**A Contemporary Technique for Lung Disease Prediction using Deep Learning**

**ABSTRACT:**

The impact of disease on health is escalating quickly as a result of environmental changes, climate change, adjustments in lifestyle, and other reasons. Health problems are now more likely as a result. In 2016, 3.4 million individuals died from chronic obstructive pulmonary disease (COPD), which is typically brought on by smoking and pollution, while asthma claimed the lives of 400,000 people. Particularly in developing and low-income nations, where millions of people struggle with poverty and air pollution, there is a very high risk of lung disorders. According to WHO estimates, illnesses linked to home air pollution, such as asthma and pneumonia, cause nearly 4 million preventable deaths each year. Worldwide, lung illness is a prevalent occurrence. These include pneumonia, asthma, TB, fibrosis, chronic obstructive pulmonary disease, and others. The early detection of lung illness is important. The method of discovering and classifying lung disorders into different groups using medical imaging has been improved because to the development of deep learning. For this, a variety of machine learning and image processing models have been created. Convolutional neural networks (CNNs), one type of existing deep learning approach, are used to forecast lung illness. CNN's basic version is inadequate. As a result, we propose a novel deep learning framework for predicting lung diseases based on the VGG16 Architecture. The goal of this research is to develop a VGG16 architecture-based lung disease detection model. Early detection and diagnosis of lung disease are essential in the medical field because doing so will make it easier to manage patients' future clinical care. The X-ray picture dataset obtained from the Kaggle source is subjected to the VGG16 Architecture. The dataset's sample and full versions are taken into consideration. The VGG16 Architecture beats existing techniques for both whole and sample datasets in terms of measures including precision, recall, F1 score, and validation accuracy. Therefore, the proposed VGG16 Architecture will make it easier for both professionals and clinicians to detect lung problems. This improvement has greatly helped the medical community's ability to treat patients quickly.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT**  **LIST OF FIGURES**  **LIST OF SYMBOLS** | i  v  vii |
| 1. | **CHAPTER 1 : INTRODUCTION**   * 1. GENERAL   2. SCOPE OF THE PROJECT   3. OBJECTIVE   1.4 EXISTING SYSTEM  1.4.1 EXISTINGSYSTEM DISADVANTAGES  1.5 LITERATURE SURVEY  1.6 PROPOSED SYSTEM  1.6.1 PROPOSED SYSTEM ADVANTAGES |  |
| 2. | **CHAPTER 2 :PROJECT DESCRIPTION**  2.1 GENERAL  2.2 METHODOLOGIES  2.2.1 MODULES NAME  2.2.2 MODULES EXPLANATION  2.3 TECHNIQUE OR ALGORITHM |  |
| 3. | **CHAPTER 3 : REQUIREMENTS**  3.1 General  3.2 Hardware REQUIREMENTS  3.3 Software REQUIREMENTS |  |
| 4. | **CHAPTER 4 :SYSTEM DESIGN**  **4.1 general**  **4.2 uml diagrams**  4.2.1 USE CASE DIAGRAM  4.2.2 CLASS DIAGRAM  4.2.3 OBJECT DIAGRAM  4.2.4 STATE DIAGRAM  4.2.5 activity diagram  4.2.6 SEQUENCE DIAGRAM  4.2.7 COLLABORATION DIAGRAM  4.2.8 COMPONENT DIAGRAM  4.2.9 DATA FLOW DIAGRAM  4.2.10 DEPLOYMENT DIAGRAM  4.2.11 SYSTEM ARCHITECTURE |  |
| 5. | **CHAPTER 5 : DEVELOPMENT TOOLS**  5.1 general  5.2 History of Python  5.3 Importance of Python  5.4 Features of Python  5.5 Libraries used in python |  |
| 6. | **CHAPTER 6 :IMPLEMENTATION**  6.1 GENERAL  6.2 IMPLEMENTATION |  |

|  |  |  |
| --- | --- | --- |
| 7. | **CHAPTER 7 :SNAPSHOTS**  7.1 GENERAL  7.2 VARIOUS SNAPSHOTS |  |
| 8. | **CHAPTER 8 :SOFTWARE TESTING**  8.1 GENERAL  8.2 DEVELOPING METHODOLOGIES  8.3 TYPES OF TESTING |  |
| 9. | **CHAPTER 9 :**  **FUTURE ENHANCEMENT**  9.1 FUTURE ENHANCEMENTS |  |
| **10** | **CHAPTER 10 :**  10.1CONCLUSION  10.2 REFERENCES |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **NAME OF THE FIGURE** | **PAGE NO.** |
|  |  |  |
| 4.1 | Use case Diagram |  |
| 4.2 | Class diagram |  |
| 4.3 | Object diagram |  |
| 4.4 | State Diagram |  |
| 4.5 | Activity Diagram |  |
| 4.6 | Sequence diagram |  |
| 4.7 | Collaboration diagram |  |
| 4.8 | Component Diagram |  |
| 4.9 | Data flow diagram |  |
| 4.10 | Deployment Diagram |  |
| 4.11 | Architecture Diagram |  |

**LIST OF SYSMBOLS**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **NOTATION**  **NAME** | **NOTATION** | **DESCRIPTION** |
| 1. | Class | *Class Name*  *-attribute*  *-attribute*  *+operation*  *+operation*  *+operation*  *+ public*  *-private*  *# protected* | Represents a collection of similar entities grouped together. |
| 2. | Association | name  Class B  Class A  Class A  Class B | Associations represents static relationships between classes. Roles represents the way the two classes see each other. |
| 3. | Actor | Class A  Class A  Class B  Class B | It aggregates several classes into a single classes. |
| 4. | Aggregation | Interaction between the system and external environment |

|  |  |  |  |
| --- | --- | --- | --- |
| 5. | Relation  (uses) | uses | Used for additional process communication. |
| 6. | Relation  (extends) | extends | Extends relationship is used when one use case is similar to another use case but does a bit more. |
| 7. | Communication |  | Communication between various use cases. |
| 8. | State | State | State of the processes. |
| 9. | Initial State |  | Initial state of the object |
| 10. | Final state |  | Final state of the object |
| 11. | Control flow |  | Represents various control flow between the states. |
| 12. | Decision box |  | Represents decision making process from a constraint |
| 13. | Use case |  | Interact ion between the system and external environment. |

|  |  |  |  |
| --- | --- | --- | --- |
| 14. | Component |  | Represents physical modules which are a collection of components. |
| 15. | Node |  | Represents physical modules which are a collection of components. |
| 16. | Data Process/State |  | A circle in DFD represents a state or process which has been triggered due to some event or action. |
| 17. | External entity |  | Represents external entities such as keyboard, sensors, etc. |
| 18. | Transition |  | Represents communication that occurs between processes. |
| 19. | Object Lifeline |  | Represents the vertical dimensions that the object communications. |
| 20. | Message | Message | Represents the message exchanged. |

**CHAPTER-1**

**INTRODUCTION**

A lung disease prediction using an x-ray images project is all about detecting whether the given image has any lung disease or not. Here, a deep learning model and VGG16 model are implemented to predict the presence or absence of lung disease from the given image by utilizing various libraries in Python like NumPy, TensorFlow, etc. With a test accuracy percentage of 91%, the project was finished successfully. Machine learning is a branch of artificial intelligence in which computers use a range of statistical, probabilistic, and optimization approaches to "learn" from prior examples and detect difficult-to-find patterns in vast, noisy, or complex data sets. Machine learning is a key technique for developing powerful, automated, and objective algorithms for analyzing high-dimensional, multimodal biological data. In medical systems, machine learning plays a critical role. Disease detection can be added, allowing us to detect illnesses earlier and more correctly, perhaps saving many lives and reducing system load. One of the primary causes of mortality is lung disease. The diagnosis and prediction of lung illness have become a need in research since it can help with patient care later on. Doctors can use Machine Learning-based decision support systems to help them with their diagnosis decisions. Patients' respiratory issues, as well as Corona, Tuberculosis, Pneumonia, and Lung Cancer, were all examined in the study. To review data and develop models for diagnosing patients, machine learning and deep learning are used. Combining patient data with data from chest X-rays, utilizing VGG16 with the well-known pre-trained model, and VGG16 for data in this fashion were among the approaches utilized in this study to diagnose lung disorders. Deep Learning were utilized to evaluate Project involves how lung disease prediction using x-ray images will predict through the binary classification model implemented, and various python libraries like Tensor Flow, Keras, NumPy, etc. are used. This research project will observe the prediction of lung diseases by using x-ray images and further the output will be predicted with the detailed example and a detailed source code. The implementation will be shown step by step.

**1.2 SCOPE OF THE PROJECT**

The scope of this lung disease prediction project using x-ray images is to provide an accurate and reliable system for early detection and diagnosis of lung diseases through the use of deep learning and machine learning techniques. This project aims to help medical professionals and clinicians in their decision-making process and improve patient care. The project also emphasizes the importance of addressing the growing incidence of lung diseases globally, especially in developing and low-income nations where the risk is high.

**1.3 OBJECTIVE**

The objective of this project is to develop a lung disease prediction model that can accurately classify X-ray images as either healthy or diseased, using a combination of machine learning and deep learning techniques. The model should be able to identify different types of lung diseases, such as pneumonia, tuberculosis, and lung cancer, with high accuracy and precision. Additionally, the project aims to create a user-friendly interface that can be used by healthcare professionals and clinicians to facilitate early diagnosis and treatment of lung diseases. The performance of the model will be evaluated based on various metrics, including sensitivity, specificity, and area under the curve (AUC), to ensure its reliability and validity. The ultimate goal of this project is to improve patient outcomes by providing accurate and timely diagnosis of lung diseases.

**1.4 EXISTING SYSTEM:**

* Chest imaging diagnostics is crucial in the medical area due to many serious lung diseases like cancers and nodules and particularly with the current pandemic of Covid-19. Machine learning approaches yield prominent results toward the task of diagnosis. Recently, deep learning methods are utilized and recommended by many studies in this domain.
* The research aims to critically examine the newest lung disease detection procedures using deep learning algorithms that use X-ray and CT scan datasets. However, different architectures of deep learning are used by many researchers but, Convolutional Neural Networks (CNN) are still state-of-art techniques in dealing with image datasets.

**1.4.1 EXISTINGSYSTEM DISADVANTAGES:**

* CNN do not encode the position and orientation of object.
* Lack of ability to be spatially invariant to the input data. Lots of training data is required

**1.5 LITERATURE SURVEY**

**Title:** Prediction of Pneumonia Using Big Data, Deep Learning, and Machine Learning Techniques

**Author:** K.R. Swetha; Niranjanamurthy M; Amulya M P; Manu Y M

**Year:** 2021.

**Description:** Using big data for prediction analysis along with machine learning or deep learning techniques or algorithms is one the most active areas of research in order to improve the health and the medical science. There is a significant increase in the size of the medical data as well as the complexity in the diagnosis of various diseases. With this being said, the diagnosis or the prediction of many terminal or fatal diseases has seen huge success through deep learning. Among those fatal diseases, pneumonia is one of the greatest threats to the life of a man affecting the lungs leading to lung failure. To diagnose a man with pneumonia, the x-ray of chest is needed, and an expert in the prediction is also required. Hence, it is more convenient to build an automated predictor to predict the pneumonia using the big data deep learning methods. Among all the other techniques, CNN (Convolutional Neural Networks) stand tall and high in this prediction along with other classifiers. Also, pre-training the CNN models for very large datasets that is for big data of healthcare units stands a high chance for accurate classification. A CNN model which is pre-trained along with an efficient feature extraction technique and various classifiers to classify the positive from negative is considered to give highly accurate results. This research work represents the Prediction of Pneumonia using Big Data, Deep Learning and Machine Learning Techniques.

**Title:** COVID-19 Chest X-Ray Image Classification Using Deep Learning

**Author:** Dey, Soumava, Gunther Correia Bacellar, Mallikarjuna Basappa

Chandrappa, and Raj Kulkarni

**Year:** 2021.

**Description**: The rise of the coronavirus disease 2019 (COVID-19) pandemic has made it necessary to improve existing medical screening and clinical management of this disease. While COVID-19 patients are known to exhibit a variety of symptoms, the major symptoms include fever, cough, and fatigue. Since these symptoms also appear in pneumonia patients, this creates complications in COVID-19 detection especially during the flu season. Early studies identified abnormalities in chest X-ray images of COVID-19 infected patients that could be beneficial for disease diagnosis. Therefore, chest X-ray image-based disease classification has emerged as an alternative to aid medical diagnosis. However, manual detection of COVID-19 from a set of chest X-ray images comprising both COVID-19 and pneumonia cases is cumbersome and prone to human error. Thus, artificial intelligence techniques powered by deep learning algorithms, which learn from radiography images and predict presence of COVID-19 have potential to enhance current diagnosis process. Towards this purpose, here we implemented a set of deep learning pre-trained models such as ResNet, VGG, Inception and EfficientNet in conjunction with developing a computer vision AI system based on our own convolutional neural network (CNN) model: Deep Learning in Healthcare (DLH)-COVID. All these CNN models cater to image classification exercise. We used publicly available resources of 6,432 images and further strengthened our model by tuning hyperparameters to provide better generalization during the model validation phase. Our final DLH-COVID model yielded the highest accuracy of 96% in detection of COVID-19 from chest X-ray images when compared to images of both pneumonia-affected and healthy individuals. Given the practicality of acquiring chest X-ray images by patients, we also developed a web application (link: https://toad.li/xray) based on our model to directly enable users to upload chest X-ray images and detect the presence of COVID-19 within a few seconds. Taken together, here we introduce a state-of-the-art artificial intelligence-based system for efficient COVID-19 detection and a user-friendly application that has the capacity to become a rapid COVID-19 diagnosis method in the near future.

**Title:** Prediction of Pneumonia Using Big Data, Deep Learning, and Machine Learning Techniques

**Author:** K.R. Swetha; Niranjanamurthy M; Amulya M P; Manu Y M

**Year:** 2021.

**Description:** Using big data for prediction analysis along with machine learning or deep learning techniques or algorithms is one the most active areas of research in order to improve the health and the medical science. There is a significant increase in the size of the medical data as well as the complexity in the diagnosis of various diseases. With this being said, the diagnosis or the prediction of many terminal or fatal diseases has seen huge success through deep learning. Among those fatal diseases, pneumonia is one of the greatest threats to the life of a man affecting the lungs leading to lung failure. To diagnose a man with pneumonia, the x-ray of chest is needed, and an expert in the prediction is also required. Hence, it is more convenient to build an automated predictor to predict the pneumonia using the big data deep learning methods. Among all the other techniques, CNN (Convolutional Neural Networks) stand tall and high in this prediction along with other classifiers. Also, pre-training the CNN models for very large datasets that is for big data of healthcare units stands a high chance for accurate classification. A CNN model which is pre-trained along with an efficient feature extraction technique and various classifiers to classify the positive from negative is considered to give highly accurate results. This research work represents the Prediction of Pneumonia using Big Data, Deep Learning and Machine Learning Techniques.

**Title:**  Classification of COVID-19 CT Images using Transfer Learning Models

**Author:** Patil, Swati, and Akshay Golellu

**Year:** 2021

**Description**: Background and objective: SAARS-COV-2 is a respiratory illness caused by the novel Coronavirus (COVID-19) disease. The virus goes into the lungs through the respiratory tracks and damages the walls and linings of the air sacs in our lungs, as our body tries to fight it, our lungs become more inflamed and fill with fluid. This makes it harder to breathe. So, at early stages, deep learning applications can be used for screening and prediction at a rapid rate for diagnosing the lungs of patients. This paper uses Transfer learning methods. Four pretrained models were used in this study - VGG-16, VGG-19, Inceptionv3, Xception This paper addresses challenges while using pre-trained models in real-world. Also, high accuracies were achieved on these models.

**Title:** FVC-NET: An Automated Diagnosis of Pulmonary Fibrosis Progression Predict ion Using Honeycombing and Deep Learning

**Author**: Yadav, Anju, et al.

**Year:** 2022**.**

**Description:** Pulmonary fibrosis is a severe chronic lung disease that causes irreversible scarring in the tissues of the lungs, which results in the loss of lung capacity. The Forced Vital Capacity (FVC) of the patient is an interesting measure to investigate this disease to have the prognosis of the disease. This paper proposes a deep learning-based FVC-Net architecture to predict the progression of the disease from the patient’s computed tomography (CT) scan and the patient’s metadata. The input to the model combines the image score generated based on the degree of honeycombing for a patient identified based on segmented lung images and the metadata. This input is then fed to a 3-layer net to obtain the final output. The performance of the proposed FVC-Net model is compared with various contemporary state-of-the-art deep learning-based models, which are available on a cohort from the pulmonary fibrosis progression dataset. The model showcased significant improvement in the performance over other models for modified Laplace Log-Likelihood (−6.64). Finally, the paper concludes with some prospects to be explored in the proposed study.

**1.6 PROPOSED SYSTEM**

* There are many existing models which predict different diseases individually but we aim to predict different diseases in a single model with the highest accuracy. we have collected all the data from the Kaggle dataset, that data will be tested and then we will train all the classes.
* Then we will train the model step by step to complete the process, at the sequential process of model summary done layer by layer. So that we have created the VGG16 Architecture to train and test our multi Lung diseases like Covid-19, Pneumonia & Tuberculosis through their related x-ray images.
* Our model give a Training accuracy of 93.1% and testing accuracy of 91.1% in finding/ predicting lung diseases.

**1.6.1 PROPOSED SYSTEM ADVANTAGES:**

* It is one of the popular algorithms for image classification.
* It is easy to use with transfer learning.
* The small-size convolution filters allows VGG to have a large number of weight layers, more layers leads to improved performance

**CHAPTER 2**

**PROJECT DESCRIPTION**

**2.1 GENERAL:**

This project aims to develop a machine learning model using deep learning techniques to predict lung diseases from x-ray images. The model will be trained on a large dataset of x-ray images and will be able to detect the presence or absence of lung diseases accurately. The project will utilize Python libraries such as TensorFlow and Keras to implement the deep learning model.

**2.2 METHODOLOGIES**

**2.2.1MODULES NAME:**

* **Dataset**
* **Importing the necessary libraries**
* **Retrieving the images**
* **Splitting the dataset**
* **Building the model**
* **Apply the model and plot the graphs for accuracy and loss**
* **Accuracy on test set**
* **Saving the Trained Model**

**2.2.2 MODULES DESCSRIPTION:**

**1) Dataset:**

In the first module, we develop the system to get the input dataset for the training and testing purpose. Dataset is given in the model folder. The dataset consists of 1075 Lungs X-ray images.

**2) Importing the necessary libraries:**

We will be using Python language for this. First we will import the necessary libraries such as keras for building the main model, sklearn for splitting the training and test data, PIL for converting the images into array of numbers and other libraries such as pandas, numpy, matplotlib and tensorflow.

**3) Retrieving the images:**

We will retrieve the images and their labels. Then resize the images to (180,180) as all images should have same size for recognition. Then convert the images into numpy array.

**4) Splitting the dataset:**

Split the dataset into train and test. 80% train data and 20% test data.

**5) Building the model:**

The concept of convolutional neural networks. They are very successful in image recognition. The key part to understand, which distinguishes CNN from traditional neural networks, is the convolution operation. Having an image at the input, CNN scans it many times to look for certain features. This scanning (convolution) can be set with 2 main parameters: stride and padding type. As we see on below picture, process of the first convolution gives us a set of new frames, shown here in the second column (layer). Each frame contains an information about one feature and its presence in scanned image. Resulting frame will have larger values in places where a feature is strongly visible and lower values where there are no or little such features. Afterwards, the process is repeated for each of obtained frames for a chosen number of times. In this project I chose a classic VGG16  model which contains only two convolution layers.

The latter layer we are convolving, the more high-level features are being searched. It works similarly to human perception. To give an example, below is a very descriptive picture with features which are searched on different CNN layers. As you can see, the application of this model is face recognition. You may ask how the model knows which features to seek. If you construct the CNN from the beginning, searched features are random. Then, during training process, weights between neurons are being adjusted and slowly CNN starts to find such features which enable to meet predefined goal, i.e. to recognize successfully images from the training set.

Between described layers there are also pooling (sub-sampling) operations which reduce dimensions of resulted frames. Furthermore, after each convolution we apply a non-linear function (called **ReLU**) to the resulted frame to introduce non-linearity to the model.

Eventually, there are also fully connected layers at the end of the network. The last set of frames obtained from convolution operations is flattened to get a one-dimensional vector of neurons. From this point we put a standard, fully-connected neural network. At the very end, for classification problems, there is a softmax layer. It transforms results of the model to probabilities of a correct guess of each class

**6) Apply the model and plot the graphs for accuracy and loss:**

We will compile the model and apply it using fit function. The batch size will be 10. Then we will plot the graphs for accuracy and loss. We got average validation accuracy of 93.00% and average training accuracy of 91.00%.

**7) Accuracy on test set:**

We got an accuracy of 91.00% on test set.

**8) Saving the Trained Model:**

Once you’re confident enough to take your trained and tested model into the production-ready environment, the first step is to save it into a .h5 or .pkl file using a library like pickle.

Make sure you have pickle installed in your environment.

Next, let’s import the module and dump the model into .h5 file.

**2.3 TECHNIQUE USED OR ALGORITHM USED**

**2.3.1 EXISTING TECHNIQUE: -**

* **Convolutional Neural Networks (CNN)**
* A CNN is a kind of network architecture for deep learning algorithms and is specifically used for image recognition and tasks that involve the processing of pixel data. There are other types of neural networks in deep learning. Neural networks accept an input image/feature vector (one input node for each entry) and transform it through a series of hidden layers, commonly using nonlinear activation functions.
* Each hidden layer is also made up of a set of neurons, where each neuron is fully connected to all neurons in the previous layer. The last layer of a neural network (i.e., the “output layer”) is also fully connected and represents the final output classifications of the network.

**2.3.2 PROPOSED TECHNIQUE USED OR ALGORITHM USED:**

* **VGG16 Architecture**
* VGG16 is a type of CNN (Convolutional Neural Network) that is considered to be one of the best computer vision models to date. The creators of this model evaluated the networks and increased the depth using an architecture with very small (3 × 3) convolution filters, which showed a significant improvement on the prior-art configurations.
* We pushed the depth to 16–19 weight layers making it approx — 138 trainable parameters. VGG16 is object detection and classification algorithm which is able to classify 1000 images of 1000 different categories with high accuracy.

.

**CHAPTER 3**

**REQUIREMENTS ENGINEERING**

**3.1 GENERAL**

We can see from the results that on each database, the error rates are very low due to the discriminatory power of features and the regression capabilities of classifiers. Comparing the highest accuracies (corresponding to the lowest error rates) to those of previous works, our results are very competitive.

**3.2 HARDWARE REQUIREMENTS**

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should what the system do and not how it should be implemented.

* PROCESSOR : DUAL CORE 2 DUOS.
* RAM : 4GB DD RAM
* HARD DISK : 250 GB

**3.3 SOFTWARE REQUIREMENTS**

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team’s progress throughout the development activity.

* Operating System : Windows 7/8/10
* Platform : Spyder3
* Programming Language : Python
* Front End : HTML,CSS

**3.4 FUNCTIONAL REQUIREMENTS**

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behavior, Firstly, the system is the first that achieves the standard notion of semantic security for data confidentiality in attribute-based deduplication systems by resorting to the hybrid cloud architecture.

**3.5 NON-FUNCTIONAL REQUIREMENTS**

**The major non-functional Requirements of the system are as follows**

**Usability**

The system is designed with completely automated process hence there is no or less user intervention.

**Reliability**

The system is more reliable because of the qualities that are inherited from the chosen platform python. The code built by using python is more reliable.

**Performance**

This system is developing in the high level languages and using the advanced back-end technologies it will give response to the end user on client system with in very less time.

**Supportability**

The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform, which is built into the system.

**Implementation**

The system is implemented in web environment using Jupyter notebook software. The server is used as the intellignce server and windows 10 professional is used as the platform. Interface the user interface is based on FLASK provides server system.

**CHAPTER 4**

**DESIGN ENGINEERING**

**4.1 GENERAL**

Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering.

**4.2 UML DIAGRAMS**

**4.2.1 USE CASE DIAGRAM**



**EXPLANATION:**

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. The above diagram consists of user as actor. Each will play a certain role to achieve the concept.

**4.2.2 CLASS DIAGRAM**

****

**EXPLANATION**

In this class diagram represents how the classes with attributes and methods are linked together to perform the verification with security. From the above diagram shown the various classes involved in our project.

**4.2.3 OBJECT DIAGRAM**



**EXPLANATION:**

In the above digram tells about the flow of objects between the classes. It is a diagram that shows a complete or partial view of the structure of a modeled system. In this object diagram represents how the classes with attributes and methods are linked together to perform the verification with security.

**4.2.4 STATE DIAGRAM**

****

**EXPLANATION:**

State diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.

**4.2.5 ACTIVITY DIAGRAM**

****

**EXPLANATION:**

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.

**4.2.6 SEQUENCE DIAGRAM**

****

**EXPLANATION:**

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. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

**4.2.7 COLLABORATION DIAGRAM**



**EXPLANATION:**

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

**4.2.8 COMPONENT DIAGRAM**



**EXPLANATION**

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators. All boxes are components and arrow indicates dependencies.

**4.2.9 Data Flow Diagram**

**Level 0:**

Read Images & Pre-processing

Apply VGG16 Architecture & save trained file like TrainedModel.h5

Lung Disease Dataset

User

**Level 1:**

Create VGG16 Architecture using tensor-flow layers

Understand about layers

Import necessary layers from tensorflow

User

**Level 2:**

user/ patient Chest X-ray Image’s input

Load related TrainedModel.h5 & apply converted data

Lung Disease detection

Array conversion, expanding dimensions & resize images

**4.2.10 DEPLOYMENT DIAGRAM**

****

**EXPLANATION:**

Deployment Diagram is a type of diagram that specifies the physical hardware on which the software system will execute. It also determines how the software is deployed on the underlying hardware. It maps software pieces of a system to the device that are going to execute it.

**SYSTEM ARCHITECTURE:**

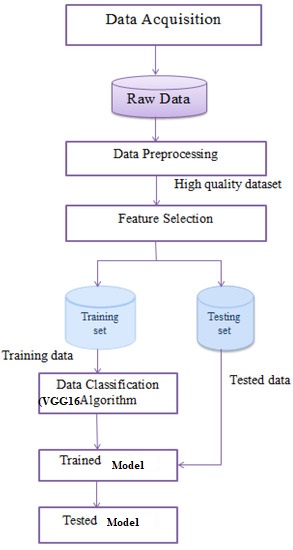
****

Fig 4.11: System Architecture

**CHAPTER 5**

**DEVELOPMENT TOOLS**

**5.1 Python**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

## 5.2 History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

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

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

#### 5.3 Importance of Python

* **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 is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

#### 5.4 Features of Python

* **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
* **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases** − Python provides interfaces to all major commercial databases.
* **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

* It supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* IT supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

**5.5 Libraries used in python**

* numpy - mainly useful for its N-dimensional array objects.
* pandas - Python data analysis library, including structures such as dataframes.
* matplotlib - 2D plotting library producing publication quality figures.
* scikit-learn - the machine learning algorithms used for data analysis and data mining tasks.



Figure : NumPy, Pandas, Matplotlib, Scikit-learn

**CHAPTER 6**

**IMPLEMENTATION**

**6.1 GENERAL**

**Coding:**

**CHAPTER 7**

**SNAPSHOTS**

**General:**

This project is implements like application using python and the Server process is maintained using the SOCKET & SERVERSOCKET and the Design part is played by Cascading Style Sheet.

**SNAPSHOTS**

**CHAPTER 8**

**SOFTWARE TESTING**

**8.1 GENERAL**

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.

**8.2 DEVELOPING METHODOLOGIES**

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

**8.3Types of Tests**

**8.3.1 Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input 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.

**8.3.2 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.

**8.3.3 System Test**

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.

**8.3.4 Performance Test**

The Performance test ensures that the output be produced within the time limits,and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

**8.3.5 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.

**8.3.6 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.

**Acceptance testing for Data Synchronization:**

* The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node
* The Route add operation is done only when there is a Route request in need
* The Status of Nodes information is done automatically in the Cache Updation process

**8.2.7 Build the test plan**

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

**CHAPTER 9**

**FUTURE ENHANCEMENT**

**9.1 FUTURE ENHANCEMENTS:**

In the future, we hope to conduct a training with more data sets and change some parameters to faster the model. Some metric parameters of the metrics will also be tested. We can experiment on pre-trained model to improve the accuracy

**CHAPTER 10**

**CONCLUSIONAND REFERENCES**

**10.1 CONCLUSION**

In this project, the effect of the lungs of a modern patient on the various researchers and the damage to the lung is clearly explained by various researchers. Since these lung diseases have been cured the necessity of identifying this disease has become essential according to many researches. One of the main concerns of this research is to identify and select a proper data sets and technique to analyze lung diseases. Chest x-ray was selected based on the comparisons and discussions that were stated in this paper. Next a proper and suitable feature extraction algorithm was chosen since the chest x-ray may contain lots of unnecessary data. This selection was based on advantages and disadvantages of using many common algorithms. Finally, a classification algorithm was also discussed based on their characteristic qualities. In short-term research, it was seen that VGG16 Architecture added additional benefits to predict the lung diseases in advance with better results. Ultimately, lung disease can be diagnosed.

**10.2 REFERENCES**

[1] Tensor Flow's main website, <https://www.tensorflow.org/>.

[2] Swetha, K. R., et al. "Predict ion of Pneumonia Using Big Data, Deep Learning, and Machine Learning Techniques." 2021 6th International Conference on Communicat ion and Electronics Systems (ICCES). IEEE, 2021.

[3] Dey, Soumava, Gunther Correia Bacellar, Mallikarjuna Basappa Chandrappa, and Raj Kulkarni. "COVID-19 Chest X-Ray Image Classificat ion Using Deep Learning." medRxiv (2021).

[4]Tensorflowlosses Module, https://www.tensorflow.org/versions/r1.5/api\_docs/python/tf/losses

[5] CNN, ht tps://en.wikipedia.org/wiki/Convolutional\_neural\_network.

[6]HyperParameters,https://en.wikipedia.org/wiki/Convolutional\_neural\_network#Hyperparameters

[7] Swetha, K. R., et al. "Predict ion of Pneumonia Using Big Data, Deep Learning, and Machine Learning Techniques." 2021 6th International Conference on Communication and Electronics Systems (ICCES). IEEE, 2021.

[8] Patil, Swat i, and Akshay Golellu. "Classification of COVID-19 CT Images using Transfer Learning Models." 2021 International Conference on Emerging Smart Computing and Informatics (ESCI). IEEE, 2021.

[9] Yadav, Anju, et al. "FVC-NET: An Automated Diagnosis of Pulmonary Fibrosis Progression Prediction Using Honeycombing and Deep Learning." Computational Intelligence and Neuroscience 2022 (2022).

[10] Kogilavani, S. V., et al. "COVID-19 detect ion based on lung CT scan using deep learning techniques." Computational and Mathematical Methods in Medicine 2022 (2022).

[11] Aggarwal, Karan, et al. "Has the Future Started? The Current Growth of Artificial Intelligence, Machine Learning, and Deep Learning." Iraqi Journal For Computer Science and Mathematics 3.1 (2022): 115-123.

[12] Ahmed, Shaymaa Taha, and Suhad Malallah Kadhem. "Using Machine Learning via Deep Learning Algorithms to Diagnose the Lung Disease Based on Chest Imaging: A Survey." International Journal of Interactive Mobile Technologies 15.16 (2021).

[13] Gowri, S., J. Jabez, J. S. Vimali, A. Sivasangari, and Senduru Srinivasulu. "Sentiment Analysis of Twitter Data Using Techniques in Deep Learning." In Data Intelligence and Cognitive Informatics, pp. 613-623. Springer, Singapore, 2021.

[14] Srinivasulu, Senduru, Jeberson Retna Raj, and S. Gowri. "Analysis and Predict ion of Myocardial Infarct ion using Machine Learning." 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI). IEEE, 2021.

[15] Priya, T., and T. Meyyappan. "Disease Predict ion by Machine Learning Over Big Data Lung Cancer." International Journal of Scientific Research in Computer Science, Engineering and Information Technology (2021): 16-24.

[16] Battineni, Gopi. "Machine Learning and Deep Learning Algorithms in the Diagnosis of Chronic Diseases." Machine Learning Approaches for Urban Computing. Springer, Singapore, 2021. 141-164.