A

Project Report

on

Detection of Lung Infection Using Machine Learning

B. Tech, Electronics and Telecommunication Engineering

Submitted by

Siddhant Pandey (70061118026)

Under the Guidance of

Prof. Sachin Sonawane

Department of Electronics and Telecommunication Engineering

S.V.K.M's NMIMS, Mukesh Patel School of Technology

Management and Engineering.

Academic Session: 2021-2022

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING
Mukesh Patel School of Technology Management & Engineering

CERTIFICATE

This is to certify that Project titled **"Detection of Lung Infection using Machine Learning"** has been successfully completed by

Siddhant Pandey (70061118026)

Under the guidance of

Prof Sachin Sonawane

in partial completion of the requirement for Bachelor degree in Electronics and Telecommunication (EXTC) of MPSTME, SVKM's NMIMS University, Mumbai, Shirpur Campus during the academic year 2021-2022.

Date: 26th March 2022

Place: Shirpur

Prof. Sachin Sonawane
Project Mentor

Prof. Atul Patil
Head,
Department of Electronics &
Telecommunication Engineering

Dr Kamal Mehta
Associate Dean
(MPSTME-NMIMS, Shirpur Campus)

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING
Mukesh Patel School of Technology Management & Engineering

CERTIFICATE FOR APPROVAL

The Project titled "Detection of Lung Infection Using Machine Learning" being submitted by

Siddhant Pandey (70061118026)

Has been examined by us and is hereby approved for the award of degree "BACHELOR OF TECHNOLOGY in Electronics & Telecommunications" discipline for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approved the project only for the purpose for which it has been submitted.

Place: Shirpur

Internal Examiner **External Examiner**

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING
Mukesh Patel School of Technology Management & Engineering

DECLARATION

I,

"Siddhant Pandey"

the students of **Bachelor of Technology** in **Electronics and Telecommunication (EXTC) discipline, Session: 2021-22, MPSTME, Shirpur Campus,** hereby declare that the work presented in this project entitled **"Detection of Lung Infection Using Machine Learning"** is the outcome of our work, is bona fide and correct to the best of our knowledge and this work has been carried out taking care of Engineering Ethics. The work presented does not infringe any patented work and has not been submitted to any other university or anywhere else for the award of any degree or any professional diploma.

Siddhant Pandey (70061118026)

Date: 26th March 2022

DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING
Mukesh Patel School of Technology Management & Engineering.

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organization. We would like to extend my sincere thanks to all of them.

Thank my god for providing us with everything that we required in completing this project.

We are highly indebted to the teacher in charge Prof. Sachin Sonawane for his constant guidance and impeccable supervision as well as for providing necessary information regarding the project and for his support in completing the project.

We would like to express my gratitude towards my parents for their kind co-operation and encouragement, which helped us in the completion of this project.

We would like to express my special gratitude and thanks to industry persons for giving us such attention and time.

Our thanks and appreciations go to our classmates and batch mates in developing the project and to the people who have willingly helped us out with their abilities.

Siddhant Pandey (70061118026)

DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING
Mukesh Patel School of Technology Management & Engineering

ABSTRACT

The objective of this project is to investigate and foresee the Lung Diseases with assistance from Machine Learning Algorithms. The most common lung diseases are Asthma, Allergies, Chronic obstructive pulmonary disease (COPD), bronchitis, pneumonia, and lung cancer and so on.

It is important to foresee the odds of lung sicknesses before it happens and by doing those individuals can be causes and make fundamental strides before it occurs. In this project, we are going to work with a collection of data and classified it with various machine-learning algorithms.

It is an inflammatory condition of the lung affecting primarily the small air sacs known as alveoli. Symptoms typically include some combination of productive or dry cough, chest pain, fever and difficulty breathing. The severity of the condition is variable. Pneumonia is usually caused by infection with viruses or bacteria and less commonly by other microorganisms, certain medications or conditions such as autoimmune diseases.

Risk factors include cystic fibrosis, chronic obstructive pulmonary disease (COPD), asthma, diabetes, heart failure, a history of smoking, a poor ability to cough such as following a stroke and a weak immune system. Diagnosis is often based on symptoms and physical examination. Chest X-ray, blood tests, and culture of the sputum may help confirm the diagnosis. The disease may be classified by where it was acquired, such as community- or hospital-acquired or healthcare-associated pneumonia.

Table of Content

List of Figures

CHAPTER: 1

INTRODUCTION

Coronavirus Disease 2019 (COVID-19) has become a global pandemic with an exponential growth rate and an incompletely understood transmission process. The virus is having most commonly with little or no symptoms, but can also lead to a rapidly progressive and often fatal pneumonia in 2–8% of those infected. The exact mortality, prevalence, and transmission dynamics remain somewhat defined in part due to the unique challenges presented by SARS-CoV-2 infection, such as peak infectiousness at or just preceding symptom onset and a poorly understood multi-organ pathophysiology with dominant features and lethality in the lungs. The rapid rate of spread has strained healthcare systems worldwide due to shortages in key protective equipment and qualified providers, partially driven by variable access to point-of-care testing methodologies, including reverse transcription polymerase chain reaction (RT-PCR). As rapid RT-PCR testing becomes more available, challenges remain, including high false negative rates, delays in processing, variabilities in test techniques, and sensitivity sometimes reported as low as 60–70%.

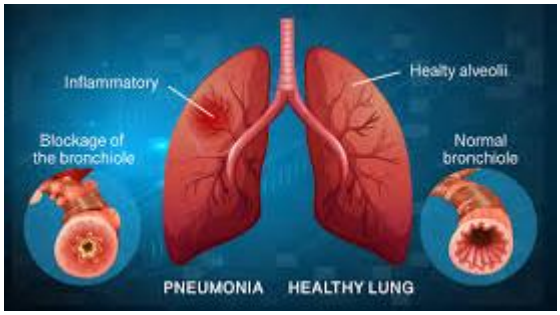

Fig 1.1: Infected Lungs

Computed tomography (CT) is a test that provides a window into pathophysiology that could shed light on several stages of disease detection and evolution. While challenges continue with rapid diagnosis of COVID-19, frontline radiologists report a pattern of infection that is somewhat characteristic with typical features including ground glass opacities in the lung periphery, rounded opacities, enlarged intra-infiltrate vessels, and later more consolidations that are a sign of progressing critical illness. While CT and RT-PCR are most often concordant, CT can also detect early COVID-19 in patients with a negative RT-PCR test, in

patients without symptoms, or before symptoms develop or after symptoms resolve. CT evaluation has been an integral part of the initial evaluation of patients with suspected or confirmed COVID-19 in multiple centres in Wuhan China and northern Italy. A recent international expert consensus report supports the use of chest CT for COVID-19 patients with worsening respiratory status or in resource-constrained environments for medical triage of patients who present with moderate–severe clinical features and a high pre-test probability of COVID-19. However, these guidelines also recommend against using chest CT in screening or diagnostic settings in part due to similar radiographic presentation with other influenza-associated pneumonias. Techniques for distinguishing between these entities may strengthen support toward use of CT in diagnostic settings.

Due to the rapid increase in number of new and suspected COVID-19 cases, there may be a role for artificial intelligence (AI) approaches for the detection or characterization of COVID-19 on imaging. CT provides a clear and expeditious window into this process, and deep learning of large multinational CT data could provide automated and reproducible biomarkers for classification and quantification of COVID-19 disease. Prior single centre studies have demonstrated the feasibility of AI for the detection of COVID-19 infection, or even differentiation from community acquired pneumonia. AI models are often severely limited in utility due to homogeneity of data sources, which in turn limits applicability to other populations, demographics, or geographies. This study aims to develop and evaluate an AI algorithm for the detection of COVID-19 on chest CT using data from a globally diverse, multi-institution dataset. Here we show robust models can be achieve up to 90% accuracy in independent test populations, maintaining high specificity in non-COVID-19 related pneumonias, and demonstrating sufficient generalizability to unseen patient populations/centres.

1.1 Diseases related to Lungs

It is an inflammatory condition of the lung affecting primarily the small air sacs known as alveoli. Symptoms typically include some combination of productive or dry cough, chest pain, fever and difficulty breathing. The severity of the condition is variable. Pneumonia is usually caused by infection with viruses or bacteria and less commonly by other microorganisms, certain medications or conditions such as autoimmune diseases.

Risk factors include cystic fibrosis, chronic obstructive pulmonary disease (COPD), asthma, diabetes, heart failure, a history of smoking, a poor ability to cough such as following a stroke and a weak immune system. Diagnosis is often based on symptoms and physical examination. Chest X-ray, blood tests, and culture of the sputum may help confirm the diagnosis. The disease may be classified by where it was acquired, such as community- or hospital-acquired or healthcare-associated pneumonia.

1.1.1 Pneumonia

Pneumonia is an infection that inflames the air sacs in one or both lungs. The air sacs may fill with fluid or pus (purulent material), causing cough with phlegm or pus, fever, chills, and difficulty breathing. A variety of organisms, including bacteria, viruses and fungi, can cause pneumonia.

Fig 1.2: Pneumonia infected Lungs

Viruses, bacteria, and fungi can all cause pneumonia. In the United States, common causes of viral pneumonia are influenza, respiratory syncytial virus (RSV), and SARS-CoV-2 (the virus that causes COVID-19). A common cause of bacterial pneumonia is Streptococcus pneumonia.

1.1.2 Chronic Obstructive Pulmonary Disease (COPD)

Chronic obstructive pulmonary disease (COPD) is a chronic inflammatory lung disease that causes obstructed airflow from the lungs. Symptoms include breathing difficulty, cough, mucus (sputum) production and wheezing.

Smoking is the main cause of COPD and is thought to be responsible for around 9 in every 10 cases. The harmful chemicals in smoke can damage the lining of the lungs and airways.

1.1.3 Asthma

Asthma is a condition in which your airways narrow and swell and may produce extra mucus. This can make breathing difficult and trigger coughing, a whistling sound (wheezing) when you breathe out and shortness of breath.

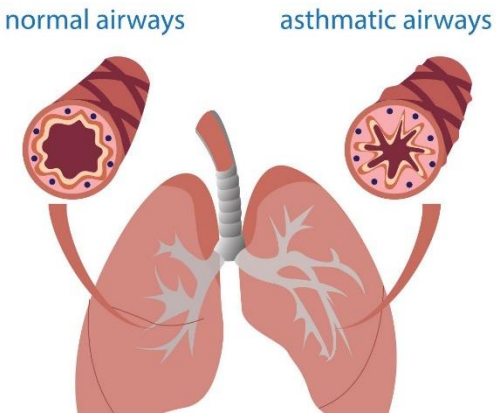

Fig 1.3: Lungs having Asthma

Common triggers include: infections like colds and flu. allergies – such as to pollen, dust mites, animal fur or feathers. smoke, fumes and pollution.

1.1.4 Bronchitis

Bronchitis is an inflammation of the lining of your bronchial tubes, which carry air to and from your lungs. People who have bronchitis often cough up thickened mucus, which can be discoloured. Bronchitis may be either acute or chronic

1.2. Impact of COVID-19 on Lungs

COVID-19 can cause lung complications such as pneumonia and, in the most severe cases, acute respiratory distress syndrome, or ARDS. Sepsis, another possible complication of COVID-19, can also cause lasting harm to the lungs and other organs. Newer coronavirus variants may also cause more airway disease, such as bronchitis, that may be severe enough to warrant hospitalization.

“As we have learned more about SARS-CoV-2 and resulting COVID-19, we have discovered that in severe COVID-19, a significant pro-inflammatory condition can result in several critical diseases, complications and syndromes,”

In pneumonia, the lungs become filled with fluid and inflamed, leading to breathing difficulties. For some people, breathing problems can become severe enough to require treatment at the hospital with oxygen or even a ventilator.

The pneumonia that COVID-19 causes tends to take hold in both lungs. Air sacs in the lungs fill with fluid, limiting their ability to take in oxygen and causing shortness of breath, cough and other symptoms.

While most people recover from pneumonia without any lasting lung damage, the pneumonia associated with COVID-19 can be severe. Even after the disease has passed, lung injury may result in breathing difficulties that might take months to improve.

“In COVID-19-related bronchitis, this is an issue of an excessive amount of sputum produced in the airways, resulting in coughing and chest congestion. The sputum also narrows the airways, making breathing more difficult,”

“As for the bronchitis, patients may experience a cough that stays with them for months after the initial infection,” he notes. “This frequent cough and ongoing chest congestion may have an impact on one’s quality of life.”

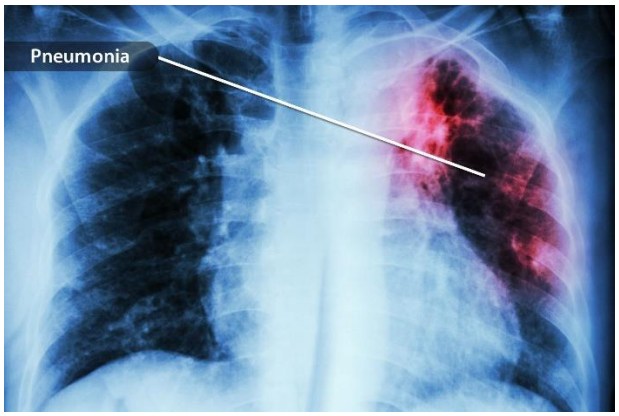

Fig 1.4: Impact of Covid-19 on Lungs

If COVID-19 pneumonia progresses, more of the air sacs can become filled with fluid leaking from the tiny blood vessels in the lungs. Eventually, shortness of breath sets in, and can lead to acute respiratory distress syndrome (ARDS), a form of lung failure. Patients with ARDS are often unable to breath on their own and may require ventilator support to help circulate oxygen in the body.

Whether it occurs at home or at the hospital, ARDS can be fatal. People who survive ARDS and recover from COVID-19 may have lasting pulmonary scarring.

Another possible complication of a severe case of COVID-19 is sepsis. Sepsis occurs when an infection reaches, and spreads through, the bloodstream, causing tissue damage everywhere it goes.

“Lungs, heart and other body systems work together like instruments in an orchestra. “In sepsis, the cooperation between the organs falls apart. Entire organ systems can start to shut down, one after another, including the lungs and heart.”

Sepsis, even when survived, can leave a patient with lasting damage to the lungs and other organs.

When a person has COVID-19, the immune system is working hard to fight the invader. This can leave the body more vulnerable to infection with another bacterium or virus on top of the COVID-19 — a superinfection. More infection can result in additional lung damage. And, about one out of four patients who develop severe COVID-19 have a superinfection, meaning these patients will take more time to heal.

CHAPTER: 2

LITERATURE REVIEW

The deep learning-based approach for chest disease detection was proposed in Abiyev and Maaitah (2018) using X-ray scans. They designed and evaluated an automated CNN model for chest disease diagnosis using an X-ray image dataset. The author disclosed significant performance of the CNN model with other soft computing techniques in terms of training accuracy, testing accuracy, and training time. The pneumonia detection from chest X-ray images has been performed using computer vision and soft computing techniques in Varela-Santos and Melin (2020). In this method, ROI were extracted using the segmentation of X-ray images, followed by texture feature extraction, and then neural network was applied for classification. CNN was used for the detection of pneumonia from chest X-ray images in Lin et al. (2019). They prepared dataset from the Kaggle repository and designed ConvNet to process the input X-ray image. Another lung disease detection approach using computer vision methods and cooperative CNN model was proposed in Wang et al. (2019).

The segmentation algorithm was applied to locate ROI in lung images and then local and global features were extracted for effective pneumonia classification. The cooperative CNN model was designed to perform the classification. Another deep learning-based approach was introduced in Thakur et al. (2021) where the author used a variant of CNN, called VGG16 for classification of pneumonia using X-ray chest images dataset. They used the transfer learning and fne-tuning approach during the learning phase. The approach for detecting pneumonia is proposed in Angeline et al. (2020) using X-ray images and CNN. They trained the CNN to classify the input X-ray image into normal or pneumonic class. The accurate and efficient pneumonia detection from the input chest X-ray images was presented in Sarkar et al. (2020). They first pre-processed the input X-ray image using bilateral faltering and contrast enhancement techniques. For classification, they build deep residual learning with the separable convolutional networks.

Another CNN-based pneumonia disease detection system was introduced in Nath and Choudhury (2020). They trained the X-ray images of normal and abnormal conditions and

prepared a model to detect the presence of pneumonia. A weighted soft computing method was proposed in Hashmi et al. (2020) using weighted anticipations from the conventional deep learning systems like DenseNet121, MobileNetV3, Exception, and ResNet. A supervised learning mechanism was proposed to predict the outcomes according to dataset representation. The ensemble technique for pneumonia detection from the chest X-ray input images was proposed in Habib et al. (2020). They designed a deep CNN model called CheXNet with VGG-19 for feature extraction. These features were ensemble for classification purposes. They introduced methods like synthetic minority oversampling technique (SMOTE), random over sampler (ROS), random under sample (RUS) to address the problem of data irregularity.

Similarly, recently various deep learning-based studies have been proposed for Covid-19 disease detection using chest X-ray images. The deep learning model proposed in Abbas et al. (2020) was called decompose, transfer, and compose (DeTraC) and used for the classification of Covid-19 disease from X-ray images. The DeTraC model is robust against the data irregularities problems. Similarly, Covid-19 disease detection from chest X-ray images using deep learning was proposed in Jain et al. (2021). They collected X-ray images of Covid-19 and normal patients. They applied pre-processing and data augmentation. For classification, CNN model was designed via automatic feature extraction. CNN was designed again in Dansana et al. (2020) for the classification of pneumonia using VGG-19, decision tree, and Inception_V2 over CT scan images and X-ray images.

The automatic framework of coronavirus disease detection and classification was proposed in Apostolopoulos and Mpesiana (2020). They build the dataset for normal and Covid-19 subjects by collecting the chest X-ray images. They prepared and analysed the CNN model for automatic disease prediction. An investigation-based approach was proposed in Pham (2021), where the authors designed fne-tuned pre-trained CNNs for the Covid-19 disease classification using chest X-ray images. They investigated the fne-tuned technique of pre-trained CNN to introduce the Artifcial Intelligence (AI) solutions for rapid and efective Covid-19 detection. The expert-designed model called COVID DetectioNet was proposed (Turkoglu 2021) for the classifcation of Covid-19 from chest X-ray images. They used features chosen from the combination of deep features. They employed a pre-trained CNN-assisted AlexNet model with a transfer learning mechanism.

The relief feature selection technique was introduced to select the robust features from all the layers of deep learning architecture. Then SVM was applied for classification. In Butt et al. (2020), the author frst reviewed the various CNN models, used for classification of lung conditions into Covid-19, viral pneumonia, and normal using chest scans.

They designed CNN model to classify pneumonia and Covid-19 lung infections using chest CT scans. In Hira et al. (2020), a method of detection and classification of Covid-19 disease into bacterial pneumonia, viral pneumonia, and the normal class was proposed. They applied the proposed methodology on various chest X-ray datasets of different sizes using a deep transfer learning approach. The two-ensemble deep transfer learning systems were designed (Gianchandani et al. 2020) for Covid-19 disease detection using chest X-ray images. They used the pre-trained models to enhance detection performance. They performed the detection of covid-19, bacterial pneumonia, and viral pneumonia.

More recently, a few deep learning models have been proposed which included CNN and transfer learning for Covid-19 prediction using chest X-ray images. In Singh et al. (2020), a CNN based model was proposed and enhanced using multiobjective adaptive differential evolution technique for Covid-19 detection using chest X-ray images. Another deep model based on densely connected convolutional networks, ResNet152V2 and VGG16 (Singh et al. 2021) was ensemble to extend the accuracy of the proposed model which classified the given chest X-ray images into Covid-19, pneumonia, tuberculosis and healthy. A modified VGG16 and DenseNet201 with ResNet152V2 was proposed for multiclass and binary classification of the chest X-ray images (Gianchandani et al. 2020). Covid-19 positive, pneumonia and normal classes have been used for multiclass classification whereas Covid-19 positive and negative classes are used for binary classification.

CHAPTER: 3

TOOLS AND SOFTWARES

3.1 Python Programming Language

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Often, programmers fall in love with Python because of the increased productivity it provides.

Fig 3.1: Python Programming Language

Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the

quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

3.2 Anaconda IDE (Anaconda Development Environment)

Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012. As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free. Package versions in Anaconda are managed by the package management system conda.

Fig 3.2: Anaconda IDE

This package manager was spun out as a separate open-source package as it ended up being useful on its own and for other things than Python.There is also a small, bootstrap version of Anaconda called Miniconda, which includes only conda, Python, the packages they depend on, and a small number of other packages. With over 25 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.

- **Open Source**

Anaconda Individual Edition is the world's most popular Python distribution platform with over 25 million users worldwide. You can trust in our long-term commitment to supporting the Anaconda open-source ecosystem, the platform of choice for Python data science.

- **Conda Packages**

Search our cloud-based repository to find and install over 7,500 data science and machine learning packages. With the conda-install command, you can start using thousands of open-source Conda, R, Python and many other packages.

- **Manage Environments**

Individual Edition is an open source, flexible solution that provides the utilities to build, distribute, install, update, and manage software in a cross-platform manner. Conda makes it easy to manage multiple data environments that can maintain and run separately without interference from each other.

Anaconda Navigator is a desktop GUI that comes with Anaconda Individual Edition. It makes it easy to launch applications and manage packages and environments without using command-line commands. Expedite your data science journey with easy access to training materials, documentation, and community resources including Anaconda.org

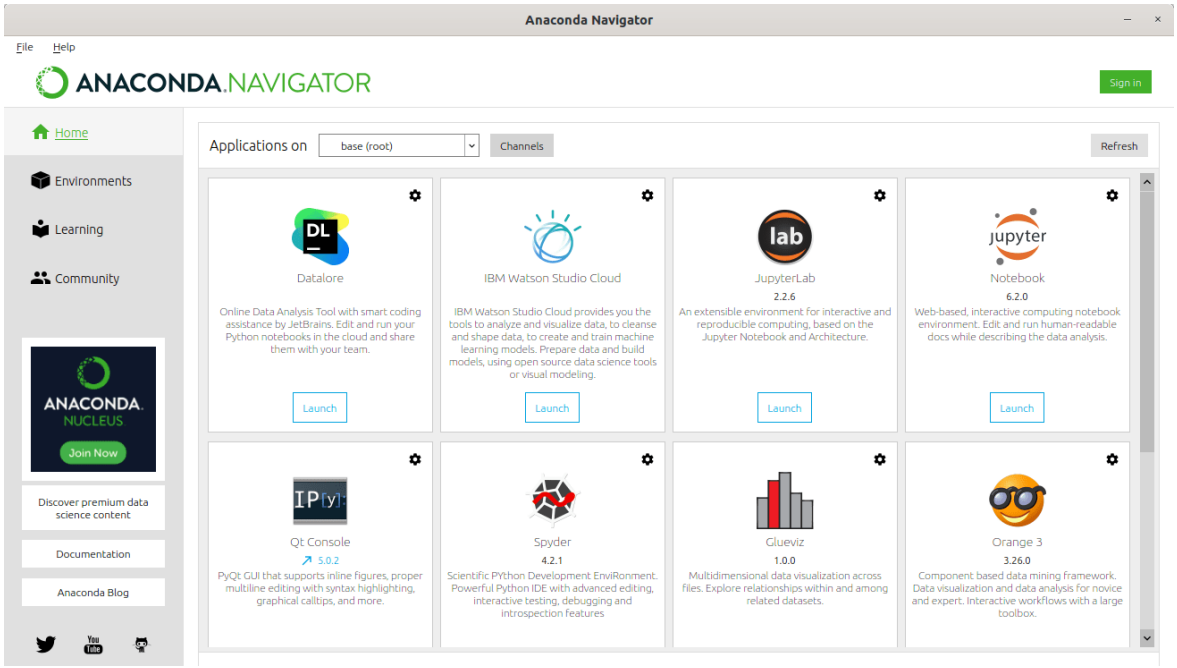

Fig 3.3 Anaconda Navigator

3.3 Visual Studio Code (VS Code)

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. Microsoft has released most of Visual Studio Code's source code on the microsoft/vscode repository of GitHub using the "Code – OSS" name, under the permissive MIT License, while the releases by Microsoft are proprietary freeware. In the Stack Overflow 2019 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 50.7% of 87,317 respondents reporting that they use it.

Microsoft at the 2015 Build conference first announced visual Studio Code on April 29, 2015. A Preview build was released shortly thereafter. On November 18, 2015, Visual Studio Code was released under the MIT License, having its source code available on GitHub. Extension support was also announced. On April 14, 2016, Visual Studio Code graduated from the public preview stage and was released to the Web.

Fig 3.4: Visual Studio Code

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python and C++. It is based on the Electron framework, which is used to develop Node.js Web applications that run on the Blink layout engine. Visual Studio Code employs the same editor component (codenamed "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services).

Instead of a project system, it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a language-agnostic code editor for any language. It supports a number of programming languages and a set of features that differs per language. Unwanted files and folders can be excluded from the project tree via the settings. Many Visual Studio Code features are not exposed through menus or the user interface but can be accessed via the command palette. Visual Studio Code can be extended via extensions, available through a central repository. This includes additions to the editor and language support. A notable feature is the ability to create extensions that add support for new languages, themes, and debuggers, perform static code analysis, and add code linters using the Language Server Protocol.

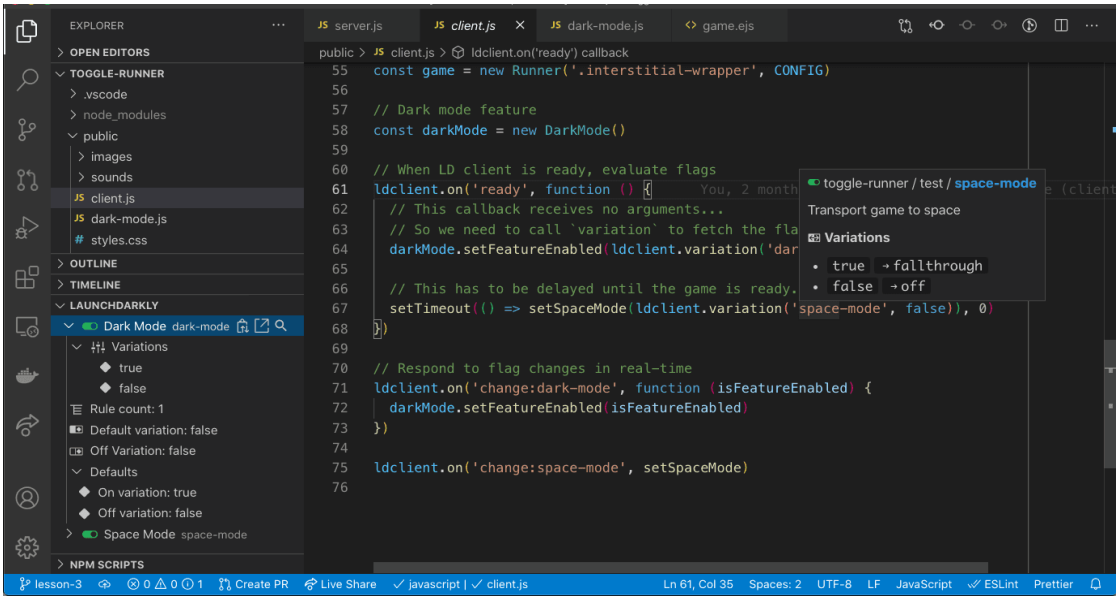

Fig 3.5: Visual Studio Code Interface

Visual Studio Code includes multiple extensions for FTP, allowing the software to be used as a free alternative for web development. Code can be synced between the editor and the server, without downloading any extra software. Visual Studio Code allows users to set the code page in which the active document is saved, the newline character, and the programming language of the active document. This allows it to be used on any platform, in any locale, and for any given programming language.

- **Language support**

Out-of-the-box, Visual Studio Code includes basic support for most common programming languages. This basic support includes syntax highlighting, bracket matching, code folding,

and configurable snippets. Visual Studio Code also ships with IntelliSense for JavaScript, TypeScript, JSON, CSS, and HTML, as well as debugging support for Node.js. Support for additional languages can be provided by freely available extensions on the VS Code Marketplace.

- **Data collection**

Visual Studio Code collects usage data and sends it to Microsoft, although this can be disabled. In addition, because of the open-source nature of the application, the telemetry code is accessible to the public, who can see exactly what is collected. According to Microsoft, the data is shared with Microsoft-controlled affiliates and subsidiaries, although law enforcement may request it as part of a legal process.

- **Version control**

Source control is a built-in feature of Visual Studio Code. It has a dedicated tab inside of the menu bar where you can access version control settings and view changes made to the current project. To use the feature you must link Visual Studio Code to any supported version control system (Git, Apache Subversion, Perforce, etc.). This allows you to create repositories as well as make push and pull requests directly from the Visual Studio Code program.

3.4 MySQL

MySQL is an open-source relational database management system (RDBMS).Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

Fig 3.6 MySQL

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB. MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often, MySQL is used with other programs to implement applications that need relational database capability.

MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Facebook, Flickr, MediaWiki, Twitter, and YouTube.

3.5 Jupyter Notebook/Lab

JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data. JupyterLab is flexible: configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning. JupyterLab is extensible and modular: write plugins that add new components and integrate with existing ones.

Fig 3.7: Jupyter Notebook/Lab

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

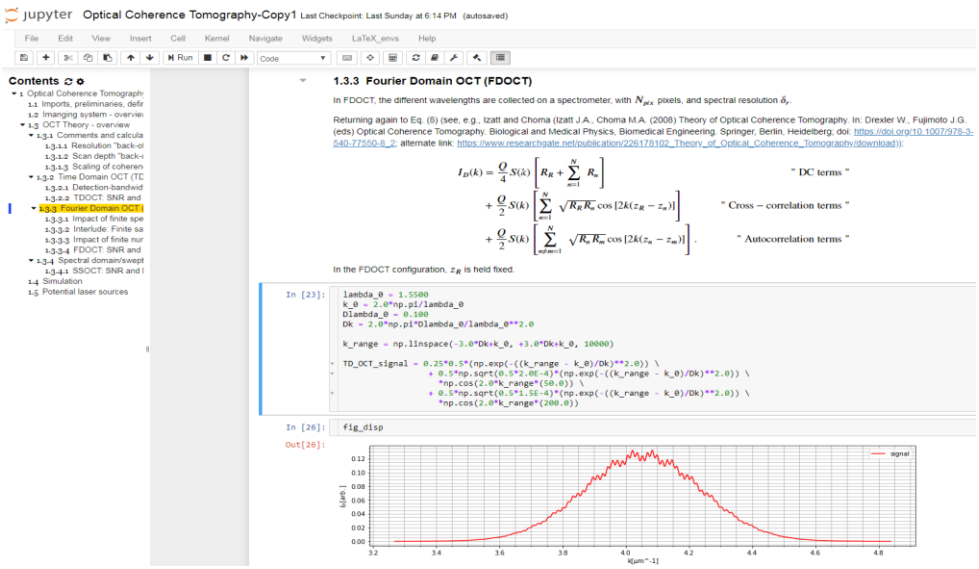

Fig 3.8: Jupyter Notebook

3.6 Machine Learning

Machine learning (ML) is the study of computer algorithms that improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine,

email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers; but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.

Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than having human programmers specify every needed step.

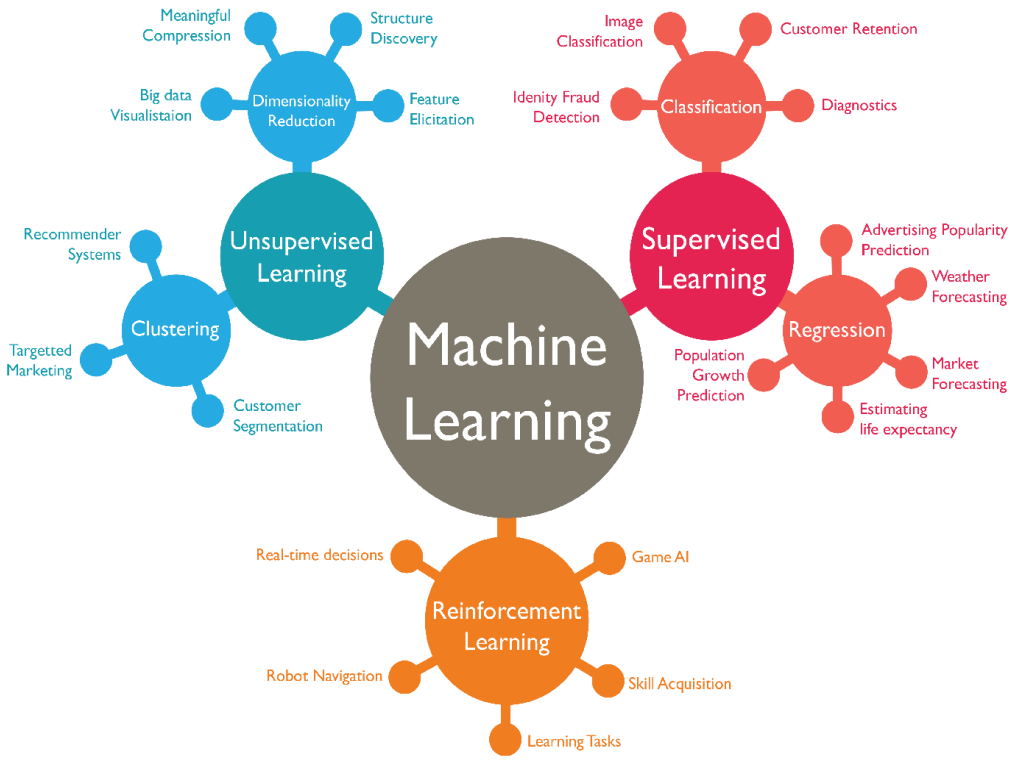

Fig 3.9: Machine Learning Chart

The discipline of machine learning employs various approaches to teach computers to accomplish tasks where no fully satisfactory algorithm is available. In cases where vast numbers of potential answers exist, one approach is to label some of the correct answers as valid.
This can then be used as training data for the computer to improve the algorithm(s) it uses to determine correct answers. For example, to train a system for the task of digital character recognition, the MNIST dataset of handwritten digits has often been used. Machine learning approaches are traditionally divided into three broad categories, depending on the nature of the "signal" or "feedback" available to the learning system:

3.6.1 Supervised learning

A support-vector machine is a supervised learning model that divides the data into regions separated by a linear boundary. Here, the linear boundary divides the black circles from the white. Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a supervisory signal. In the mathematical model, each training example is represented by an array or vector, sometimes called a feature vector, and the training data is represented by a matrix. Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs.

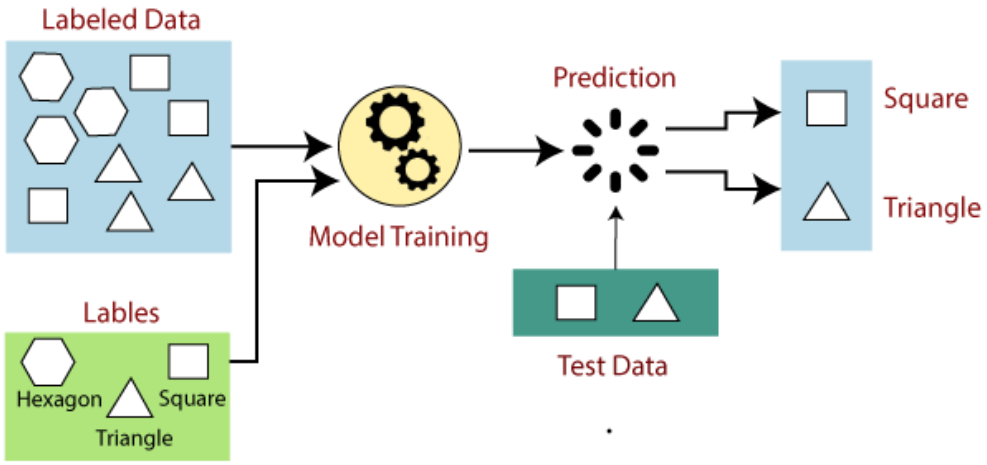

Fig 3.10: Supervised Learning

Types of supervised learning algorithms include active learning, classification and regression. Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical

value within a range. As an example, for a classification algorithm that filters emails, the input would be an incoming email, and the output would be the name of the folder in which to file the email.

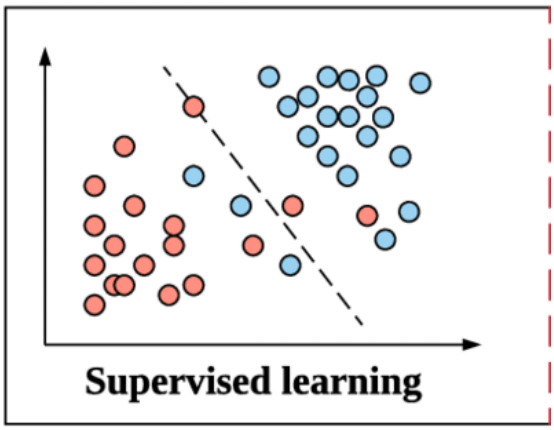

Fig 3.11: Supervised Learning Graphical Representation

3.6.2 Unsupervised Learning

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of density estimation in statistics, such as finding the probability density function. However, unsupervised learning encompasses other domains involving summarizing and explaining data features.

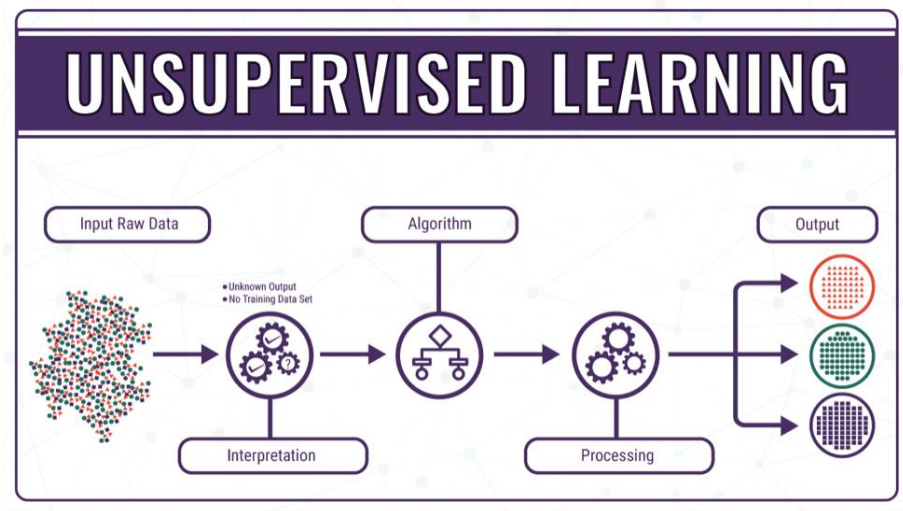

Fig 3.12: Unsupervised Learning

Cluster analysis is the assignment of a set of observations into subsets (called clusters) so that observations within the same cluster are similar according to one or more predesignated criteria, while observations drawn from different clusters are dissimilar. Different clustering techniques make different assumptions on the structure of the data, often defined by some similarity metric and evaluated, for example, by internal compactness, or the similarity between members of the same cluster, and separation, the difference between clusters. Other methods are based on estimated density and graph connectivity.

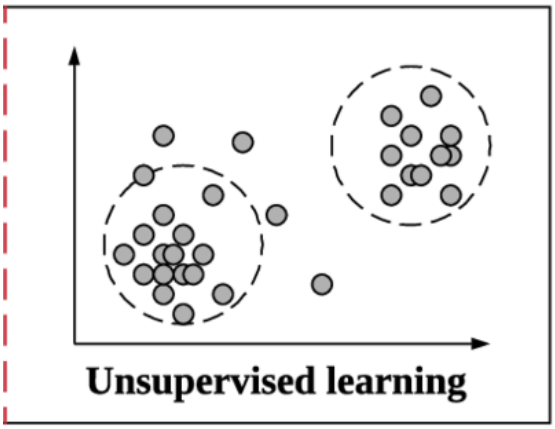

Fig 3.12: Unsupervised Learning Graphical Representation

3.6.3 Semi-Supervised Learning

Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data). Some of the training examples are missing training labels, yet many machine-learning researchers have found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce a considerable improvement in learning accuracy.

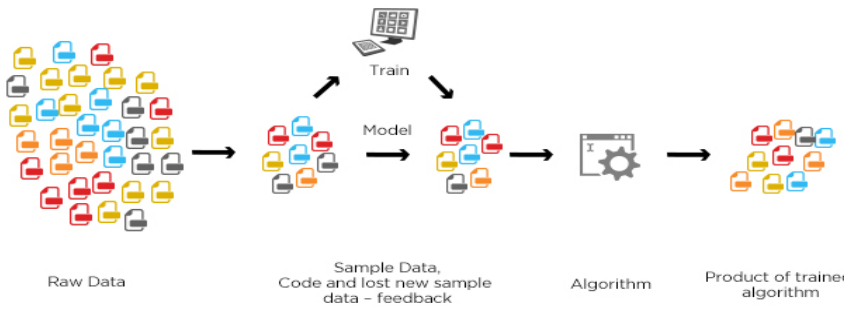

Fig 3.12: Semi-Supervised Learning

In weakly supervised learning, the training labels are noisy, limited, or imprecise; however, these labels are often cheaper to obtain, resulting in larger effective training sets.

3.6.5 Reinforcement Learning

Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward. Due to its generality, the field is studied in many other disciplines, such as game theory, control theory, operations research, information theory, simulation-based optimization, multi-agent systems, swarm intelligence, statistics and genetic algorithms. In machine learning, the environment is typically represented as a Markov decision process (MDP). Many reinforcement-learning algorithms use dynamic programming techniques. Reinforcement learning algorithms do not assume knowledge of an exact mathematical model of the MDP, and are used when exact models are infeasible. Reinforcement learning algorithms are used in autonomous vehicles or in learning to play a game against a human opponent.

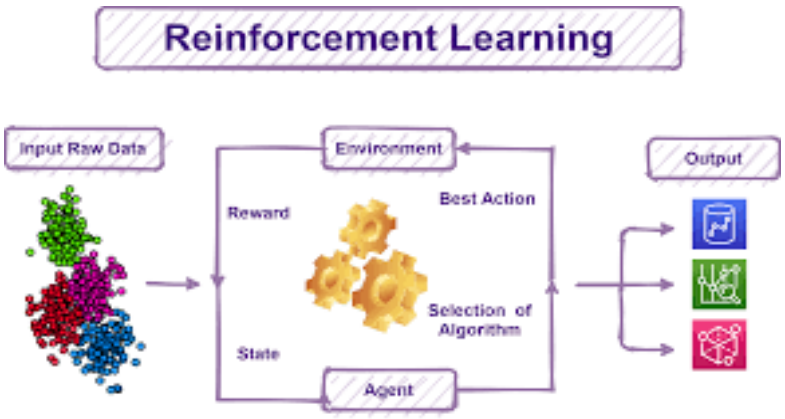

Fig 3.14: Reinforcement Learning

3.7 Deep Learning

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost.

Deep learning utilizes both structured and unstructured data for training. Practical examples of deep learning are Virtual assistants, vision for driverless cars, money laundering, face recognition and many more.

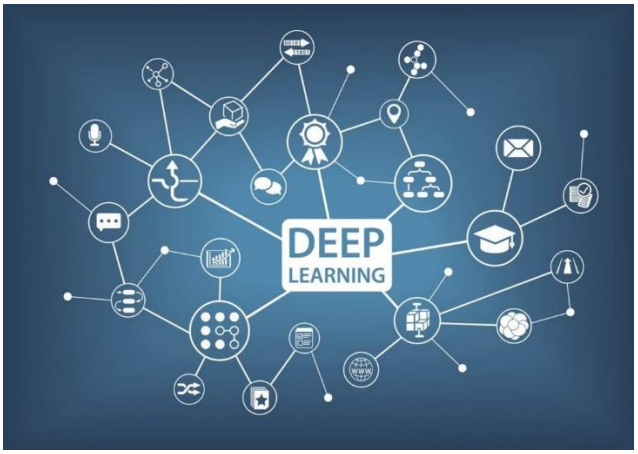

Fig 3.15: Deep Learning

Deep Learning is called Deep because of the number of additional "Layers" we add to learn from the data. If you do not know it already, when a deep learning model is learning, it is simply updating the weights through an optimization function. A Layer is an intermediate row of so-called "Neurons"

CHAPTER: 4

BLOCK DIAGRAM

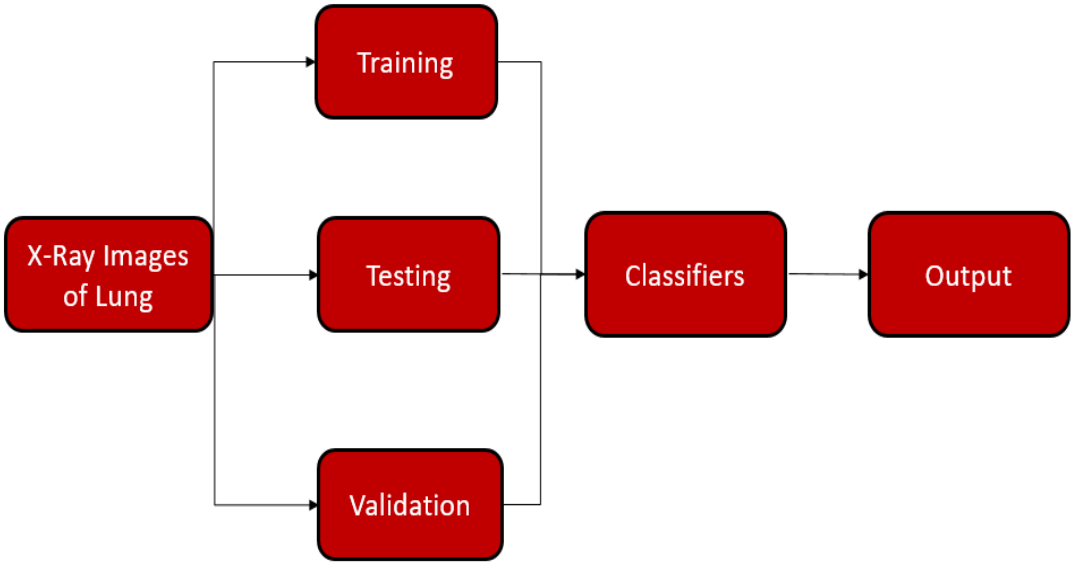

Fig 4.1: Block Diagram

- **Input:** This is the first step in this Project is that we will be providing a dataset of lungs X-Ray Images of patients suffering from lung disease. It will contain two types of images Normal lung X-Ray image and infected lung X-Ray Image.

- **Training:** The sample of the data used to fit the model. The actual dataset used to train the model (weights and biases in the case of Neural Networks) the model sees and learn from the data. We are allocating 70% of the dataset to the training set.

- **Testing:** The Testing is done because the validation set is heavily used in model creation, it is important to hold back a completely separate stronghold of data - the test set. We can run evaluation metrics on the test set at the very end of the project, to get a sense of how well the model will do in production. We are allocating 20% of the dataset to the test set.

- **Validation:** The validation set is a separate section of the dataset that we will use during training to get a sense of how well the model is doing on images that are not being used in training. We are holding out 10% of the dataset for the validation set.

- **Classification:** In this step the X-Ray image will be classified into different groups by the use of different classifiers like Convolutional Neural Network (CNN), K-Nearest Neighbor, Decision Tree.

CHAPTER: 5

METHODOLOGY

There are certain steps to develop the Machine Learning Model to achieve the desired output.

Step 1: Collecting the Dataset

Contents in the Dataset: Lungs X-Ray Images

Link of the Dataset: https://www.kaggle.com/code/kashyapgohil/pneumonia-detection-using-cnn/notebook

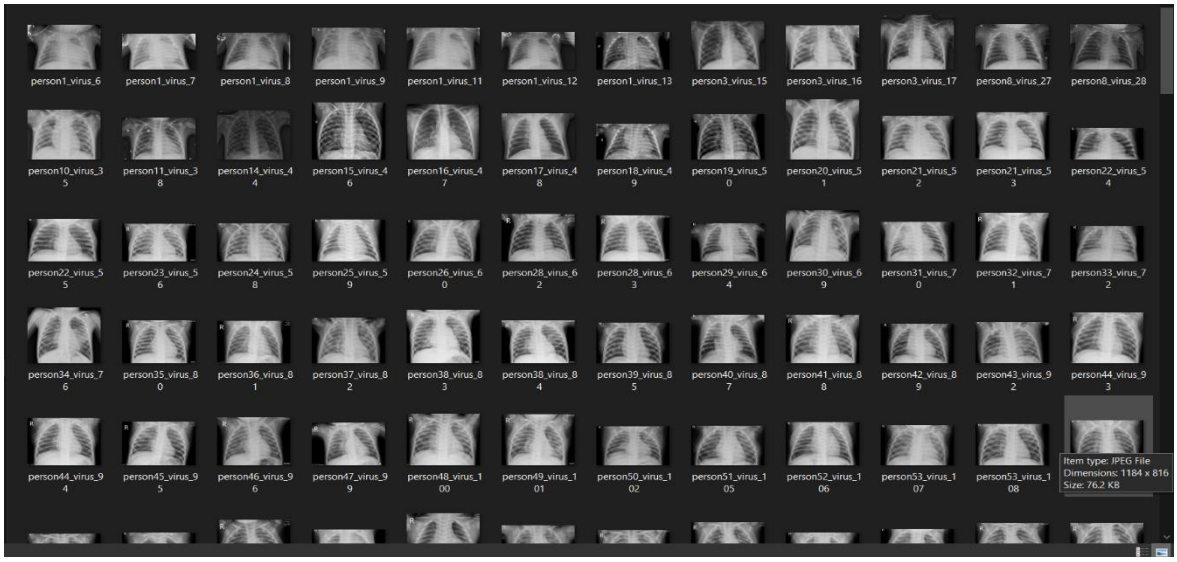

Fig 5.1: Lung X-Ray Images

Fig 5.2: Lungs X-Ray Images

Step 2: Importing the Dataset.

Step 3: Dividing the imported dataset into 3 parts namely:-
Training, Testing, Validation.

Step 4: Creating the algorithms for the particular Dataset.

Step 5: Designing the Neural Network.

1. Convolutional Neural Network (CNN)
2. K-Nearest Neighbor (KNN)
3. Decision Tree

Step 6: Performance Analysis of the model on different Parameters like:

1. Accuracy
2. Recall
3. Specificity
4. F1 Score
5. ROC (Receivers Operating Characteristics)
6. Confusion Matrix
7. Precision Recall or PR Curve
8. PR Vs ROC Curve

- **Accuracy:** Machine learning model accuracy is the measurement used to determine which model is best at identifying relationships and patterns between variables in a dataset based on the input, or training, data.

- **Recall:** Recall is a metric that quantifies the number of correct positive predictions made out of all positive predictions that could have been made. Unlike precision that only comments on the correct positive predictions out of all positive predictions, recall provides an indication of missed positive predictions.

- **Specificity:** Specificity is defined as the proportion of actual negatives, which got predicted as the negative (or true negative). This implies that there will be another

proportion of actual negative, which got predicted as positive and could be termed as false positives.

- **F1 Score:** F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution.

- **ROC (Receivers Operating Characteristics):** A receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The method was originally developed for operators of military radar receivers starting in 1941, which led to its name. An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate. False Positive Rate.

- **Confusion Matrix:** A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model.

- **Precision Recall or PR Curve:** A PR curve is simply a graph with Precision values on the y-axis and Recall values on the x-axis. In other words, the PR curve contains TP/(TP+FN) on the y-axis and TP/(TP+FP) on the x-axis. It is important to note that Precision is also called the Positive Predictive Value (PPV).

CHAPTER: 6

RESULTS AND DISCUSSION

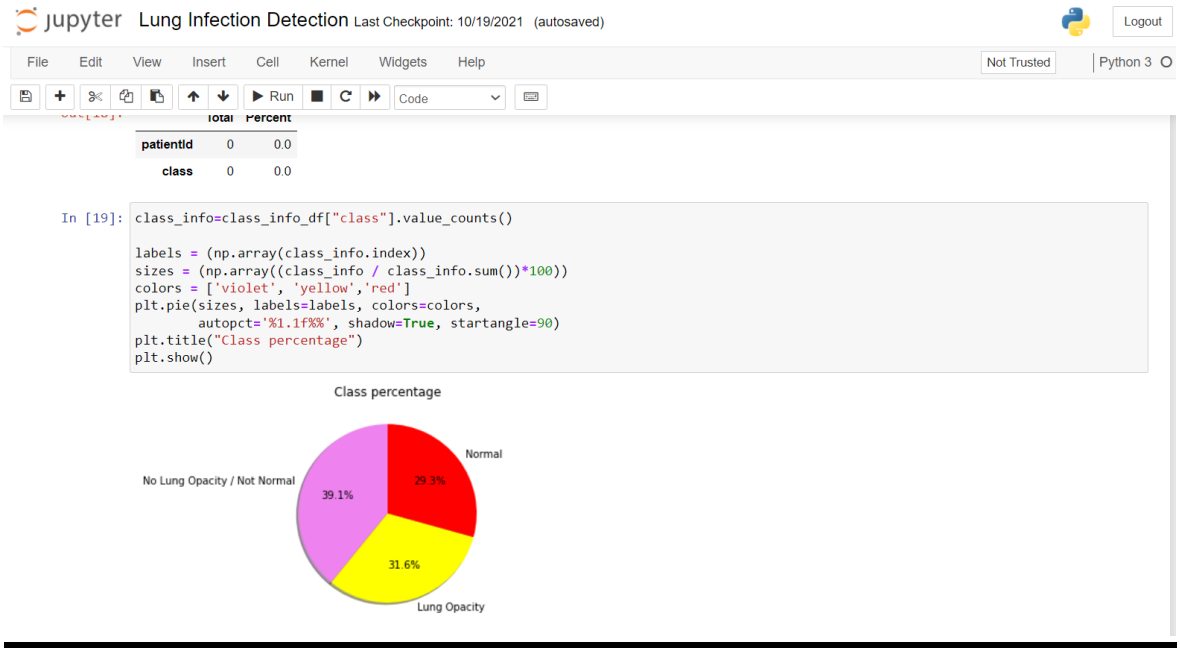

Fig 6.1: First Result

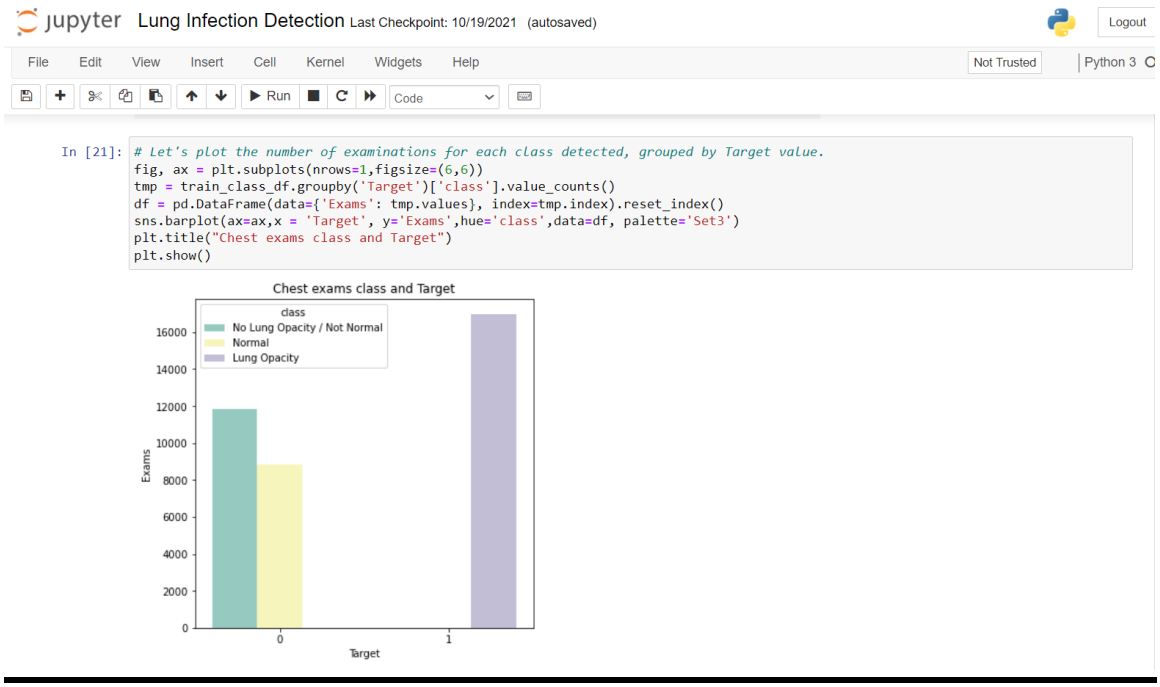

Fig 6.2: Second Result

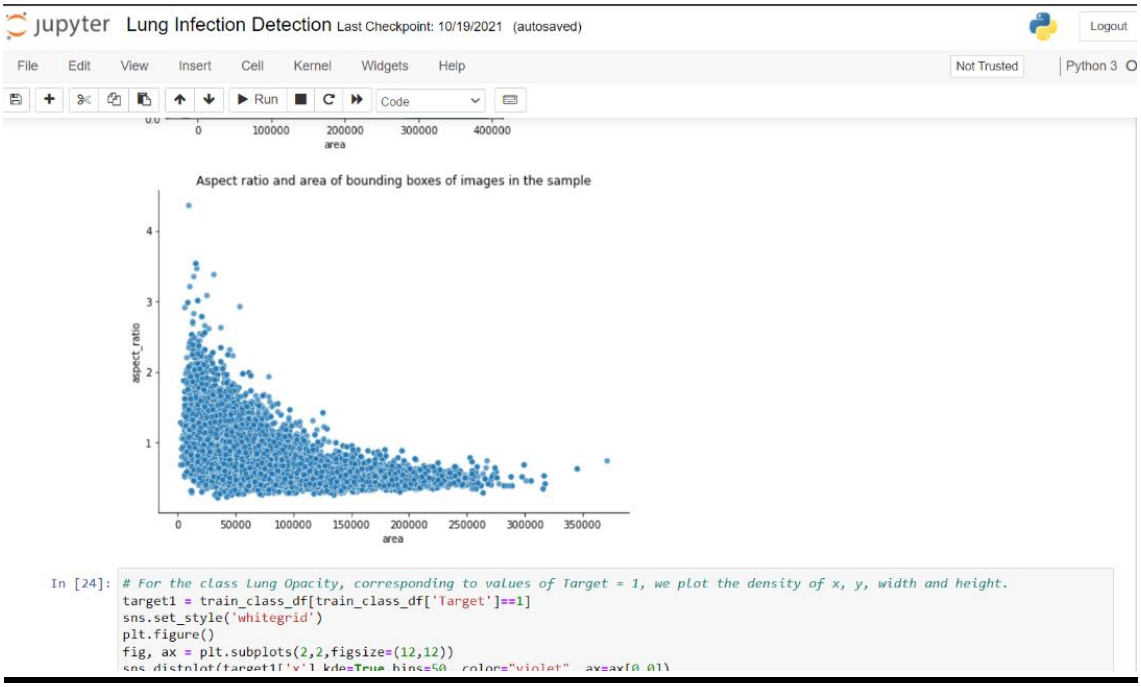

Fig 6.3: Third Result

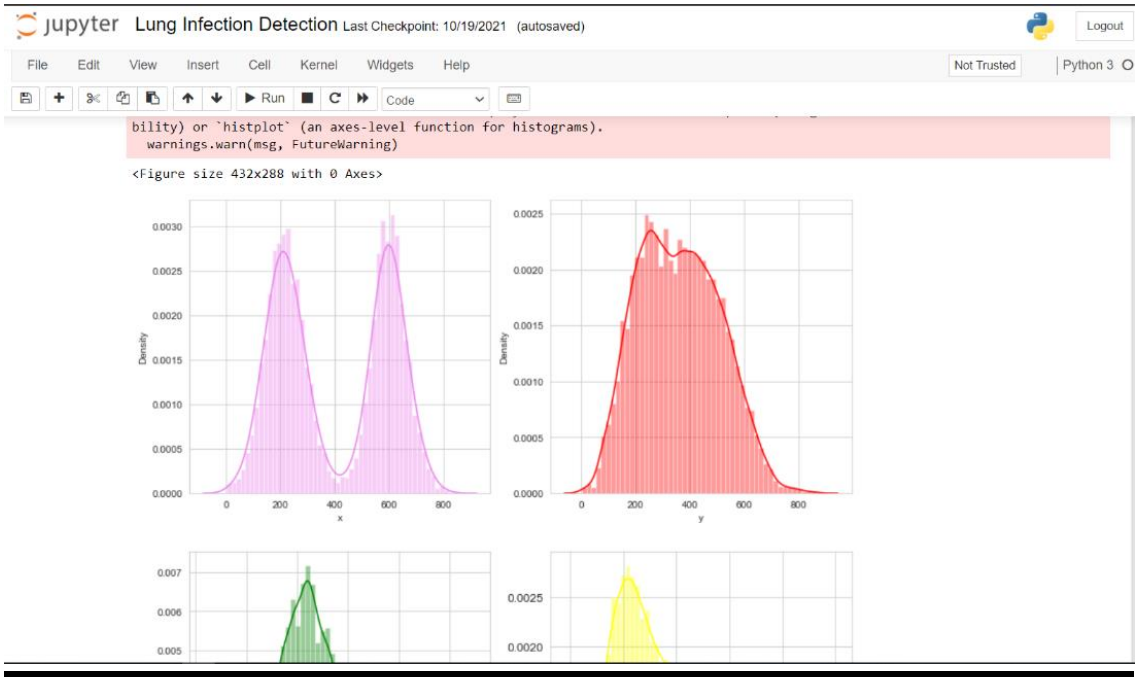

Fig 6.4: Fourth Result

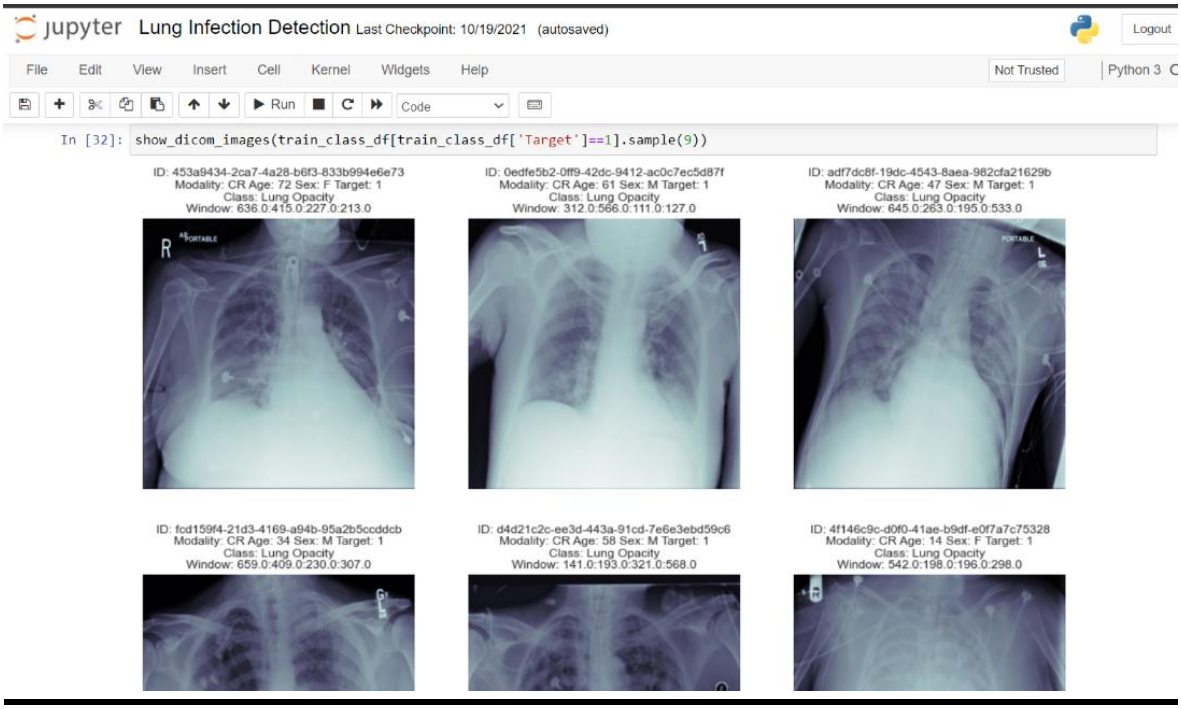

Fig 6.5: Fifth Result

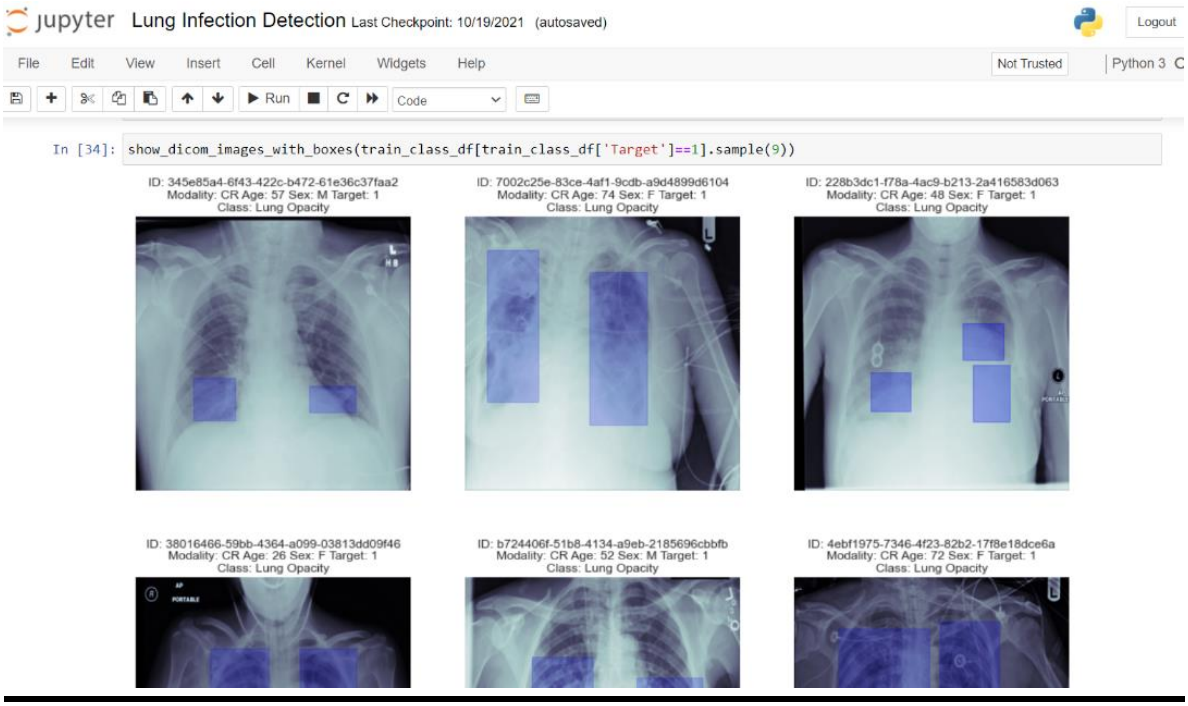

Fig 6.6: Sixth Result

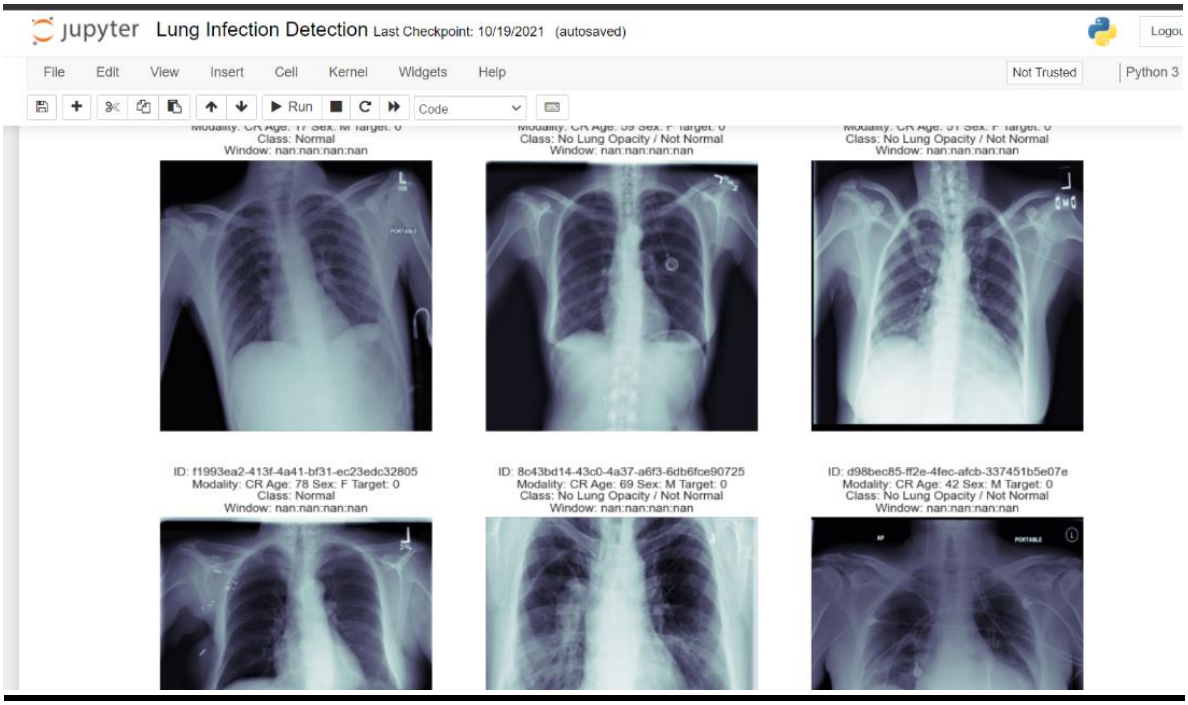

Fig 6.7: Seventh Result

CHAPTER: 7

CONCLUSION

In this study, we implemented a two-path CNN pipeline that incorporates the three distinct input images, to automatically segment the infected tissues inside the lung caused due to the COVID-19 from CT images. For a better demonstration of the tissues to extract more key features inside the CNN model, we showed the input CT image represented in the two other different ways in which each of them includes some unique information. Due to inflammation inside the lung because of COVID-19, infected areas near the border of the lung are highly difficult to segment.

So, our algorithm first employed a Z score normalization technique to obtain a more distinguishable lung border from the original image. Then, by using a fuzzy clustering method, all tissues in the image is clustered and obtain a distinct pixel value for all pixels corresponding to each cluster. This approach helps the CNN pipeline for decreasing the convolutional layers for extracting some key features and leads to a drop in the training time of the pipeline and increase the final efficiency. Then, an LDN encoding approach was implemented for representing the information of the images in another form to extract more essential details from the input image. This strategy roots in the fact that sometimes by changing the representation domain (like frequency domain rather than the time domain) some other substantial features can be observed.

We also represented a new two-route CNN model that considered semi global and local information to categorize each pixel in the input image to one of the two normal and infected tissues. The number of the convolutional layers in the global route is more than the local route, while the kernel size for all convolutional layers is the same. To overcome the overfitting problems and boost efficiency, using data augmentation methods, the number of samples has been increased. Lastly, using the CT image and two obtained images, our CNN structure was trained. The suggested two-route segmentation pipeline was appraised on a public dataset which 70% of data for training, 10% for validating, and 20% for testing were used.

Our significant findings demonstrate that our CNN pipeline and three distinct input images gained the following:

(1) Acceptable performance even if the infected area shared an extended border with touching tissues.

(2) Appropriately robust as indicated by the negligible standard deviations which show the uniformity of the values for all the nine criteria.

(3) Accomplished well in the detection and segmentation process even for the intricate cases with numerous unalike categories of the infection, which had the amoeboid shapes and analogous thicknesses.

The proposed architecture satisfactorily overcomes the difficulty of failing in accurate detection of the lesions at the presence of the similar adjacent tissues and identification of an uneven border where it seemed to not properly appear to exist with an aim to reach superior outcomes. In addition, the employed technique does not require more extra parameters for feeding into the algorithm apart from one CT image to define the position of the lesions and border detection. But the functional limitation of this architecture is that the white matter (pulmonary nodules) inside the normal lung near the border of a lesion cannot properly be recognized from the infected tissue. We think that by increasing the training samples this problem can be solved.

CHAPTER: 8

FUTURE SCOPE

Since machine learning needs you to know computer programming, statistics and data evaluation, the future scope of your machine learning career can also be in leadership roles in automation or analytics environments that use data science, big data analysis, AI integration.

- The Main objective to build this Machine Learning Model is to help the society from the deadly viruses like COVID-19.
- By doing some upgradation this model can also be used in many other medical experiments and can also be used as a medical equipment to help the patients suffering from Lung Disease.

- **Optimising Operations**

The most common use case in optimising operations is in document management. Today, there are a large number of robotic process automation and computer vision companies such as UIPath, Xtracta, and ABBYY etc. enabling this. The future of machine learning will aim higher though. There are emerging ML technologies that enable retail stores to monitor body temperatures and mask-wearing using thermal imaging and computer vision tech towards a safer return from COVID-19 to normalcy. Sensors and IoT technologies are helping manufacturing operations optimise granularly across the supply chain. The renewable energy industry is using AI to mitigate the unpredictability of sources.

- **Safer Healthcare**

We have been seeing significant growth in machine learning being used to predict and support COVID-19 strategies. The healthcare industry itself has been long using ML for a wide range of purposes; we believe that the future scope of machine learning will undertake more complex use cases. Robots performing complicated surgeries precisely. ML programs reading patient history, records, reports etc. to devise personalised treatment plans. IBM Watson Oncology is an important project in this space. Wearable technology for disease prevention and elder healthcare monitoring is also making great strides.

- **Fraud Prevention**

Banks and other financial institutions use machine-learning-based fraud detection technology to stop malpractices (although the irony of proving 'I am not a robot' to a machine is not lost!).

Banks are building machine-learning algorithms based on historical data to predict fraudulent transactions. Classification and regression methods are being used to identify and filter out phishing emails. Machine learning and computer vision algorithms are checking for identity matching across key databases in real-time to prevent identity theft. These pattern-matching techniques are also used to identify fake documents to prevent forgery.

- **Mass Personalisation**

Retail, social media and entertainment platforms use ML to give customers personalised services and experiences. The face swap filter uses algorithms based on image recognition and computer vision to detect and (well, almost) accurately exchange facial features. E-commerce and media platforms are using ML to offer hyper-personalised experiences, as well as offer freemium models of payment.

CHAPTER: 9

REFERENCES

[1] A. Jaiswal, N. Gianchandani, D. Singh, V. Kumar, and M. Kaur, "Classification of the COVID-19 infected patients using DenseNet201 based deep transfer learning," Journal of Biomolecular Structure and Dynamics, vol. 2, pp. 1–8, 2020.

[2] F. Shan, Y. Gao, J. Wang et al., "Lung infection quantification of COVID-19 in CT images with deep learning," 2020, arXiv preprint arXiv:2003.04655.

[3] M. Ahmadi, A. Sharifi, S. Dorosti, S. Jafarzadeh Ghoushchi, and N. Ghanbari, "Investigation of effective climatology parameters on COVID-19 outbreak in Iran," Science of the Total Environment, vol. 729, p. 138705, 2020.

[4] X. Wang, X. Deng, Q. Fu et al., "A weakly-supervised framework for COVID-19 classification and lesion localization from chest CT," IEEE Transactions on Medical Imaging, vol. 39, no. 8, pp. 2615–2625, 2020.

[5] S. Dorosti, S. Jafarzadeh Ghoushchi, E. Sobhrakhshankhah, M. Ahmadi, and A. Sharifi, "Application of gene expression programming and sensitivity analyses in analyzing effective parameters in gastric cancer tumor size and location," Soft Computing, vol. 24, no. 13, pp. 9943–9964, 2020.

[6] L. Liu, S. Oza, D. Hogan et al., "Global, regional, and national causes of under-5 mortality in 2000-15: an updated systematic analysis with implications for the Sustainable Development Goals," The Lancet, vol. 388, no. 10063, pp. 3027–3035, 2016.

[7] P. Rajpurkar, J. Irvin, K. Zhu et al., "Chexnet: radiologist-level pneumonia detection on chest X-rays with deep learning," 2017.

[8] S.-A. Zhou and A. Brahme, "Development of phase-contrast X-ray imaging techniques and potential medical applications," Physica Medica, vol. 24, no. 3, pp. 129–148, 2008.

[9] T. Paulraj, K. S. Chelliah, and S. Chinnasamy,“Lung computed axial tomography image segmentation using possibilistic fuzzy C-means approach for computer aided diagnosis system,” International Journal of Imaging Systems and Technology, vol. 29, no. 3, pp. 374–381, 2019.

[10] R. H. Kallet, “The vexing problem of ventilator-associated pneumonia: observations on pathophysiology, public policy, and clinical science,” Respiratory Care, vol. 60, no. 10, pp. 1495–1508, 2015.