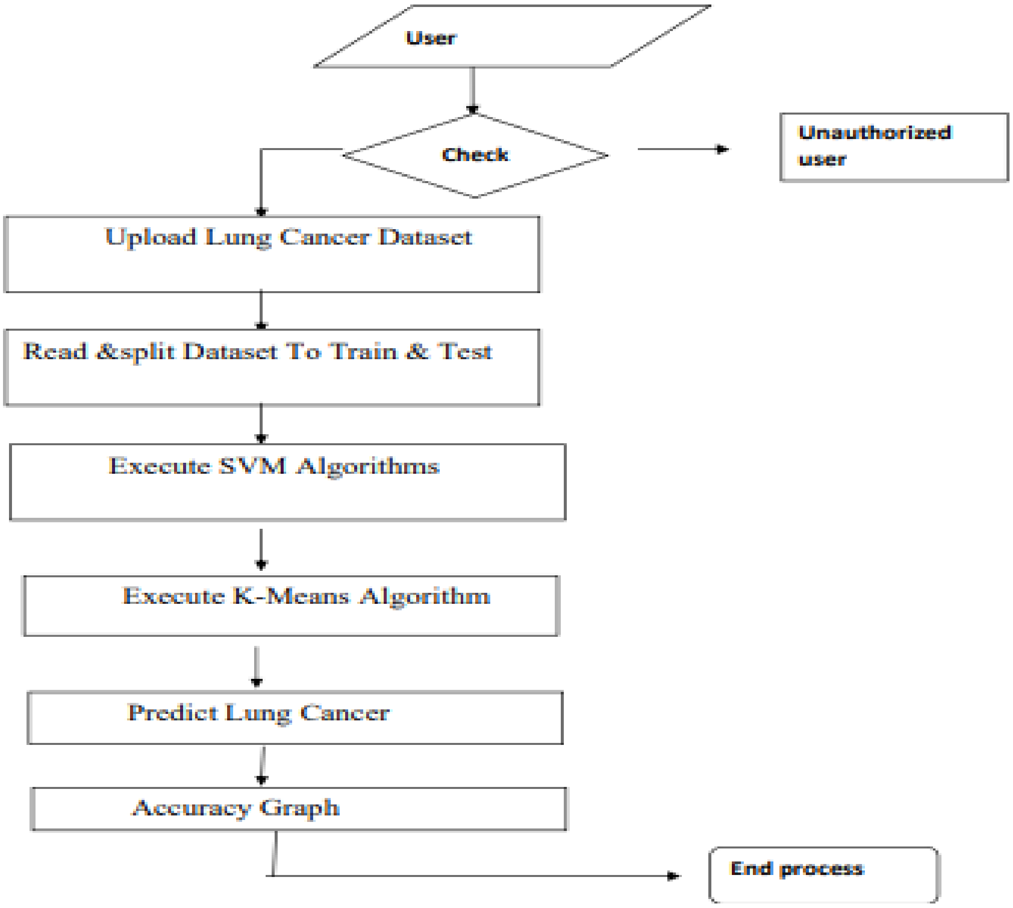


Figure 6.2.1- Data Flow Diagram

6.3 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

6.4 GOALS

The Primary goals in the design of the UML are as follows:

- Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
- Provide extendibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development process.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of OO tools market.
- Support higher level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices.

6.5 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

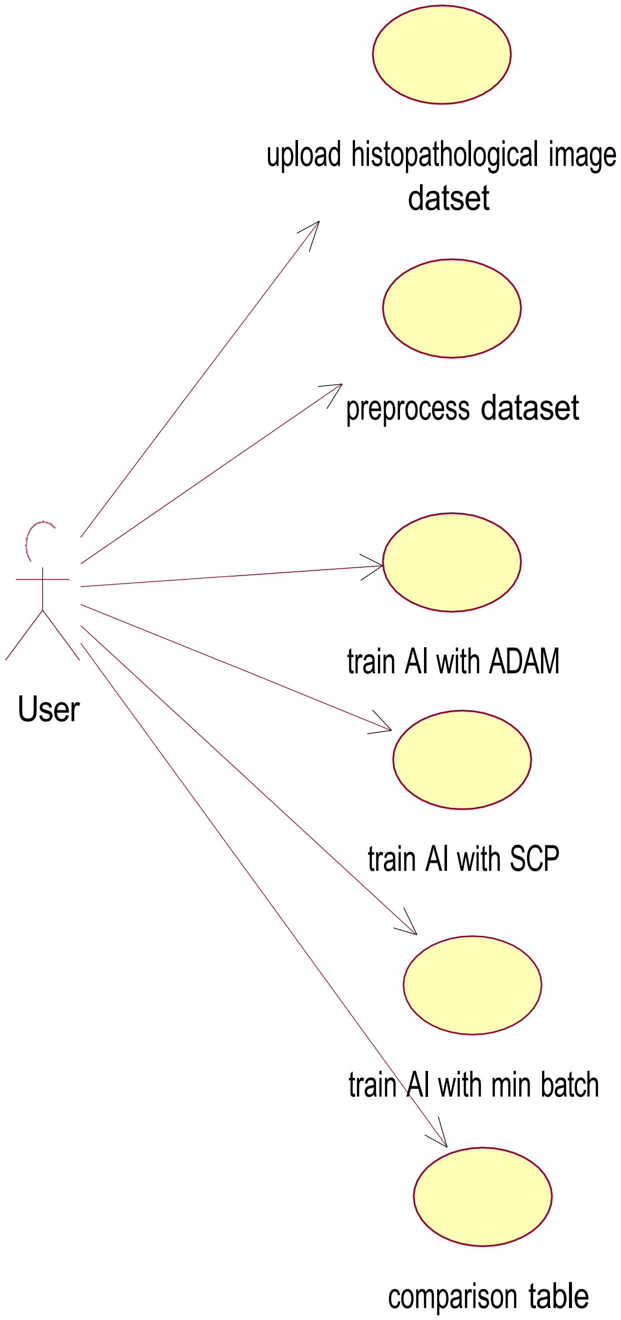


Figure 6.5.1 – Use Case Diagram

6.6 CLASS DIAGRAM

In software engineering, 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 the classes. It explains which class contains information.

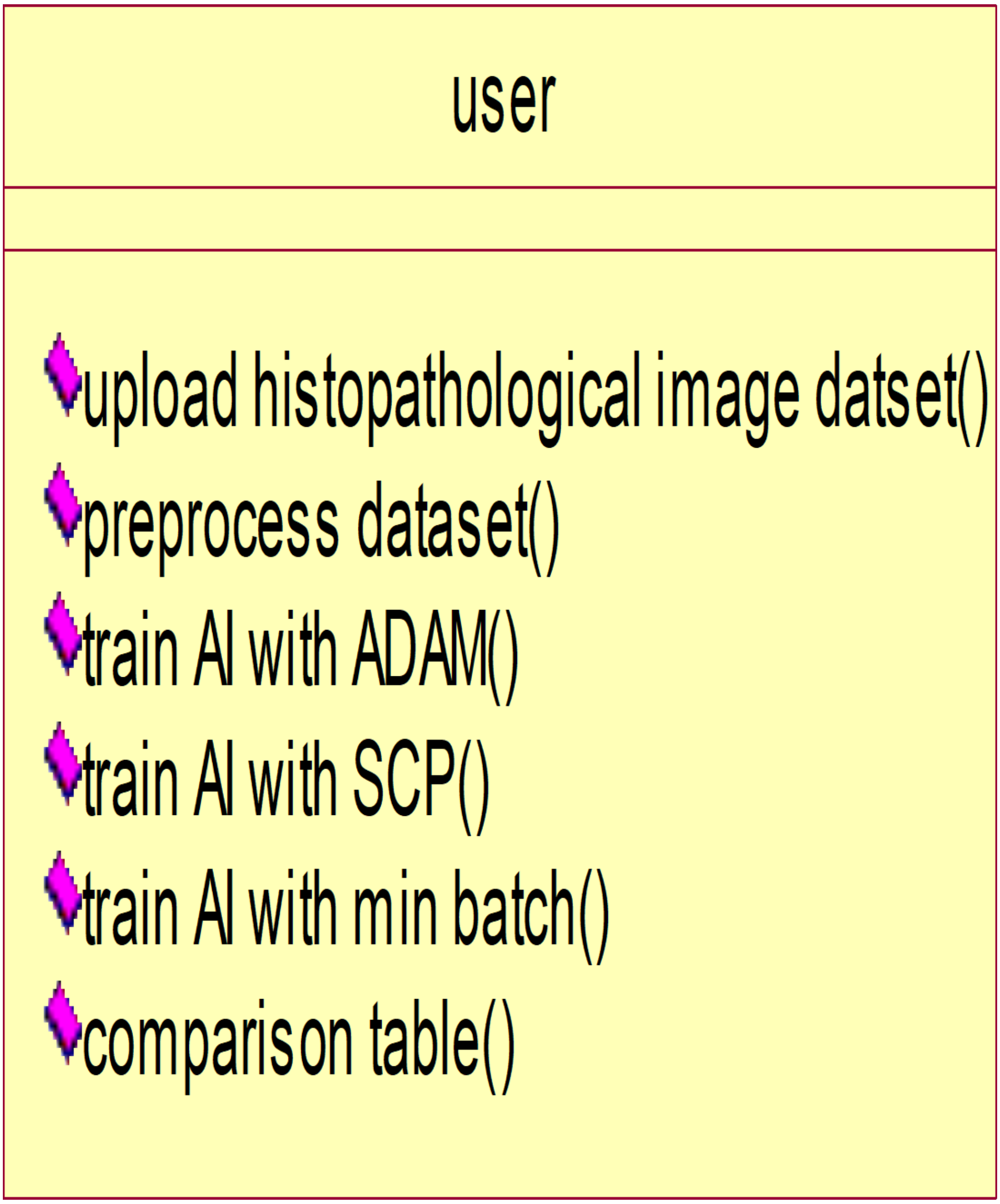


Figure 6.6.1 – Class Diagram

6.7 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

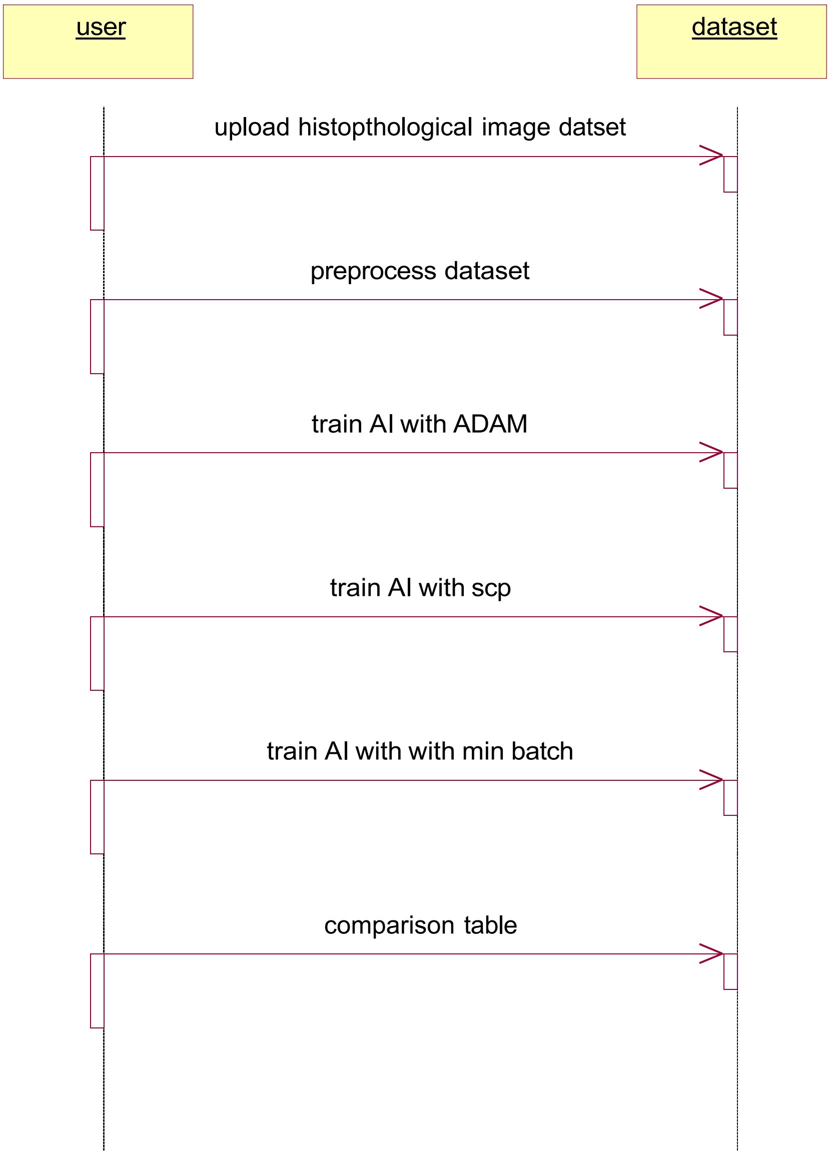


Figure 6.7.1 – Sequence Diagram

6.8 COLLABRATION DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

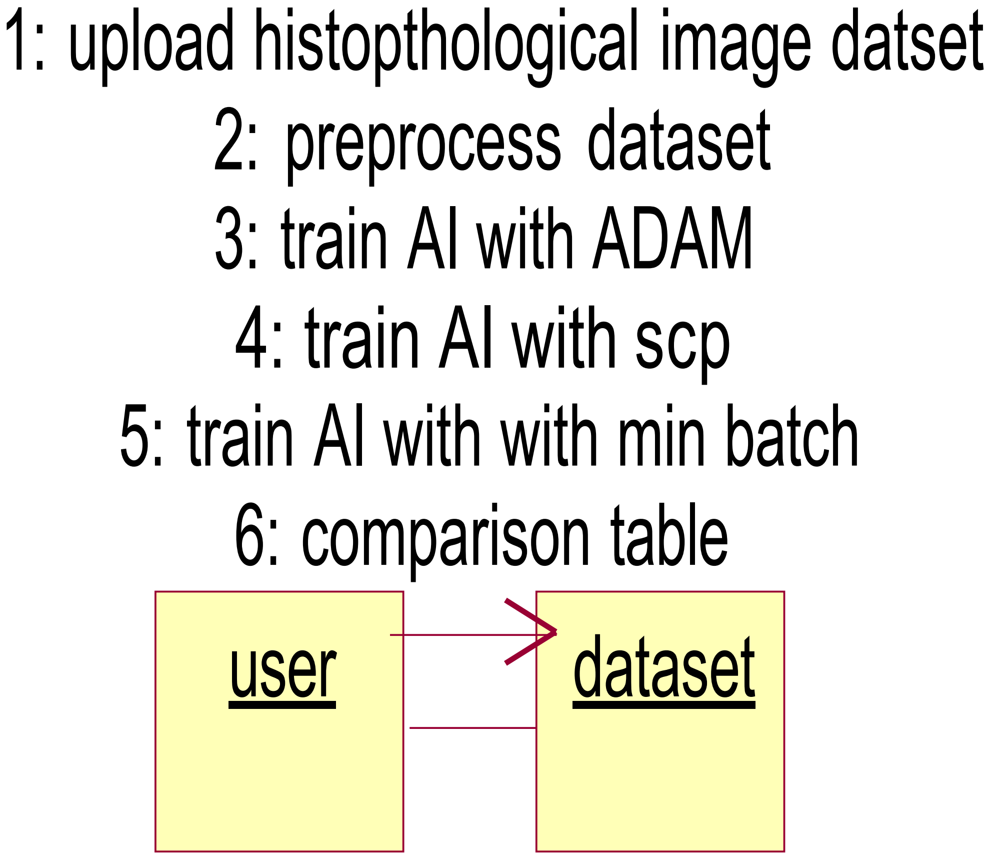


Figure 6.8.1 – Collaboration Diagram

7. IMPLEMENTATION

7.1 MODULES

- Upload MRI images dataset .
- Read &spilt images dataset to train & test model.
- Execute SVM Algorithms
- Execute K-Means Algorithm
- Execute KNN Algorithm
- Execute CNN Algorithm
- Predict Lung Cancer
- Accuracy Graph
- Error Rate Graph
- Survival Rate Graph.

MODULES DESCRIPTION:

1. Upload MRI images dataset:

Use this module to then upload dataset folder.

2. Read &spilt images dataset to train & test model :

Read & Split Dataset to Train & Test' button to split dataset into train and test parts and application split 80% dataset for training and 20% dataset to test trained model.

3. Execute SVM Algorithms :

Execute SVM Algorithm' button to run SVM on loaded dataset and to get Accuracy.

4. Execute K-Means Algorithm :

Execute K-Means Algorithm" button to run KMEANS algorithm on loaded dataset.

5. Execute KNN Algorithm :

Execute KNN Algorithm button to run KNN Algorithm on loaded dataset.

6. Execute CNN Algorithm:

Execute CNN Algorithm button to run CNN Algorithm on loaded dataset. . And compare survival rate of patients.

7. Predict Lung Cancer:

Predict Lung Cancer' button to upload new test image and then application will give prediction result

8. Accuracy Graph :

Accuracy Graph button to execute accuracy graph button to get the accuracy of algorithm

9. Error Rate Graph:

Error Rate Graph button to get accuracy of error rate algorithm

10. Survival Rate Graph:

Survival Rate Graph button to execute to get accuracy of survival rate algorithm

7.2 SOURCE CODE

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
import matplotlib.pyplot as plt
import numpy as np
from tkinter.filedialog import askopenfilename
import pandas as pd
import os
import cv2
import numpy as np
from sklearn import svm
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.decomposition import PCA
from keras.utils.np_utils import to_categorical
from keras.layers import  MaxPooling2D
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Convolution2D
from keras.models import Sequential
```

```
main = tkinter.Tk()
main.title("Detection of Lung cancer from CT image using SVM classification and
compare the survival rate of patients using 3D Convolutional neural network(3D
CNN)on lung nodules data set")
main.geometry("1300x1200")

global filename
global classifier
global svm_sr, cnn_sr
global X, Y
global X_train, X_test, y_train, y_test
global pca

def uploadDataset():
    global filename
    filename = filedialog.askdirectory(initialdir=".")
    text.delete('1.0', END)
    text.insert(END,filename+" loaded\n");



def splitDataset():
    global X, Y
    global X_train, X_test, y_train, y_test
    global pca
    text.delete('1.0', END)
    X = np.load('features/X.txt.npy')
    Y = np.load('features/Y.txt.npy')
    X = np.reshape(X, (X.shape[0],(X.shape[1]*X.shape[2]*X.shape[3])))

    pca = PCA(n_components = 100)
    X = pca.fit_transform(X)
    print(X.shape)
    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)
    text.insert(END,"Total CT Scan Images Found in dataset : "+str(len(X))+"\n")
    text.insert(END,"Train split dataset to 80% : "+str(len(X_train))+"\n")
    text.insert(END,"Test split dataset to 20%  : "+str(len(X_test))+"\n")



def executeSVM():
    global classifier
    global svm_sr
    text.delete('1.0', END)
    cls = svm.SVC()
    cls.fit(X_train, y_train)
    predict = cls.predict(X_test)
    svm_sr = accuracy_score(y_test,predict) * 100
    classifier = cls
    text.insert(END,"SVM Survival Rate : "+str(svm_sr)+"\n")

def executeCNN():
    global cnn_sr
    X = np.load('features/X.txt.npy')
    Y = np.load('features/Y.txt.npy')
    Y = to_categorical(Y)
    classifier = Sequential()
    classifier.add(Convolution2D(32, 3, 3, input_shape = (64, 64, 3), activation = 
'relu'))
    classifier.add(MaxPooling2D(pool_size = (2, 2)))
    classifier.add(Convolution2D(32, 3, 3, activation = 'relu'))
    classifier.add(MaxPooling2D(pool_size = (2, 2)))
    classifier.add(Flatten())
    classifier.add(Dense(output_dim = 256, activation = 'relu'))
    classifier.add(Dense(output_dim = 2, activation = 'softmax'))
    print(classifier.summary())
    classifier.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = 
['accuracy'])
    hist = classifier.fit(X, Y, batch_size=16, epochs=10, shuffle=True, verbose=2)
    hist = hist.history
    acc = hist['accuracy']
    cnn_sr = acc[9] * 100
    text.insert(END,"CNN Survival Rate : "+str(cnn_sr)+"\n")


def predictCancer():
    filename = filedialog.askopenfilename(initialdir="testSamples")
    img = cv2.imread(filename)
    img = cv2.resize(img, (64,64))
    im2arr = np.array(img)
    im2arr = im2arr.reshape(64,64,3)
    im2arr = im2arr.astype('float32')
    im2arr = im2arr/255
    test = []
    test.append(im2arr)
    test = np.asarray(test)
    test = np.reshape(test, (test.shape[0],(test.shape[1]*test.shape[2]*test.shape[3])))
    test = pca.transform(test)
    predict = classifier.predict(test)[0]
    msg = ''
    if predict == 0:
        msg = "Uploaded CT Scan is Normal"
    if predict == 1:
        msg = "Uploaded CT Scan is Abnormal"
    img = cv2.imread(filename)
    img = cv2.resize(img, (400,400))
    cv2.putText(img, msg, (10, 25),  cv2.FONT_HERSHEY_SIMPLEX,0.7, (0, 255,
255), 2)
    cv2.imshow(msg, img)
    cv2.waitKey(0)

def graph():
    height = [svm_sr, cnn_sr]
    bars = ('SVM Survival Rate','CNN Survival Rate')
    y_pos = np.arange(len(bars))
    plt.bar(y_pos, height)
    plt.xticks(y_pos, bars)
    plt.show()

font = ('times', 14, 'bold')
title = Label(main, text='Detection of Lung cancer from CT image using SVM
classification and compare the survival rate of patients using 3D Convolutional neural
network(3D CNN)on lung nodules data set')
title.config(bg='deep sky blue', fg='white')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)

font1 = ('times', 12, 'bold')
text=Text(main,height=20,width=150)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=50,y=120)
text.config(font=font1)



font1 = ('times', 13, 'bold')
uploadButton = Button(main, text="Upload Lung Cancer Dataset",
command=uploadDataset)
uploadButton.place(x=50,y=550)
uploadButton.config(font=font1)

readButton = Button(main, text="Read & Split Dataset to Train & Test",
command=splitDataset)
readButton.place(x=350,y=550)
readButton.config(font=font1)
```

```
svmButton = Button(main, text="Execute SVM Algorithms",
command=executeSVM)
svmButton.place(x=50,y=600)
svmButton.config(font=font1)

kmeansButton = Button(main, text="Execute CNN Algorithm",
command=executeCNN)
kmeansButton.place(x=350,y=600)
kmeansButton.config(font=font1)

predictButton = Button(main, text="Predict Lung Cancer", command=predictCancer)
predictButton.place(x=50,y=650)
predictButton.config(font=font1)

graphButton = Button(main, text="Survival Rate Graph", command=graph)
graphButton.place(x=350,y=650)
graphButton.config(font=font1)

main.config(bg='LightSteelBlue3')
main.mainloop()
```

8. LIBRARIES USED IN PROJECT

8.1 TensorFlow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

8.2 Numpy

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

8.3 Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

8.4 Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the python module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

8.5 Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. **.**

9. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

9.1 TYPES OF TESTS

9.1.1 Unit testing

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

9.1.2 Integration testing

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

9.1.3 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- Valid Input : identified classes of valid input must be accepted.
- Invalid Input : identified classes of invalid input must be rejected.
- Functions : identified functions must be exercised.
- Output : identified classes of application outputs must be exercised.
- Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

9.1.4 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

9.1.5 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

9.1.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

9.1.7 Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

9.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

9.3 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

10. INPUT DESIGN AND OUTPUT DESIGN

10.1 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

- Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
- It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
- When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

10.2 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

- Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
- Select methods for presenting information.
- Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

11. SCREENSHOTS

11.1 CLASSIFICATION OF LUNG CANCER NODULES TO MONITOR PATIENTS HEALTH USING NEURAL NETWORK TOPOLOGY WITH SVM ALGORITHM & COMPARE WITH K-MEANS ACCURACY

11.1.1 CT SCAN LUNG CANCER NODULES DATASET

In this project we are using CT Scan Lung Cancer Nodules dataset to predict patient health using SVM and K-Means algorithm and then comparing prediction accuracy between them. To implement this project we are using lung cancer images dataset and below screen showing dataset details and this dataset saved inside 'Dataset' folder.

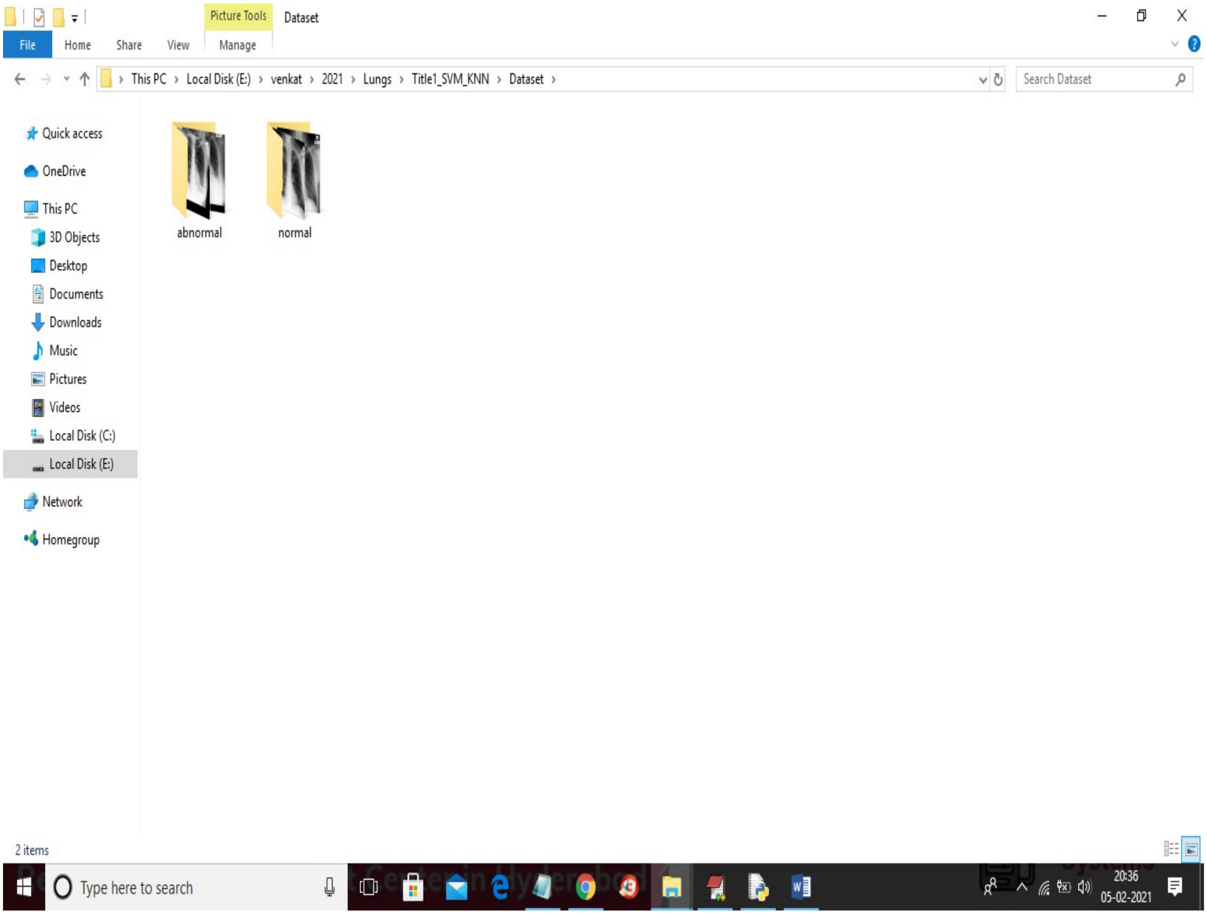
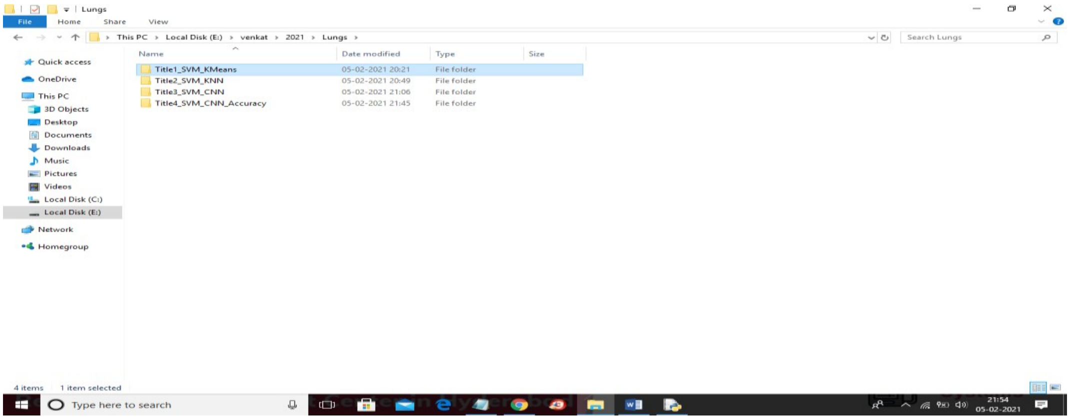


Figure 11.1.1 – Dataset In a form of two types of images normal and abnormal

In above screen in dataset we have two types of images such as normal and abnormal and then SVM and KMEANS will get train on above dataset and when we upload new image then SVM will predict whether new image is normal or abnormal. To implement 4 titles we created 4 folders to separate algorithms for each title and you can run one by one.

11.1.2 UPLOAD DATASET

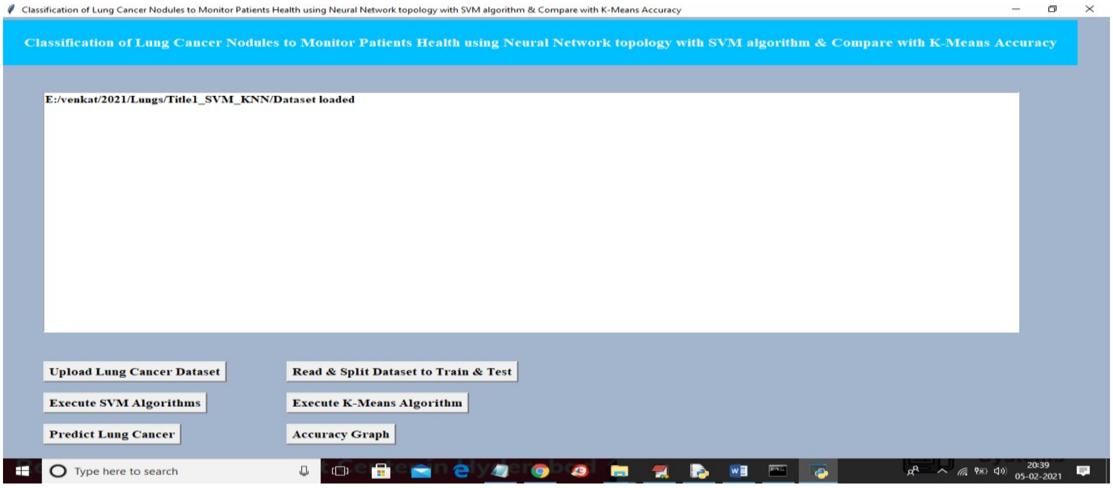


Name	Date modified	Type	Size
Title1_SVM_KMeans	05-02-2021 20:21	File folder	
Title2_SVM_KNN	05-02-2021 20:49	File folder	
Title3_SVM_CNN	05-02-2021 21:06	File folder	
Title4_SVM_CNN_Accuracy	05-02-2021 21:45	File folder	

Figure 11.1.2.– Lungs Based Files to run the project

In above fig 11.1.2 we can upload the selected lung cancer dataset whichcontains two sub-folders ‘Abnormal’ and ‘Normal’ from the user directory. In above screen by selecting ‘Upload Lung Cancer Dataset’ button we upload the dataset from the ‘Dataset’ folder and then click on ‘Select Folder’ button to load dataset and to get below screen

11.1.3 READ &SPILT DATASET TO TRAIN &TEST

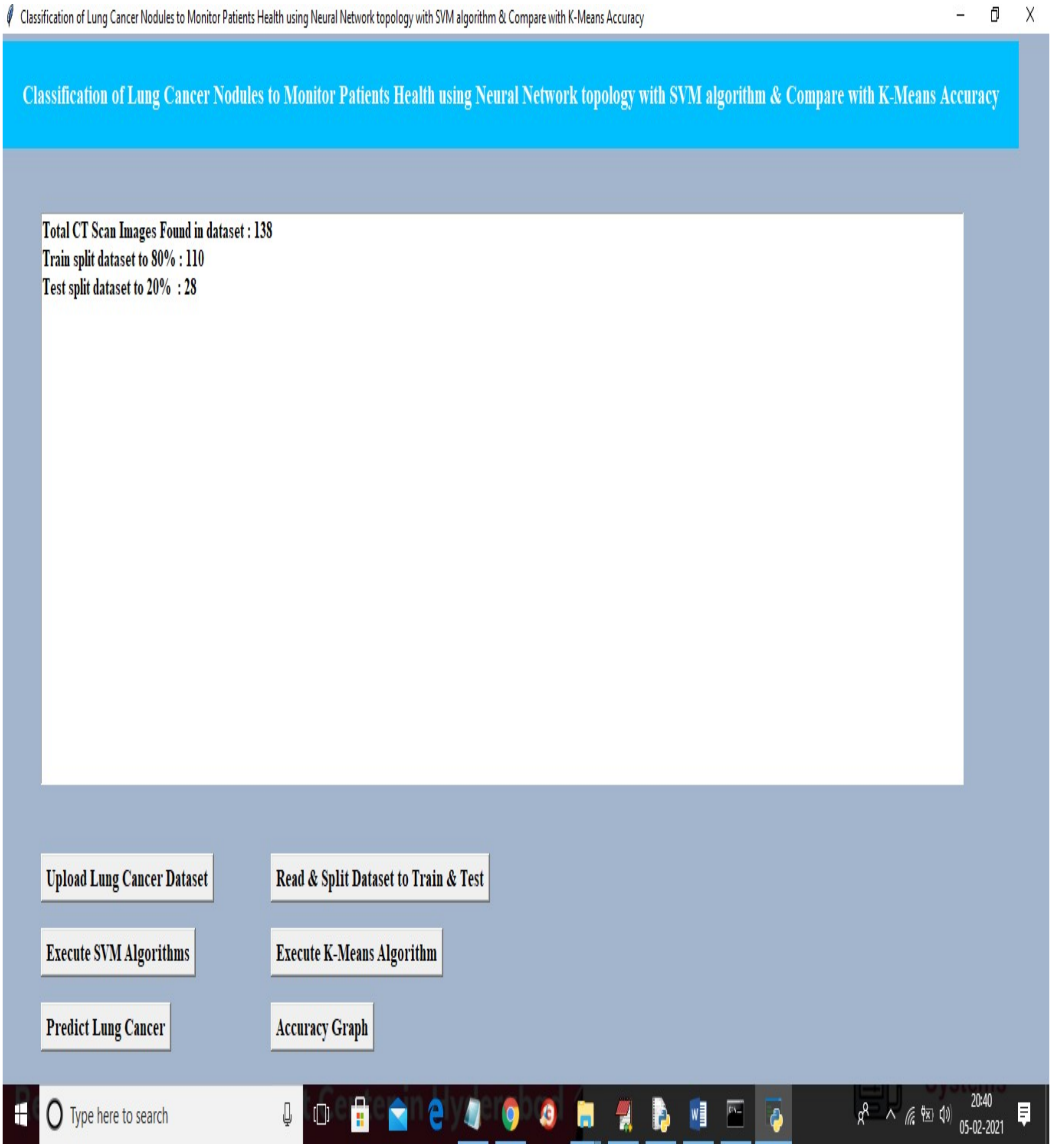

Classification of Lung Cancer Nodules to Monitor Patients Health using Neural Network topology with SVM algorithm & Compare with K-Means Accuracy

Classification of Lung Cancer Nodules to Monitor Patients Health using Neural Network topology with SVM algorithm & Compare with K-Means Accuracy

E:/venkat/2021/Lungs/Title1_SVM_KNN/Dataset loaded

Figure 11.1.3 – Read and Split Dataset to train and test

In above screen dataset loaded and now click on ‘Read & Split Dataset to Train & Test’ button to split dataset into train and test parts and application split 80% dataset for training and 20% dataset to test trained model



Classification of Lung Cancer Nodules to Monitor Patients Health using Neural Network topology with SVM algorithm & Compare with K-Means Accuracy

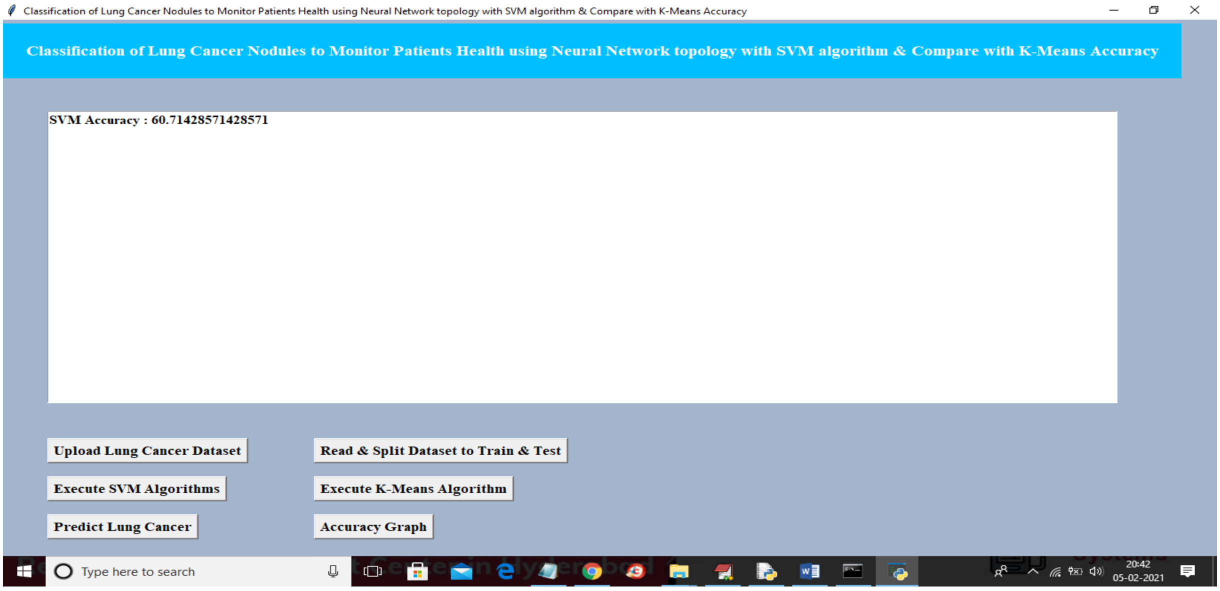
Classification of Lung Cancer Nodules to Monitor Patients Health using Neural Network topology with SVM algorithm & Compare with K-Means Accuracy

```
Total CT Scan Images Found in dataset : 138
Train split dataset to 80% : 110
Test split dataset to 20%  : 28
```

Figure 11.1.4 – Data Set shows no of images using training and test data

In above screen we can see dataset contains total 138 images and then application using 110 images for training and 28 images for testing and now data is ready and now click on ‘Execute SVM Algorithm’ button to run SVM on loaded dataset and to get below accuracy

11.1.4 EXECUTION OF SVM ALGORITHM

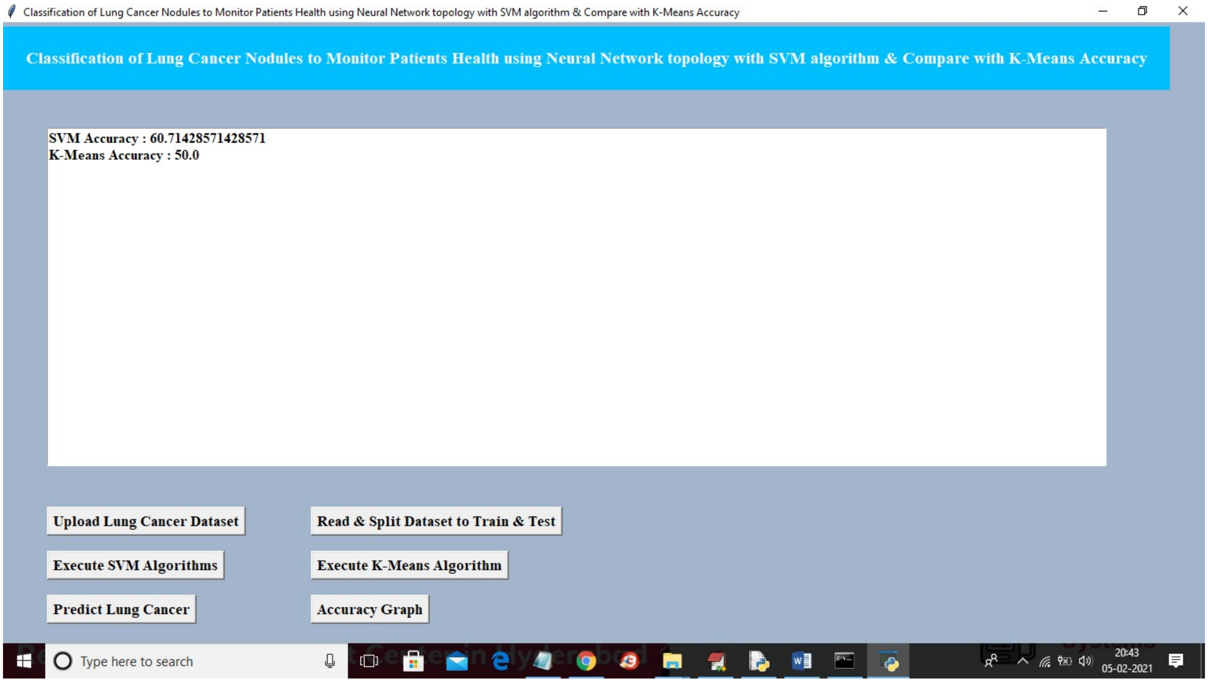

Classification of Lung Cancer Nodules to Monitor Patients Health using Neural Network topology with SVM algorithm & Compare with K-Means Accuracy
Classification of Lung Cancer Nodules to Monitor Patients Health using Neural Network topology with SVM algorithm & Compare with K-Means Accuracy
Upload Lung Cancer Dataset
Read & Split Dataset to Train & Test
Execute SVM Algorithms
Execute K-Means Algorithm
Predict Lung Cancer
Accuracy Graph
Type here to search
20:42
05-02-2021

```
SVM Accuracy : 60.71428571428571
```

Figure 11.1.5 – Execute K-Means Algorithm

In above screen SVM accuracy is 60% and now click on "Execute K-Means Algorithm" button to run KMEANS algorithm on loaded dataset and to get below screen

11.1.5 K-MEANS EXECUTION

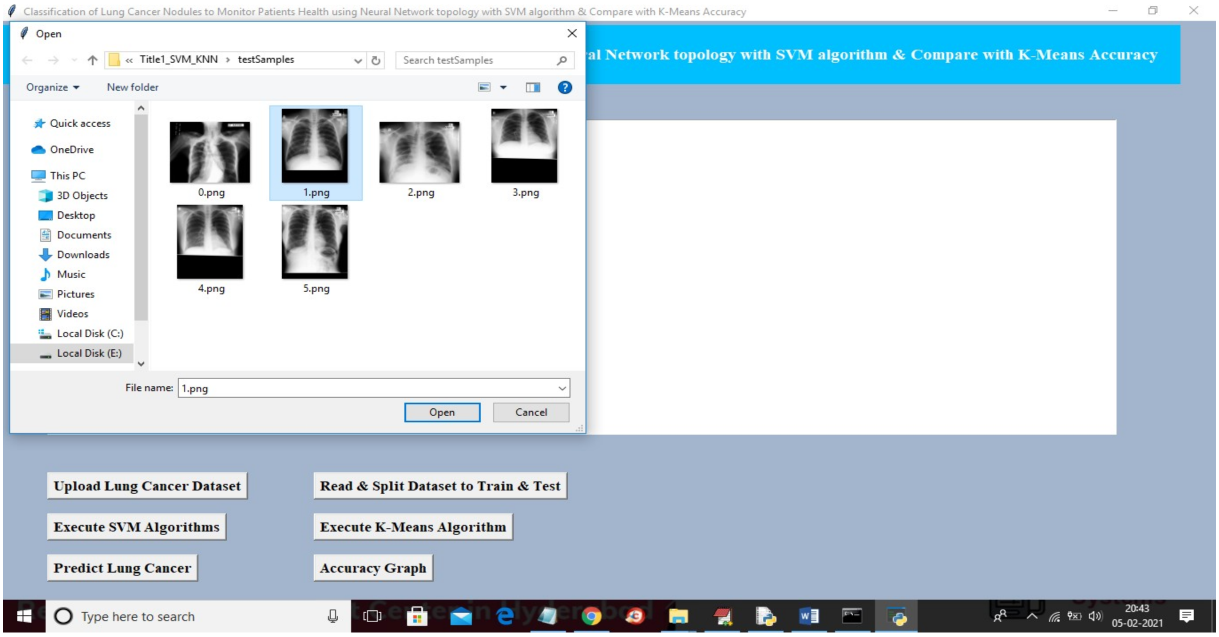

Classification of Lung Cancer Nodules to Monitor Patients Health using Neural Network topology with SVM algorithm & Compare with K-Means Accuracy
Classification of Lung Cancer Nodules to Monitor Patients Health using Neural Network topology with SVM algorithm & Compare with K-Means Accuracy
Upload Lung Cancer Dataset
Read & Split Dataset to Train & Test
Execute SVM Algorithms
Execute K-Means Algorithm
Predict Lung Cancer
Accuracy Graph
Type here to search
20:43
05-02-2021

```
SVM Accuracy : 60.71428571428571
K-Means Accuracy : 50.0
```

Figure 11.1.6 – Predicting Lung Cancer using K-Means

In above screen KMEANS got 50% accuracy and now click on 'Predict Lung Cancer' button to upload new test image and then application will give prediction result

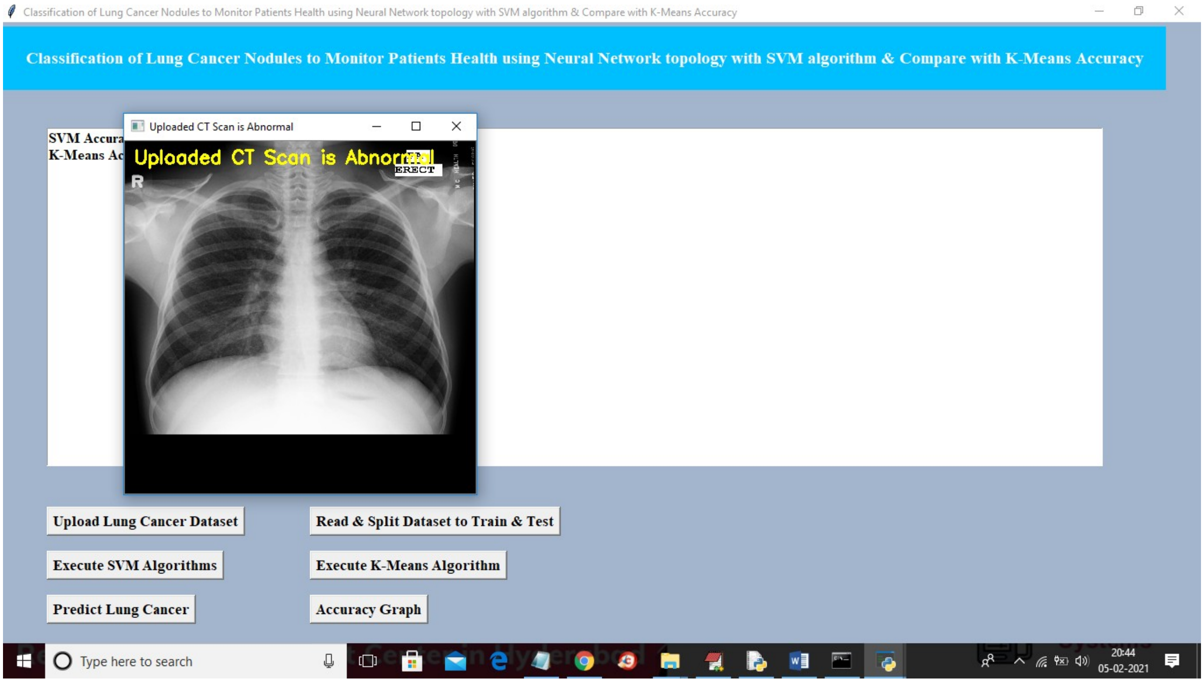
11.1.6 PREDICTING LUNG CANCER


Classification of Lung Cancer Nodules to Monitor Patients Health using Neural Network topology with SVM algorithm & Compare with K-Means Accuracy
al Network topology with SVM algorithm & Compare with K-Means Accuracy
Open
« Title1_SVM_KNN › testSamples
Search testSamples
Organize ▾
New folder
Quick access
OneDrive
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Local Disk (E:)
0.png
1.png
2.png
3.png
4.png
5.png
Open
Cancel
Upload Lung Cancer Dataset
Read & Split Dataset to Train & Test
Execute SVM Algorithms
Execute K-Means Algorithm
Predict Lung Cancer
Accuracy Graph
Type here to search
20:43
05-02-2021

File name: 1.png

Figure 11.1.7 – Using K-Means To prdict the output

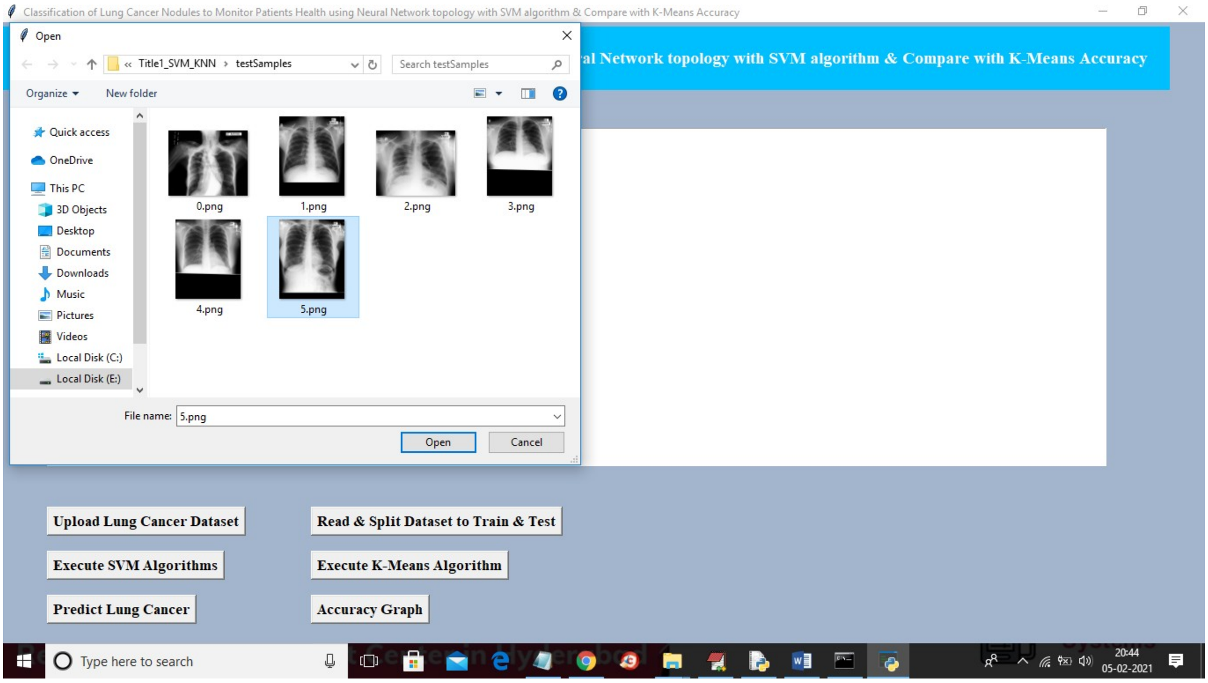
In above screen selecting and uploading '1.png' file and then click on 'Open' button to get below result


Classification of Lung Cancer Nodules to Monitor Patients Health using Neural Network topology with SVM algorithm & Compare with K-Means Accuracy
Classification of Lung Cancer Nodules to Monitor Patients Health using Neural Network topology with SVM algorithm & Compare with K-Means Accuracy
SVM Accura
K-Means Ac
Uploaded CT Scan is Abnormal
ERECT
R
Upload Lung Cancer Dataset
Read & Split Dataset to Train & Test
Execute SVM Algorithms
Execute K-Means Algorithm
Predict Lung Cancer
Accuracy Graph
Type here to search
20:44
05-02-2021

Uploaded CT Scan is Abnormal

Figure 11.1.8– Image Shows as Abnormal Condition

In above screen uploaded image predicted as Abnormal and now test with another image

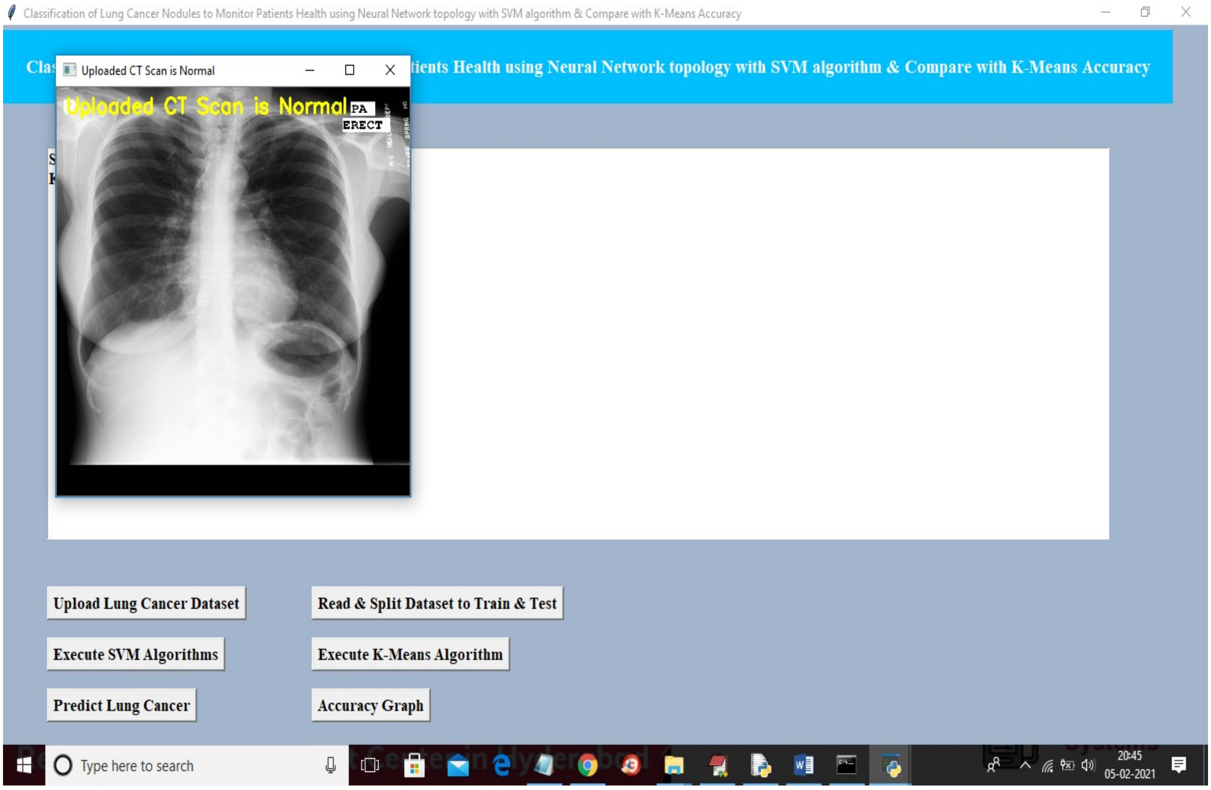


Classification of Lung Cancer Nodules to Monitor Patients Health using Neural Network topology with SVM algorithm & Compare with K-Means Accuracy

File name: 5.png

Figure 11.1.9 – Selecting another image

In above screen uploading ‘5.png’ and below is the result



Classification of Lung Cancer Nodules to Monitor Patients Health using Neural Network topology with SVM algorithm & Compare with K-Means Accuracy

Uploaded CT Scan is Normal

Uploaded CT Scan is Normal

PA
ERECT
Cla
tients Health using Neural Network topology with SVM algorithm & Compare with K-Means Accuracy
Upload Lung Cancer Dataset
Read & Split Dataset to Train & Test
Execute SVM Algorithms
Execute K-Means Algorithm
Predict Lung Cancer
Accuracy Graph
Type here to search
20:45
05-02-2021

Figure 11.1.10 – Image Shows Normal Condition

Above image predicted as Normal and similarly you can upload any image and perform prediction and now click on ‘Accuracy Graph’ button to get below graph

11.1.7 COMPARSION OF SVM AND K-MEANS

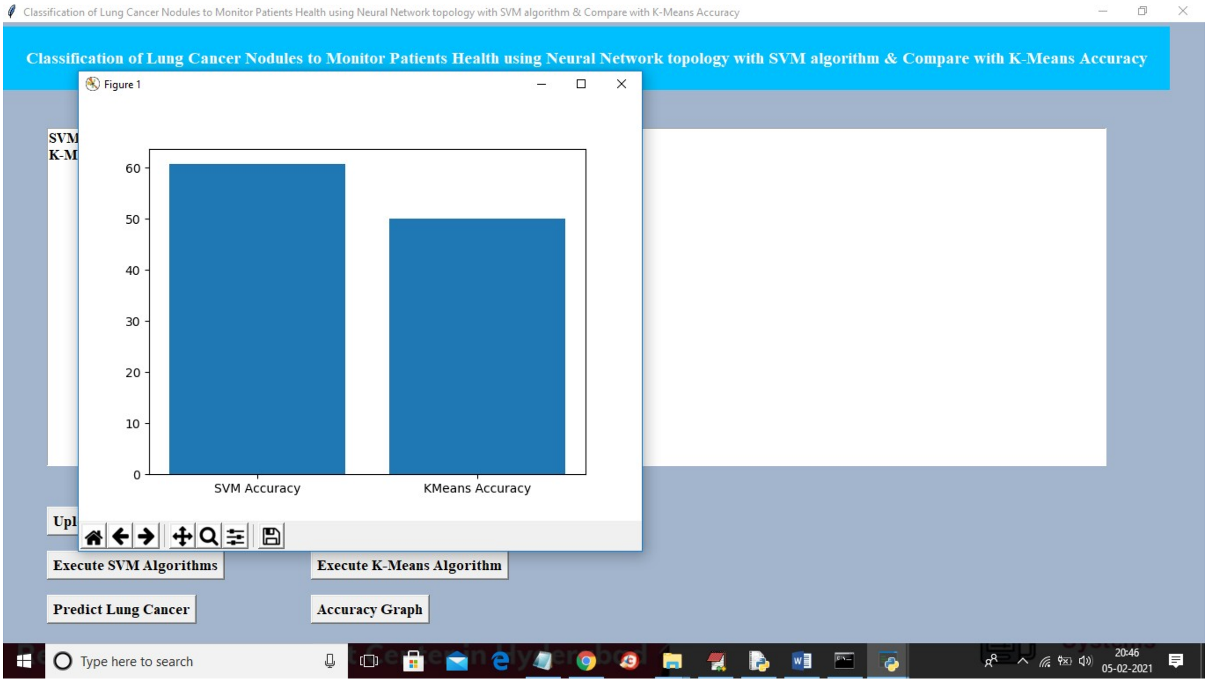

Classification of Lung Cancer Nodules to Monitor Patients Health using Neural Network topology with SVM algorithm & Compare with K-Means Accuracy
Classification of Lung Cancer Nodules to Monitor Patients Health using Neural Network topology with SVM algorithm & Compare with K-Means Accuracy
Figure 1
60
50
40
30
20
10
0
SVM Accuracy
KMeans Accuracy
Upl
Execute SVM Algorithms
Execute K-Means Algorithm
Predict Lung Cancer
Accuracy Graph
Type here to search
20:46
05-02-2021

Figure 11.1.11- Shows Accuracy level and Comparison of SVM and K-Means

In above screen x-axis represents algorithm name and y-axis represents accuracy of those algorithms and from above graph we can conclude that SVM is better than KMEANS in prediction.

11.2 A HYBRID MODEL CLASSIFICATION OF LUNG NODULES ACROSS SEQUENTIAL AND TIMEVARIANTDATA TO ASSIST DOCTORS TO TREAT THEDISEASEOVER SVM ALGORITHM AND COMPARE THEERRORRATE WITH K-NEAREST NEIGHBOURS

In this project we are using same above dataset to train SVM and KNN algorithm and then calculating error rate between these two algorithms and this error rate refers to wrong classification percentage. For example, if application predicted 18 records correctly out of 20 records then error rate will be (1 – (18/20) = 0.1.

11.2.1 EXCUTING SVM AND KNN ALGORITHM

To run project double click on ‘run.bat’ file from ‘Title2_SVM_KNN’ folder to get below screen

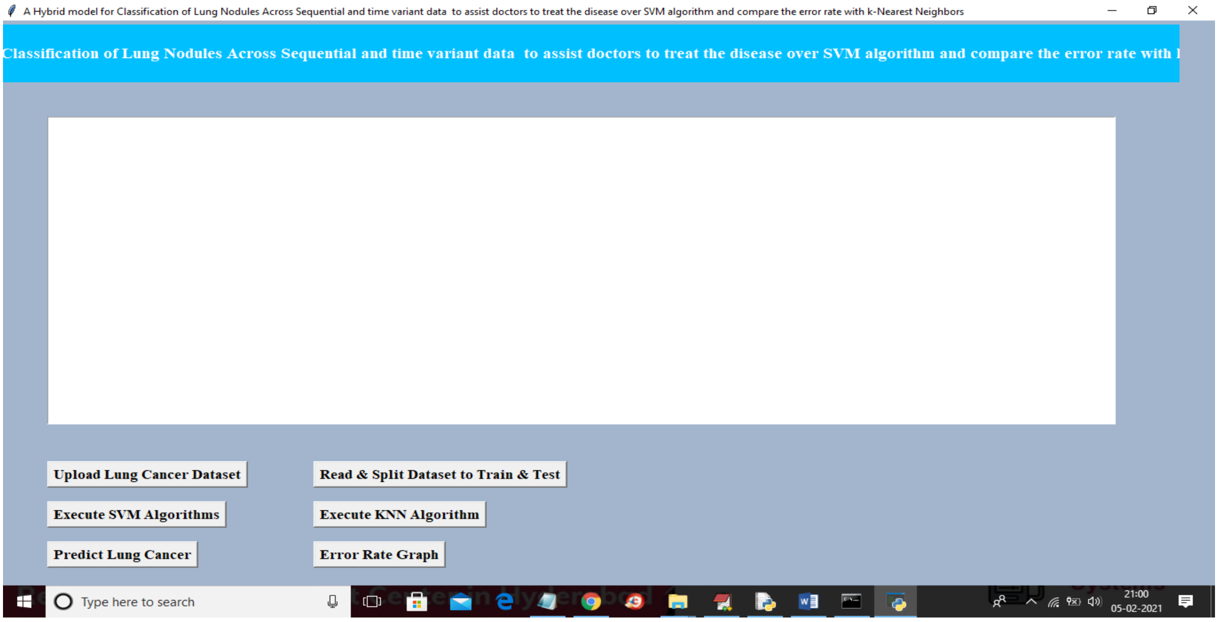
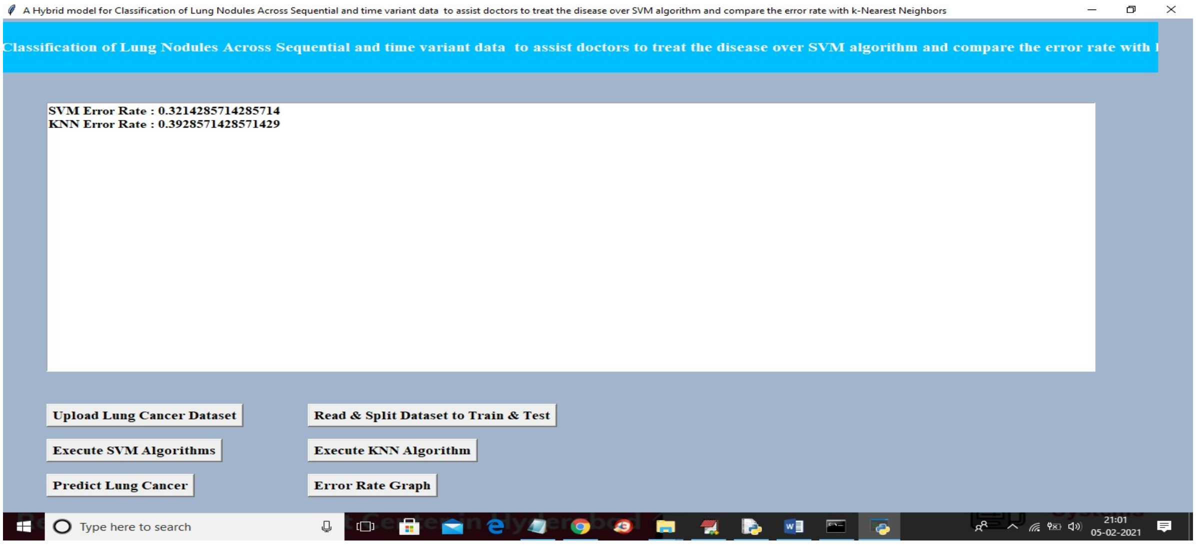

A Hybrid model for Classification of Lung Nodules Across Sequential and time variant data to assist doctors to treat the disease over SVM algorithm and compare the error rate with k-Nearest Neighbors
Classification of Lung Nodules Across Sequential and time variant data to assist doctors to treat the disease over SVM algorithm and compare the error rate with l
Upload Lung Cancer Dataset
Read & Split Dataset to Train & Test
Execute SVM Algorithms
Execute KNN Algorithm
Predict Lung Cancer
Error Rate Graph
Type here to search
21:00
05-02-2021

Figure 11.2.1 – DataSet regarding SVM AND KNN

In above screen upload dataset and then read dataset and then execute SVM and KNN and then you can predict and calculate error rate

11.2.2 CALCULATING ERROR RATE OF SVM AND KNN ALGORITHM


A Hybrid model for Classification of Lung Nodules Across Sequential and time variant data to assist doctors to treat the disease over SVM algorithm and compare the error rate with k-Nearest Neighbors
Classification of Lung Nodules Across Sequential and time variant data to assist doctors to treat the disease over SVM algorithm and compare the error rate with l

```
SVM Error Rate : 0.3214285714285714
KNN Error Rate : 0.3928571428571429
```

Upload Lung Cancer Dataset
Read & Split Dataset to Train & Test
Execute SVM Algorithms
Execute KNN Algorithm
Predict Lung Cancer
Error Rate Graph
Type here to search
21:01
05-02-2021

Figure 11.2.2 – Shows Error Rate of SVM and KNN

In above screen we can see SVM error rate is 0.32% and KNN error rate is 0.39 and similarly like first project screen shots u can run prediction and graph button. In above screen datasets will be splitted randomly so for every run train and test data may change due to random selection so accuracy or error rate may vary

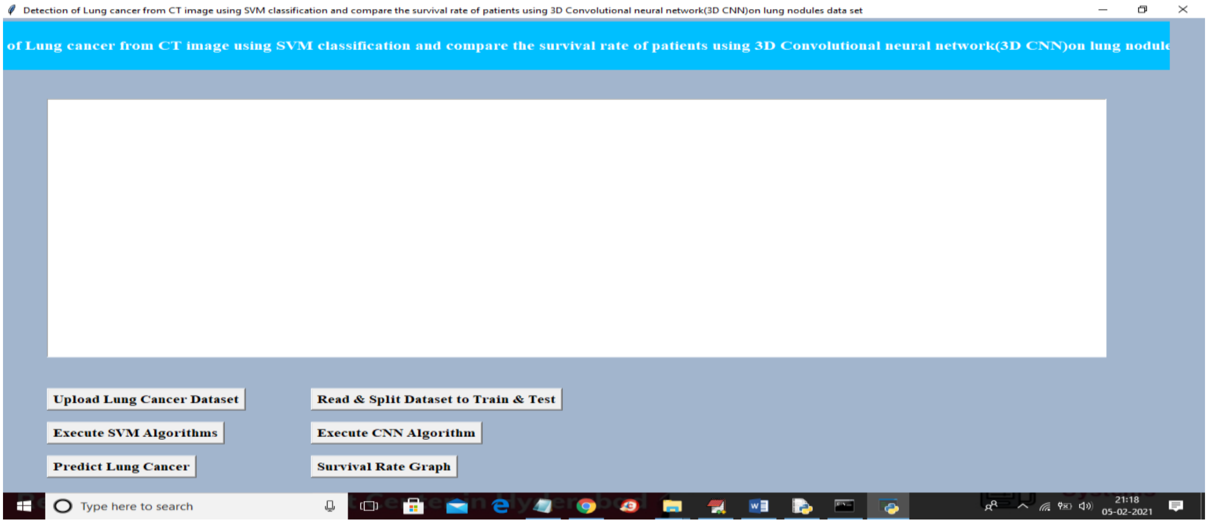
11.3 DETECTION OF LUNG CANCER FROM CT IMAGE USING SVM CLASSIFICATION AND COMPARE THE SURVIVAL RATE OF PATIENTS USING 3D CONVOLUTIONAL NEURAL NETWORK (3D CNN) ON LUNG NODULES DATA SET

Detection of Lung cancer from CT image using SVM classification and compare the survival rate of patients using 3D Convolutional neural network(3D CNN)on lung nodules data set

In this project we are using same above Lungs dataset to train CNN and SVM algorithm and then calculate survival rate of patients by using both algorithms. If algorithm predicted 18 records correctly out of 20 records then survival rate will be (18/20 * 100) = 90%.

11.3.1 EXECUTING SVM AND CNN ALGORITHM

To run project double click on ‘run.bat’ file from ‘Title3_SVM_CNN’ folder to get below screen



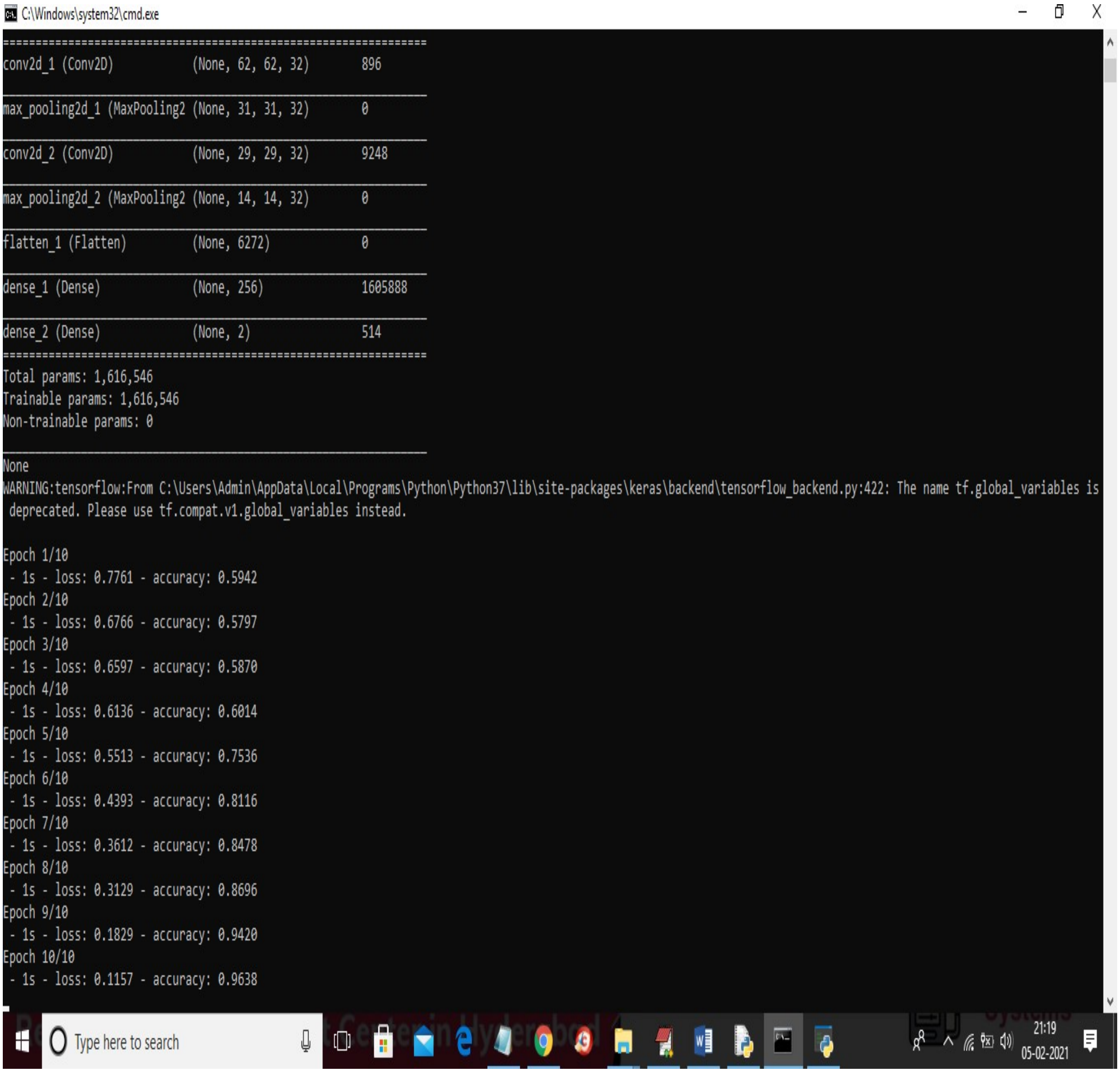
Detection of Lung cancer from CT image using SVM classification and compare the survival rate of patients using 3D Convolutional neural network(3D CNN)on lung nodules data set

of Lung cancer from CT image using SVM classification and compare the survival rate of patients using 3D Convolutional neural network(3D CNN)on lung nodule

Figure 11.3.1 – DataSet Regarding SVM and CNN

Similar to first two projects here also you upload ‘Dataset’ folder and then click on “read & split” button and then execute SVM and CNN and then predict cancer and go for survival rate graph. For CNN results you can refer to black console below

11.3.2 Execution of CNN algorithm



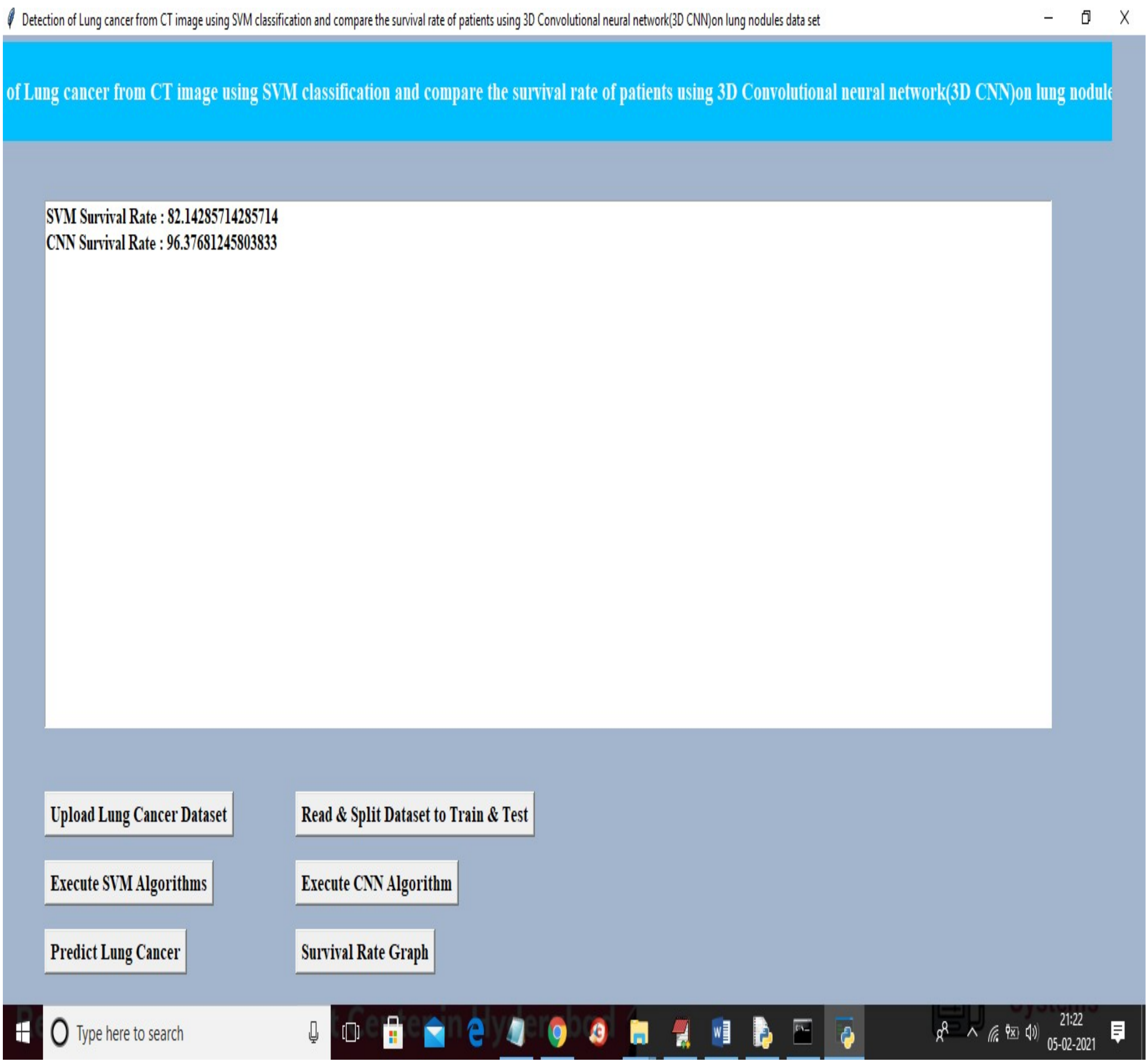
```
=================================================================
conv2d_1 (Conv2D)            (None, 62, 62, 32)        896
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 31, 31, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 29, 29, 32)        9248
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 14, 14, 32)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 6272)              0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               1605888
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 514
=================================================================
Total params: 1,616,546
Trainable params: 1,616,546
Non-trainable params: 0
_________________________________________________________________
None
WARNING:tensorflow:From C:\Users\Admin\AppData\Local\Programs\Python\Python37\lib\site-packages\keras\backend\tensorflow_backend.py:422: The name tf.global_variables is
 deprecated. Please use tf.compat.v1.global_variables instead.

Epoch 1/10
 - 1s - loss: 0.7761 - accuracy: 0.5942
Epoch 2/10
 - 1s - loss: 0.6766 - accuracy: 0.5797
Epoch 3/10
 - 1s - loss: 0.6597 - accuracy: 0.5870
Epoch 4/10
 - 1s - loss: 0.6136 - accuracy: 0.6014
Epoch 5/10
 - 1s - loss: 0.5513 - accuracy: 0.7536
Epoch 6/10
 - 1s - loss: 0.4393 - accuracy: 0.8116
Epoch 7/10
 - 1s - loss: 0.3612 - accuracy: 0.8478
Epoch 8/10
 - 1s - loss: 0.3129 - accuracy: 0.8696
Epoch 9/10
 - 1s - loss: 0.1829 - accuracy: 0.9420
Epoch 10/10
 - 1s - loss: 0.1157 - accuracy: 0.9638
```

Figure 11.3.2 – Filtering of DataSet using CNN

In above screen you can see for CNN we use multiple filters to filter dataset for better prediction result and in above screen in first layer CNN use 62 X 62 image size with 32 filters and in second layer for 31 X 31 image size also it uses 32 filters and for each filter we will have best image features and prediction accuracy will be better. In above screen to run CNN I used 10 epoch/iteration and for each increase iteration accuracy get better and better and for last epoch we got 0.96% accuracy and below is the final accuracy result for both SVM and CNN

11.3.3 PREDICTING SURVIVAL RATE


Detection of Lung cancer from CT image using SVM classification and compare the survival rate of patients using 3D Convolutional neural network(3D CNN)on lung nodules data set
of Lung cancer from CT image using SVM classification and compare the survival rate of patients using 3D Convolutional neural network(3D CNN)on lung nodule
Upload Lung Cancer Dataset
Read & Split Dataset to Train & Test
Execute SVM Algorithms
Execute CNN Algorithm
Predict Lung Cancer
Survival Rate Graph
Type here to search
21:22
05-02-2021

```
SVM Survival Rate : 82.14285714285714
CNN Survival Rate : 96.37681245803833
```

Figure 11.3.3 – Comparison of Survival Rate of SVM and CNN

In above screen SVM survival rate is 82% and CNN survival rate is 96% and similarly you can go for predict button and graph button.

Prediction of time-to-event outcomes in diagnosing lung cancer based on SVM and compare the accuracy of predicted outcome with Deep CNN algorithm

In this project we are training SVM and CNN with same LUNG dataset and then calculating and comparing accuracy of both algorithms

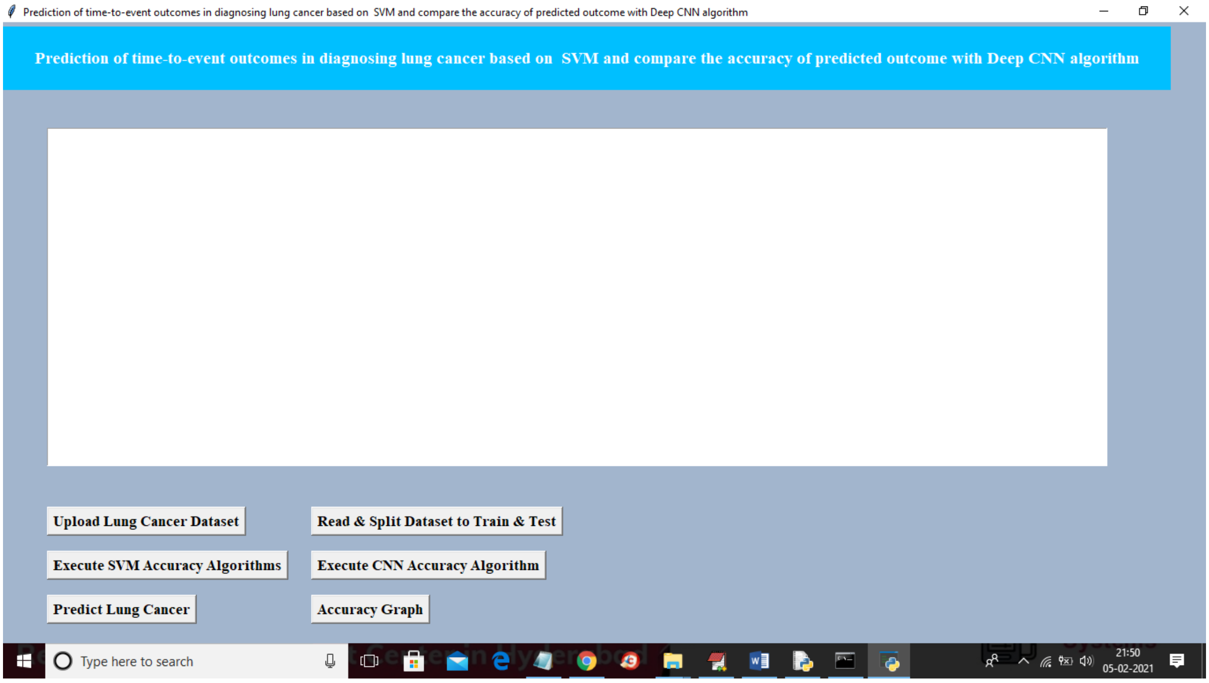
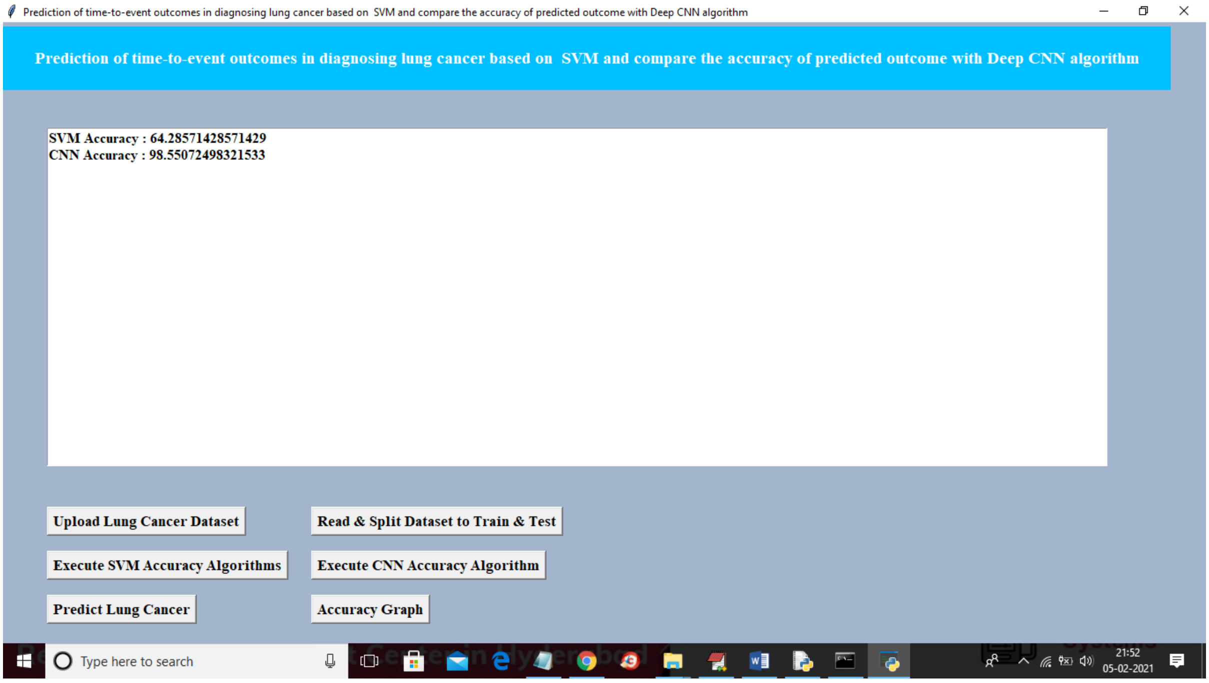

Prediction of time-to-event outcomes in diagnosing lung cancer based on SVM and compare the accuracy of predicted outcome with Deep CNN algorithm
Prediction of time-to-event outcomes in diagnosing lung cancer based on SVM and compare the accuracy of predicted outcome with Deep CNN algorithm
Upload Lung Cancer Dataset
Read & Split Dataset to Train & Test
Execute SVM Accuracy Algorithms
Execute CNN Accuracy Algorithm
Predict Lung Cancer
Accuracy Graph
Type here to search
21:50
05-02-2021

Figure 11.3.4 – Execution of SVM and CNN with their respective Accuracy

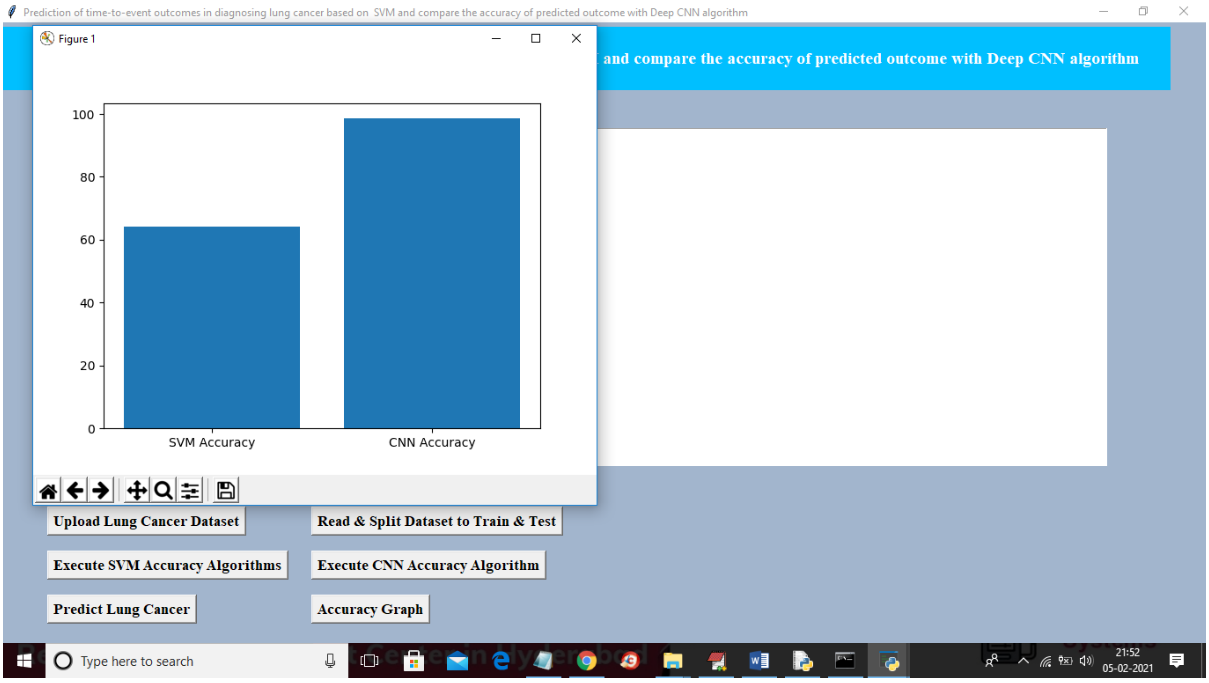
In above screen similar to first two projects upload dataset and then click on ‘read and split dataset’ button and then execute SVM with accuracy and CNN with accuracy and then you can go for predict lung cancer and accuracy graph


Prediction of time-to-event outcomes in diagnosing lung cancer based on SVM and compare the accuracy of predicted outcome with Deep CNN algorithm
Prediction of time-to-event outcomes in diagnosing lung cancer based on SVM and compare the accuracy of predicted outcome with Deep CNN algorithm
Upload Lung Cancer Dataset
Read & Split Dataset to Train & Test
Execute SVM Accuracy Algorithms
Execute CNN Accuracy Algorithm
Predict Lung Cancer
Accuracy Graph
Type here to search
21:52
05-02-2021

```
SVM Accuracy : 64.28571428571429
CNN Accuracy : 98.55072498321533
```

Figure 11.3.5 – Result of SVM and CNN Accuracy

In above screen SVM accuracy is 64% and CNN accuracy is 98% and below is the comparison graph for title 4



Prediction of time-to-event outcomes in diagnosing lung cancer based on SVM and compare the accuracy of predicted outcome with Deep CNN algorithm
and compare the accuracy of predicted outcome with Deep CNN algorithm
Figure 1
100
80
60
40
20
0
SVM Accuracy
CNN Accuracy
Upload Lung Cancer Dataset
Read & Split Dataset to Train & Test
Execute SVM Accuracy Algorithms
Execute CNN Accuracy Algorithm
Predict Lung Cancer
Accuracy Graph
Type here to search
21:52
05-02-2021

Figure 11.3.6 – Comparison Of Final Result of SVM accuracy and CNN accuracy

12. CONCLUSION

The CAD Systems are beneficial to detect cancerous nodules & have a lot to offer to modern medicine. A nodule is identified with required area by using circle fit algorithm with maximum radius which eliminates the unnecessary selection of wrong nodules. After every iteration, we get more accurate results. This led the system to provide Accuracy of 91.6%. The Sensitivity & Specificity of the system is 93.1% &98% respectively. Based on CT images, this system will give accurate and effective result of lung nodule detections benign or malignant lung nodule. In Future work, this system will help to diagnose cancer in different organs of human body. Techniques used in this system can be implemented in reducing the growth of abnormal cells or spreading to other parts of body. This system can be enhanced for MRI and Ultrasound images. The results obtained from ANN classifier are more precise and accurate but it requires more number of data inputs as compared with SVM classifier.

In existing paper, a picture handling procedures has been utilized to recognize beginning time lung malignant growth in CT examine pictures. The CT filter picture is pre-prepared pursued by division of the ROI of the lung. Discrete waveform Transform is connected for picture pressure and highlights are extricated utilizing a GLCM. The outcomes are encouraged into a SVM classifier to decide whether the lung picture is carcinogenic or not. The SVM classifier is assessed dependent on a LIDC dataset. In future the advanced level of algorithmis used to increase the level of prediction while we are in process to include the Extreme gradient boosting Algorithm to use the data set more effectively.

12.FUTURE WORK

Future scope includes enhancing the imaging system to include more features for calculating parameters like sensitivity and specificity. Future work also includes constantly upgrading the accuracy of the classifier by training it on larger and more comprehensive datasets. Furthermore, different machine learning algorithms can be incorporated to compare and understand which technique returns the best results in the context of cancer detection.

For the Improvement of the early detection and management of lung cancer that will hypothetically reduce its censorious death rate, more development can we made on CNN models. Several different knowledge based models can be trained on a larger dataset with effective hyper-parameter tuning and the model with optimal accuracy can be used for prediction. AlexNet CNN model can be trained on large dataset in order to further increase the accuracy of the model. Designing AlexNet CNN model with hyper-parameter tuning and new architecture solely for analysis. Further more advanced techniques can be added in the implementation process. More detailed image processing can be done which will improve the dataset quality. For deploying the model in real time, prediction result can be shown in web application or desktop application. The further implementation can be done in Artificial Intelligence environment in order to optimize the work. Furthermore, the CNN model can be compared with various transfer-learning models on a larger scale.

13. BIBLIOGRAPHY

- World Health Organisation.Cancer: fact Sheet no. 297. 2015 July 8. http://www.who.int/mediacentre/factsheets/fs297/en/.Jemal A, Siegel R, Xu J, et al. Cancer statistics, 2015. CA Cancer J Clin.2015; 60(5):277–300.
- The diagnosis of lung cancer (update) Published by the National Collaborating Centre for Cancer (2nd Floor, Front Suite, Park House, Greyfriars Road, Cardiff, CF10 3AF) at Velindre NHS T rust, Cardiff, Wales [2011] Database from: The Cancer Imaging Archive. http://doi.org/10.7937/K9/TCIA.2015.A6V7JIWX
- Nanusha, "Lung Nodule Detection Using Image Segmentation Methods", International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE) Volume 6, Issue 7, July 2017
- Awais Mansoor Ph-D et al, "Segmentation and Image Analysis of Abnormal Lungs at CT: Current Approaches, Challenges, and Future Trends", Radio Graphics 2015; 35:1056–1076 Published online 10.1148/rg.2015140232
- Ozge Gunaydin et al "Comparision of Lung Cancer Detection Algorithm" 2019, 978-1-7281-1013-4/19/ $31.00 © 2019 IEEE
- Madhura J, Dr. Ramesh Babu D R "A Survey on Noise Reduction Techniques for Lung Cancer Detection" International Conference on Innovative Mechanisms for Industry Applications (ICIMIA 2017), 978-1-5090-5960-7/17/$31.00 ©2017 IEEE
- Ashwin S, Kumar SA, Ramesh J, Gunavathi K: Efficient and reliable lung nodule detection using a neural network-based computer aided diagnosis system. In Emerging Trends in Electrical Engineering and Energy Management (ICETEEEM), 2012 International Conference 2012:135–142.
- Rachid Sammouda "Segmentation and Analysis of CT Chest Images for Early Lung Cancer Detection" 2016 Global Summit on Computer & Information Technology 978-1-5090-2659-3/17 $31.00 © 2017 IEEE [8] M. Egmonet-Petersen et al "Image Processing with Neural Networks- a Review", Patteren Recognation, Volum-35, No. 10, PP. 2279-2301, 2002.
- V. Vapnik, "An overview of Statastical Learning Theory", IEEE transactions on Neural Networks, Vol-10 No. 5, PP. 988-999, 1999

- A. Kanakatte, N. Mani, B. Srinivasan, and J. Gubbi, “Pulmonary Tumor Volume Detection from Positron Emission T omography Images,” 2008 International Conference on BioMedical Engineering and Informatics, 2008.
- A. Hashemi, A. H. Pilevar, and R. Rafeh, “Mass Detection in Lung CT Images Using Region Growing Segmentation and Decision Making Based on Fuzzy Inference System and Artificial Neural Network,” International Journal of Image, Graphics and Signal Processing, vol. 5, no. 6, pp. 16–24, 2013.
- Nidhi S. Nadkarni, “Detection of Lung Cancer in CT Images using Image Processing” Proceedings of the Third International Conference on Trends in Electronics and Informatics (ICOEI 2019).
- Automatic detection of a tiny lung nodule on CT utilized in a local density maximum algorithms Author: Binsheng Zhao, Gordon Gamsu.
- Quantification of the Nodule Detection in Chest CT Author: Farag, Shireen Y. Elhabian, Salwa A. Elshazly.