**ABSTRACT**

In this project, we will be using Deep Learning algorithm and machine learning algorithm such as Random Forest Classifier and yolov7- model to detect and classification of chest xray abnormalities. The datasets will be collected from VINDR-CXR dataset. Dataset Augmentation is used to increase the amount of datasets. After augmentation, Image Resizing technique is used for dataset preprocessing. Feature extraction methods such as InceptionV3, Xception, InceptionResNetV2 and NASNetLarge under transfer learning are used to extract the features. Then, it will be trained using machine learning algorithm such as Random Forest Classifier and deep learning algorithm such as yolov7- model to effectively determine the lung disease. When an input image is given for prediction process, it can easily predict and classifies the lung disease such as Atelectasis, Emphysema, Lung Opacity, Lung cavity etc. A web application is developed for providing input image for detecting and displaying output results.

**TECHNOLOGIES USED:**

**1.2.1 Deep Learning**

Deep learning is a computer software that **mimics the network of neurons in a brain**. It is a subset of machine learning and is called deep learning because it makes use of deep **neural networks.**

Deep learning algorithms are constructed with connected layers.

* The first layer is called the Input Layer
* The last layer is called the Output Layer
* All layers in between are called Hidden Layers. The word deep means the network join neurons in more than two layers.

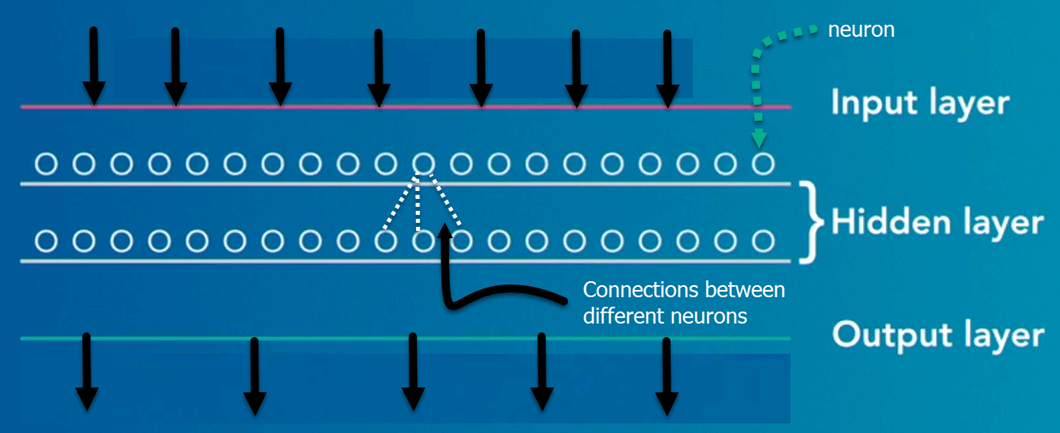
[](https://www.guru99.com/images/tensorflow/083018_0542_WhatisDeepl1.png)

Figure 1.1 Deep Learning Layers

Each Hidden layer is composed of neurons. The neurons are connected to each other. The neuron will process and then propagate the input signal it receives the layer above it. The strength of the signal given the neuron in the next layer depends on the weight, bias and activation function.

The network consumes large amounts of input data and operates them through multiple layers; the network can learn increasingly complex features of the data at each layer.

## **IMPORTANCE OF DEEP LEARNING**

Deep learning is a powerful tool to make prediction an actionable result. Deep learning excels in pattern discovery (unsupervised learning) and knowledge-based prediction. Big data is the fuel for deep learning. When both are combined, an organization can reap unprecedented results in term of productivity, sales, management, and innovation.

Deep learning can outperform traditional method. For instance, deep learning algorithms are 41% more accurate than machine learning algorithm in image classification, 27 % more accurate in facial recognition and 25% in voice recognition.

## **DEEP LEARNING PROCESS**

A deep neural network provides state-of-the-art accuracy in many tasks, from object detection to speech recognition. They can learn automatically, without predefined knowledge explicitly coded by the programmers.

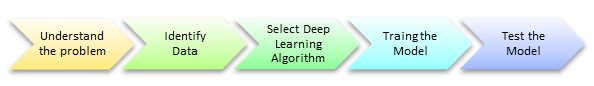
[](https://www.guru99.com/images/tensorflow/083018_0542_WhatisDeepl2.png)

Figure 1.2 Deep Learning Process

To grasp the idea of deep learning, imagine a family, with an infant and parents. The toddler points objects with his little finger and always says the word 'cat.' As its parents are concerned about his education, they keep telling him 'Yes, that is a cat' or 'No, that is not a cat.' The infant persists in pointing objects but becomes more accurate with 'cats.' The little kid, deep down, does not know why he can say it is a cat or not. He has just learned how to hierarchies’ complex features coming up with a cat by looking at the pet overall and continue to focus on details such as the tails or the nose before to make up his mind.

A neural network works quite the same. Each layer represents a deeper level of knowledge, i.e., the hierarchy of knowledge. A neural network with four layers will learn more complex feature than with that with two layers.

The learning occurs in two phases.

* The first phase consists of applying a nonlinear transformation of the input and create a statistical model as output.
* The second phase aims at improving the model with a mathematical method known as derivative.

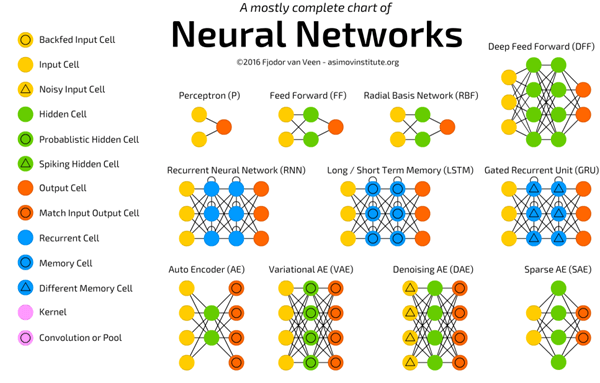
The neural network repeats these two phases hundreds to thousands of time until it has reached a tolerable level of accuracy. The repeat of this two-phase is called an iteration.

## **CLASSIFICATION OF NEURAL NETWORKS**

**Shallow neural network**: The Shallow neural network has only one hidden layer between the input and output.

**Deep neural network**: Deep neural networks have more than one layer. For instance, Google LeNet model for image recognition counts 22 layers.

Nowadays, deep learning is used in many ways like a driverless car, mobile phone, Google Search Engine, Fraud detection, TV, and so on.

[](https://www.guru99.com/images/tensorflow/083018_0542_WhatisDeepl3.png)

## Figure 1.3 Types of Deep Learning Networks

## **FEED-FORWARD NEURAL NETWORKS**

The simplest type of artificial neural network. With this type of architecture, information flows in only one direction, forward. It means, the information's flows starts at the input layer, goes to the "hidden" layers, and end at the output layer. The network does not have a loop. Information stops at the output layers.

## **RECURRENT NEURAL NETWORKS (RNNS)**

RNN is a multi-layered neural network that can store information in context nodes, allowing it to learn data sequences and output a number or another sequence. In simple words it an Artificial neural networks whose connections between neurons include loops. RNNs are well suited for processing sequences of inputs.

* The RNN neurons will receive a signal that point to the start of the sentence.
* The network receives the word "Do" as an input and produces a vector of the number. This vector is fed back to the neuron to provide a memory to the network. This stage helps the network to remember it received "Do" and it received it in the first position.
* The network will similarly proceed to the next words. It takes the word "you" and "want." The state of the neurons is updated upon receiving each word.
* The final stage occurs after receiving the word "a." The neural network will provide a probability for each English word that can be used to complete the sentence. A well-trained RNN probably assigns a high probability to "café," "drink," "burger," etc.

Common uses of RNN

* Help securities traders to generate analytic reports
* Detect abnormalities in the contract of financial statement
* Detect fraudulent credit-card transaction
* Provide a caption for images
* Power chatbots
* The standard uses of RNN occur when the practitioners are working with time-series data or sequences (e.g., audio recordings or text).

## **CONVOLUTIONAL NEURAL NETWORKS (CNN)**

CNN is a multi-layered neural network with a unique architecture designed to extract increasingly complex features of the data at each layer to determine the output. CNN's are well suited for perceptual tasks.

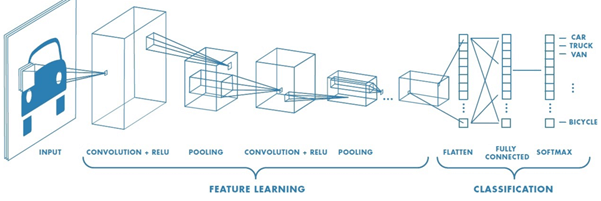
[](https://www.guru99.com/images/tensorflow/083018_0542_WhatisDeepl5.png)

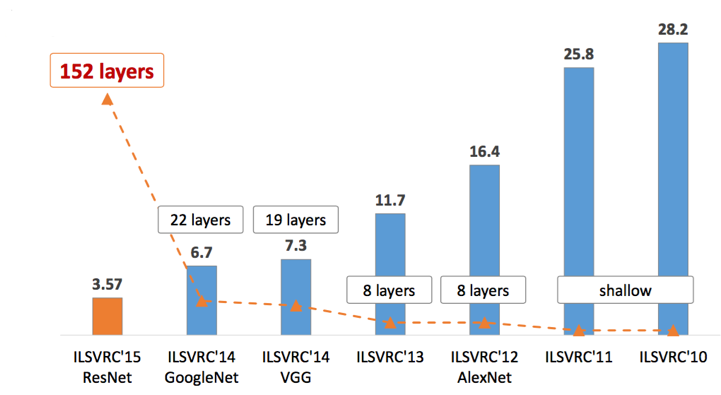
Figure 1.4 CNN

CNN is mostly used when there is an unstructured data set (e.g., images) and the practitioners need to extract information from it

For instance, if the task is to predict an image caption:

* The CNN receives an image of let's say a cat, this image, in computer term, is a collection of the pixel. Generally, one layer for the greyscale picture and three layers for a color picture.
* During the feature learning (i.e., hidden layers), the network will identify unique features, for instance, the tail of the cat, the ear, etc.
* When the network thoroughly learned how to recognize a picture, it can provide a probability for each image it knows. The label with the highest probability will become the prediction of the network.

**A Convolutional Neural Network** (**CNN**, or **ConvNet**) are a special kind of multi-layer neural networks, designed to recognize visual patterns directly from pixel images with minimal pre-processing. The **ImageNet** project is a large visual database designed for use in visual object recognition software research. The ImageNet project runs an annual software contest, the **ImageNet Large Scale Visual Recognition Challenge (**[**ILSVRC**](https://en.wikipedia.org/wiki/ImageNet#ImageNet_Challenge)**)**, where software programs compete to correctly classify and detect objects and scenes.



## Figure 1.5 Types of CNN

**1.2.2 Machine Learning**

**Machine Learning** is the field of study that gives computers the capability to learn without being explicitly programmed. ML is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that makes it more similar to humans: The ability to learn. Machine learning is actively being used today, perhaps in many more places than one would expect.

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.

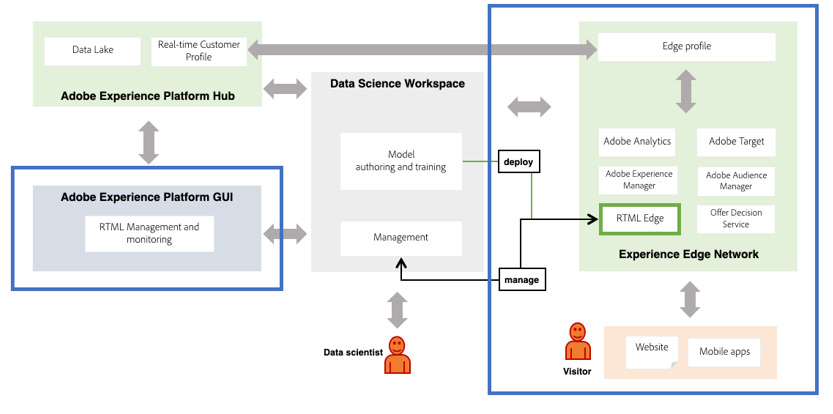


Figure 1.6 Machine learning architecture

**Machine Learning approaches**

Machine learning approaches are traditionally divided into three broad categories, depending on the nature of the "signal" or "feedback" available to the learning system:

[**Supervised learning**](https://en.wikipedia.org/wiki/Supervised_learning)**:**

Supervised learning as the name indicates the presence of a supervisor as a teacher. Basically, supervised learning is a learning in which we teach or train the machine using data which is well labelled that means some data is already tagged with the correct answer. After that, the machine is provided with a new set of examples (data) so that supervised learning algorithm analyses the training data (set of training examples) and produces a correct outcome from labelled data.

Supervised learning is where there are input variables (x) and an output variable (Y) and an algorithm is used to learn the mapping function from the input to the output.

Y = f(X)

The goal is to approximate the mapping function so well that when there is a new input data (x) that the output variables (Y) for that data can be predicted easily.

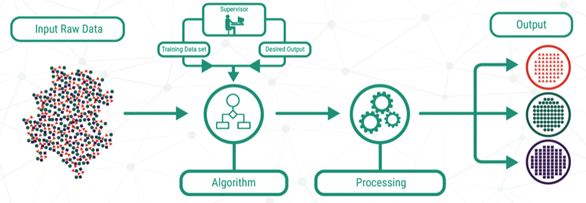


Figure 1.7 Supervised learning flowchart

**Unsupervised learning:**

Unsupervised learning is the training of machine using information that is neither classified nor labelled and allowing the algorithm to act on that information without guidance. Here the task of machine is to group unsorted information according to similarities, patterns and differences without any prior training of data.

Unlike supervised learning, no teacher is provided that means no training will be given to the machine. Therefore, machine is restricted to find the hidden structure in unlabelled data by itself.

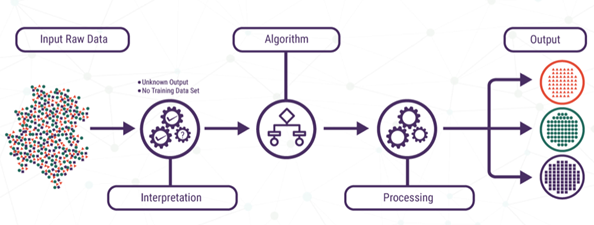


Figure 1.8 Unsupervised learning flowchart

[**Reinforcement learning**](https://en.wikipedia.org/wiki/Reinforcement_learning)**:**

A computer program interacts with a dynamic environment in which it must perform a certain goal. As it navigates its problem space, the program is provided feedback that's analogous to rewards, which it tries to maximize.



Figure 1.9 Reinforcement learning flowchart

* 1. **OBJECTIVES OF THE PROJECT:**
* To develop a method using Deep Learning and Machine Learning algorithms to effectively predict the lung disease.
* To helps in effective diagnosis of chest xray abnormalities with higher accuracy.
* To determine the cheap way to classify the cancer.
  1. **SCOPE OF THE PROJECT:**
* It can be used in all hospitals.
* It can be used in all laboratories.
* It can be used in day care centers.
* It can be used for test centers.

**CHAPTER - 3**

**PROPOSED SYSTEM**

**3.1 EXISTING SYSTEM**

Interpretation of chest radiographs (CXR) is a difficult but essential task for detecting thoracic abnormalities. Recent artificial intelligence (AI) algorithms have achieved radiologist-level performance on various medical classification tasks. However, only a few studies addressed the localization of abnormal findings from CXR scans, which is essential in explaining the image-level classification to radiologists. Additionally, the actual impact of AI algorithms on the diagnostic performance of radiologists in clinical practice remains relatively unclear. To bridge these gaps, we developed an explainable deep learning system called VinDr-CXR that can classify a CXR scan into multiple thoracic diseases and, at the same time, localize most types of critical findings on the image. VinDr-CXR was trained on 51,485 CXR scans with radiologist-provided bounding box annotations. It demonstrated a comparable performance to experienced radiologists in classifying 6 common thoracic diseases on a retrospective validation set of 3,000 CXR scans, with a mean area under the receiver operating characteristic curve (AUROC) of 0.967 (95% confidence interval [CI]: 0.958–0.975). The VinDr-CXR was also externally validated in independent patient cohorts and showed its robustness. For the localization task with 14 types of lesions, our free-response receiver operating characteristic (FROC) analysis showed that the VinDr-CXR achieved a sensitivity of 80.2% at the rate of 1.0 false-positive lesion identified per scan. A prospective study was also conducted to measure the clinical impact of the VinDr-CXR in assisting six experienced radiologists. The results indicated that the proposed system, when used as a diagnosis supporting tool, significantly improved the agreement between radiologists themselves with an increase of 1.5% in mean Fleiss’ Kappa. We also observed that, after the radiologists consulted VinDr-CXR’s suggestions, the agreement between each of them and the system was remarkably increased by 3.3% in mean Cohen’s Kappa. Altogether, our results highlight the potentials of the proposed deep learning system as an effective assistant to radiologists in clinical practice.

**3.2 DISADVANTAGES OF EXISTING SYSTEM**

* In the existing system, there is no clear evidence that the DLS helps improve the sensitivity or specificity of the radiologist in CXR interpretation.
* Further research is needed to validate the model prospectively and determine its utility in clinical settings.

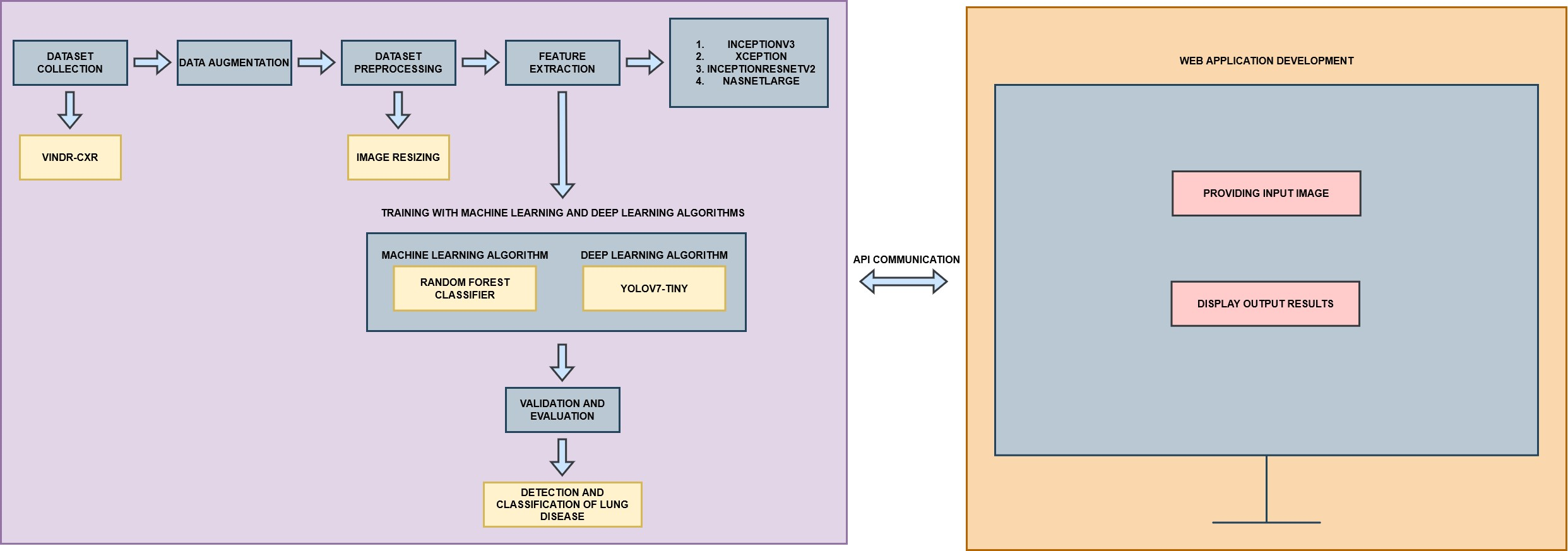
**3.3 PROPOSED SYSTEM**

In this project, we will be using Deep Learning algorithm and machine learning algorithm such as Random Forest Classifier and yolov7- model to detect and classification of chest xray abnormalities. The datasets will be collected from VINDR-CXR dataset. Dataset Augmentation is used to increase the amount of datasets. After augmentation, Image Resizing technique is used for dataset preprocessing. Feature extraction methods such as InceptionV3, Xception, InceptionResNetV2 and NASNetLarge under transfer learning are used to extract the features. Then, it will be trained using machine learning algorithm such as Random Forest Classifier and deep learning algorithm such as yolov7- model to effectively determine the lung disease. When an input image is given for prediction process, it can easily predict and classifies the lung disease such as Atelectasis, Emphysema, Lung Opacity, Lung cavity etc. A web application is developed for providing input image for detecting and displaying output results.

**3.4 ADVANTANGES OF PROPOSED SYSTEM**

* In this project we provide an efficient technique to classify using deep learning and machine learning algorithms.
* Easy prediction and analyses the classification of cancer.
* Saves time required for diagnosis of cancer

**3.5 SYSTEM ARCHITECTURE**

Figure 3.1 Proposed system architecture

**3.6 WORKING**

In this project, we will be using Deep Learning algorithm and machine learning algorithm such as Random Forest Classifier and yolov7- model to detect and classification of 28 different chest xray abnormalities at an early stage. So, the first step in the project will be collecting the dataset from VINDR-CXR dataset and then we will be separating these datasets into training as well as testing dataset where the testing dataset will be kept separate and the training dataset will be used to train the model. Then these datasets are pre-processed using image resizing is used to preprocessing the image datasets. Then dataset augmentation will be done which is used to increase the amount of datasets. After augmentation, Image Resizing technique is used for dataset preprocessing. Feature extraction methods such as InceptionV3, Xception, InceptionResNetV2 and NASNetLarge under transfer learning are used to extract the features. Then, it will be trained using machine learning algorithm such as Random Forest Classifier and deep learning algorithm such as yolov7- model to effectively determine the lung disease. When an input image is given for prediction process, it can easily predict and classifies the lung disease such as Atelectasis, Emphysema, Lung Opacity, Lung cavity etc. A web application is developed for providing input image for detecting and displaying output results.

**CHAPTER – 4**

**SYSTEM ANALYSIS**

**4.1 MODULE DESCRIPTION:**

* Dataset Collection
* Data Augmentation
* Dataset Preprocessing
* Feature Extraction
* Training using Machine Learning and Deep Learning algorithms
* Validation and Evaluation
* Detection and Classification of Lung Disease
* Web Application Development

**4.1.1 Dataset Collection**

In this project, we are going to collect the dataset and it will be fed for training with the deep learning algorithms. Increasing the amount of dataset increases the accuracy. For this project dataset is collected from VINDR-CXR dataset

A data set is a collection of data. Deep Learning has become the go-to method for solving many challenging real-world problems. It’s definitely by far the best performing method for computer vision tasks. The image above showcases the power of deep learning for computer vision. With enough training, a deep network can segment and identify the “key points” of every person in the image. These deep learning machines that have been working so well need fuel lots of fuel; that fuel is data. The more **labelled data** available, the better our model performs. The idea of more data leading to better performance has even been explored at a large-scale by Google with a dataset of 300 million images! When deploying a Deep Learning model in a real-world application, **data must be constantly fed**to continue improving its performance. And, in the deep learning era, data is very well arguably the most valuable resource. There are three steps of collecting data

**Scraping From the Web**

Manually finding and downloading images takes a long time simply due to the amount of human work involved. The task probably has some kind of common objects are to be detected. And so that becomes the keyword for web-scraping. It also becomes the class name for that object. Every single pixel in the image is required. To get those, it’s best to use some really great image annotation tools that are already out there. Given a rough set of polygon points around an object, can generate the pixel labels for segmentation. Deep extreme cut is also quite similar except they use only the four extreme points around the object. This will then give some nice bounding box and segmentation labels. Another option is to use an existing image annotation GUIs.

**Third-party:**

Since data has become such a valuable commodity in the deep learning era, many start-ups have started to offer their own image annotation services they’ll gather and label the data. Given a description of what kind of data and annotations needed. Mighty is one that has been doing self-driving car image annotation and has become pretty big in the space were at CVPR 2018 too. Payment AI are less specialized than Mighty AI, offering image annotation for any domain.

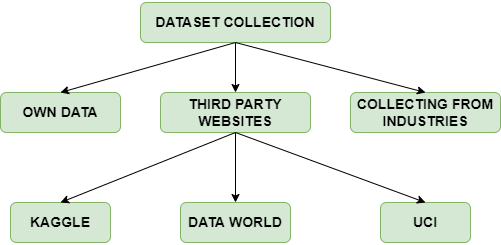
****

Figure 4.1 Dataset collection

**4.1.2 Data Augmentation**

Data augmentation in data analysis are techniques used to increase the amount of data by adding slightly modified copies of already existing data or newly created synthetic data from existing data. It acts as a regularizer and helps reduce overfitting when training a deep learning model. It is closely related to oversampling in data analysis.

Deep learning applications especially in deep learning domain continue to diversify and increase rapidly. Data augmentation techniques may be a good tool against challenges which artificial intelligence world faces. Data augmentation is useful to improve performance and outcomes of deep learning models by forming new and different examples to train datasets. If dataset in a deep learning model is rich and sufficient, the model performs better and more accurate. For deep learning models, collecting and labeling of data can be exhausting and costly processes. Transformations in datasets by using data augmentation techniques allow companies to reduce these operational costs.

One of the steps into a data model is cleaning data which is necessary for high accuracy models. However, if cleaning reduces the representability of data, then the model cannot provide good predictions for real world inputs. Data augmentation techniques enable deep learning models to be more robust by creating variations that the model may see in the real world.

Image data augmentation is perhaps the most well-known type of data augmentation and involves creating transformed versions of images in the training dataset that belong to the same class as the original image. Transforms include a range of operations from the field of image manipulation, such as shifts, flips, zooms, and much more.

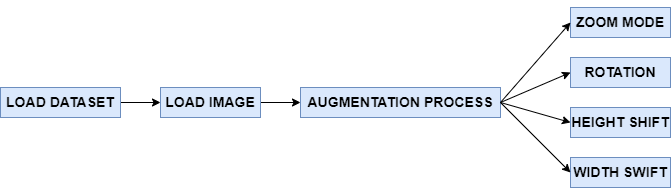
****

Figure 4.2 Dataset Augmentation

**4.1.3 Dataset Pre-processing**

In this project, Image Resizing technique is used for dataset preprocessing.

Deep learning has truly come into the mainstream in the past few years. Deep learning uses neural nets with a lot of hidden layers (dozens in today’s state of the art) and requires large amounts of training data. These models have been particularly effective in gaining insight and approaching human-level accuracy in perceptual tasks like vision, speech, language processing. The theory and mathematical foundations were laid several decades ago. Primarily two phenomena have contributed to the rise of machine learning a) Availability of huge data-sets/training examples in multiple domains and b) Advances in raw compute power and the rise of efficient parallel hardware.

Building an effective neural network model requires careful consideration of the network architecture as well as the input data format. The most common image data input parameters are the number of images, image height, image width, number of channels, and the number of levels per pixel. Typically, there are 3 channels of data corresponding to the colours Red, Green, Blue (RGB) Pixel levels are usually [0,255]. For this exercise let’s choose the following values

* number of images = 100
* image width, image height =100
* 3 channels, pixel levels in the range [0–255]

**Uniform aspect ratio:** One of the first steps is to ensure that the images have the same size and aspect ratio. Most of the neural network models assume a square shape input image, which means that each image needs to be checked if it is a square or not, and cropped appropriately. Cropping can be done to select a square part of the image, as shown. While cropping, we usually care about the part in the center.

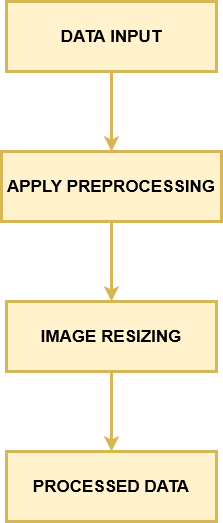


Figure 4.3 Dataset preprocessing

**4.1.4 Feature Extraction**

Feature extraction methods such as InceptionV3, Xception, InceptionResNetV2 and NASNetLarge under transfer learning are used to extract the features.

Transfer learning is the reuse of a pre-trained model on a new problem. It’s currently very popular in deep learning because it can train deep neural networks with comparatively little data. This is very useful in the data science field since most real-world problems typically do not have millions of labeled data points to train such complex models.

Transfer learning has several benefits, but the main advantages are saving training time, better performance of neural networks (in most cases), and not needing a lot of data. Usually, a lot of data is needed to train a neural network from scratch but access to that data isn't always available — this is where transfer learning comes in handy. With transfer learning a solid machine learning model can be built with comparatively little training data because the model is already pre-trained. This is especially valuable in natural language processing because mostly expert knowledge is required to create large labeled data sets. Additionally, training time is reduced because it can sometimes take days or even weeks to train a deep neural network from scratch on a complex task.

The reuse of a pre-trained model on a new problem is known as transfer learning in machine learning. A machine uses the knowledge learned from a prior assignment to increase prediction about a new task in transfer learning. You could, for example, use the information gained during training to distinguish beverages when training a classifier to predict whether an image contains cuisine. The knowledge of an already trained machine learning model is transferred to a different but closely linked problem throughout transfer learning.

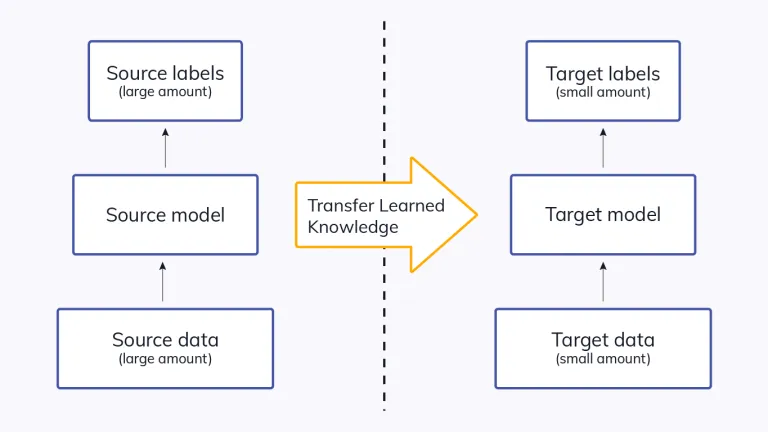


Figure 4.4 Feature extraction

**4.1.5 Training using Machine Learning and Deep Learning algorithms**

After feature extraction, it will be trained using machine learning algorithm such as Random Forest Classifier and deep learning algorithm such as yolov7- model to effectively determine the lung disease.

**4.1.5.1 Random Forest Classifier**

Random forest is a commonly-used machine learning algorithm trademarked by Leo Breiman and Adele Cutler, which combines the output of multiple decision trees to reach a single result. Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems.

The random forest algorithm is an extension of the bagging method as it utilizes both bagging and feature randomness to create an uncorrelated forest of decision trees. Feature randomness, also known as feature bagging or “the random subspace method”, generates a random subset of features, which ensures low correlation among decision trees. This is a key difference between decision trees and random forests. While decision trees consider all the possible feature splits, random forests only select a subset of those features.

Random forest algorithms have three main hyperparameters, which need to be set before training. These include node size, the number of trees, and the number of features sampled. From there, the random forest classifier can be used to solve for regression or classification problems. The random forest algorithm is made up of a collection of decision trees, and each tree in the ensemble is comprised of a data sample drawn from a training set with replacement, called the bootstrap sample. Of that training sample, one-third of it is set aside as test data, known as the out-of-bag (oob) sample. Since random forest can handle both regression and classification tasks with a high degree of accuracy, it is a popular method among data scientists. Feature bagging also makes the random forest classifier an effective tool for estimating missing values as it maintains accuracy when a portion of the data is missing.

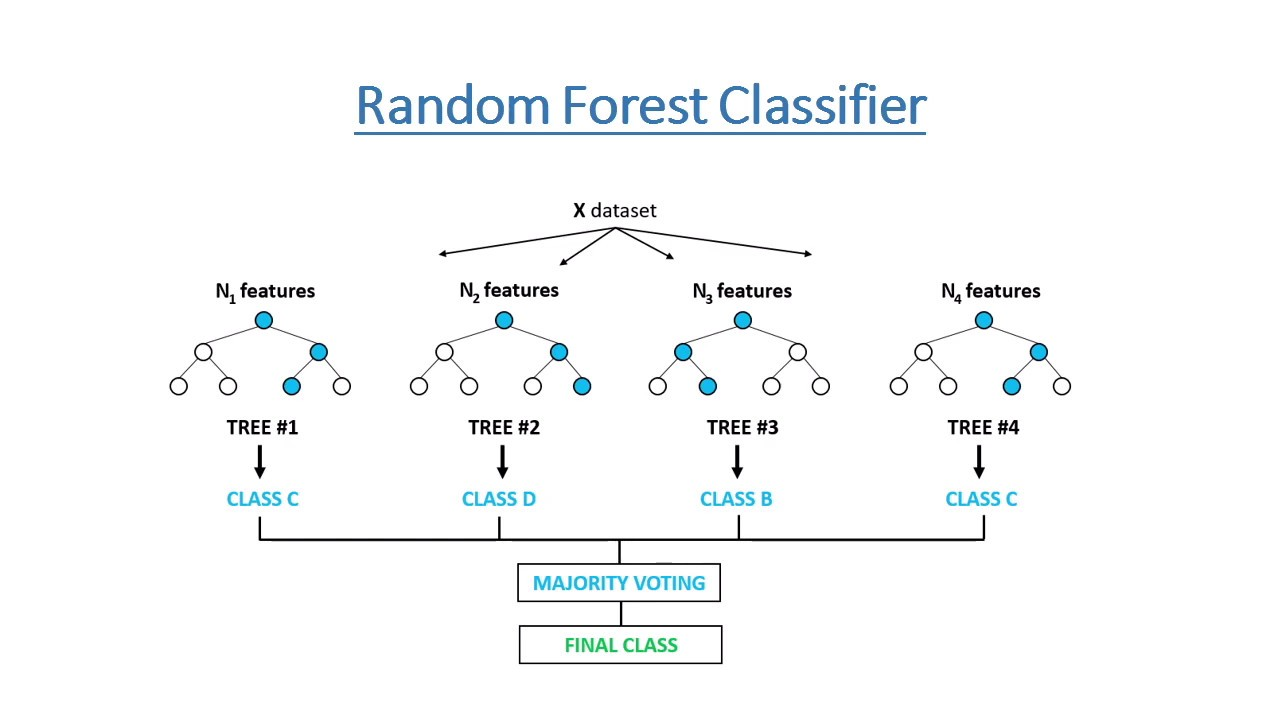


Figure 4.5 Random Forest Classifier

**4.1.5.2 Yolov7-**

YOLOv7 is a single-stage real-time object detector. It was introduced to the YOLO family in July’22. According to the YOLOv7 paper, it is the fastest and most accurate real-time object detector to date. YOLOv7 established a significant benchmark by taking its performance up a notch. Starting with the YOLOv7- model, the smallest in the family with just over 6 million parameters. With a validation AP of 35.2%, it beats YOLOv4- models with similar parameters.

YOLOv7- is a basic model optimized for edge GPU. The suffix “” of computer vision models means that they are optimized for Edge AI and deep learning workloads, and more lightweight to run ML on mobile computing devices or distributed edge servers and devices. This model is important for distributed real-world computer vision applications. Compared to the other versions, the edge-optimized YOLOv7- uses leaky ReLU as the activation function, while other models use SiLU as the activation function. Compared to the edge-optimized version YOLOv4-, YOLOv7- reduces the number of parameters by 39%, while also reducing computation by 49%, while achieving the same AP.

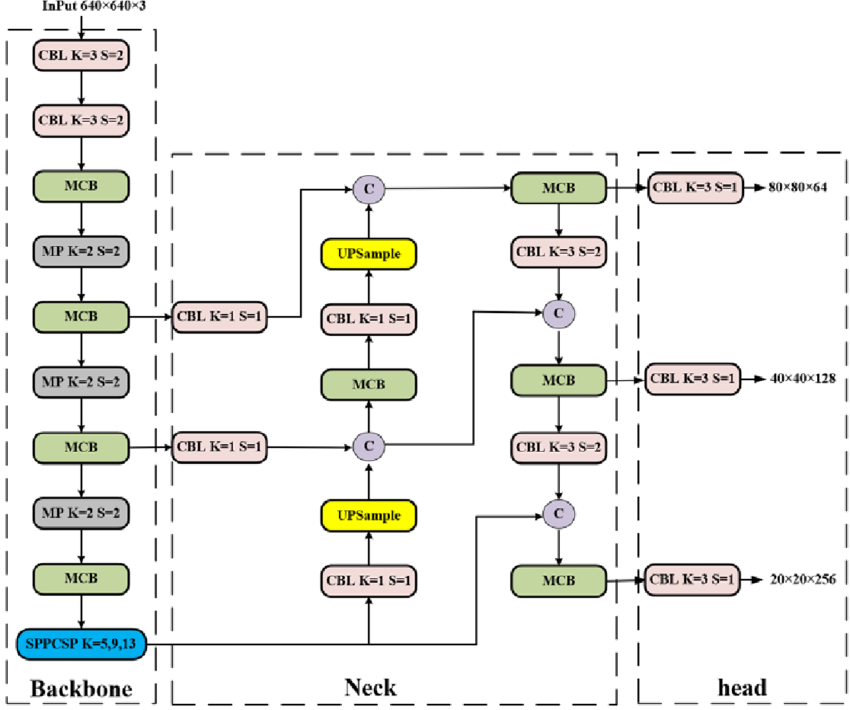


Figure 4.6 Yolov7-

**4.1.6 Validation and Evaluation**

After training with machine learning and deep learning algorithms, it will validate and evaluate the datasets. Validation in machine learning or deep learning is like a authorization or authentication of the prediction done by a trained model. While on the other hand, evaluation in machine learning or deep learning refers to assessment or test of entire machine or deep learning model and its performance in various circumstances. It involves assessment of machine learning or deep learning model training process and how accurate is the predictions given in different situations.

**4.1.7 Detection and Classification of Lung Disease**

After validation and evaluation, when an input image is given for prediction process, it can easily predict and classifies the lung disease such as Atelectasis, Emphysema, Lung Opacity, Lung cavity etc.

**4.1.8 Web Application Development**

ReactJS is JavaScript library used for building reusable UI components. According to React official documentation, following is the definition −

React is a library for building composable user interfaces. It encourages the creation of reusable UI components, which present data that changes over time. Lots of people use React as the V in MVC. React abstracts away the DOM from us, offering a simpler programming model and better performance. React can also render on the server using Node, and it can power native apps using React Native. React implements one-way reactive data flow, which reduces the boilerplate and is easier to reason about than traditional data binding.

**JSX**

JSX stands for JavaScript XML. JSX allows us to write HTML in React. JSX makes it easier to write and add HTML in React.

It is faster than normal JavaScript as it performs optimizations while translating to regular JavaScript. It makes easier for us to create templates. Instead of separating the markup and logic in separated files, React uses components for this purpose. We will learn about components in details in further articles.

**Components**

Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and returns HTML via a render function.

Components come in two types, Class components and Function components, in this tutorial we will concentrate on Class components.

**Unidirectional data flow and Flux**

React implements one-way data flow which makes it easy to reason about our app. Flux is a pattern that helps keeping our data unidirectional.

In this project the sensor data will be displayed, Air quality API will be integrated in the backend and machine learning prediction will also be integrated for prediction of type of crop.

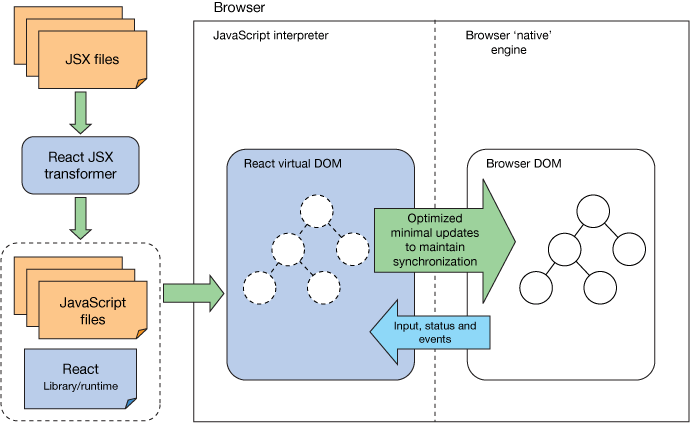


Figure 4.7 Web app development

**CHAPTER – 5**

**SOFTWARE DESCRIPTION**

The purpose of the Software Requirement Specification is to produce the specification of the analysis task and also to establish complete information about the requirement, behavior and also the other constraint like functional performance and so on. The main aim of the Software Requirement Specification is to completely specify the technical requirements for the software product in a concise and in unambiguous manner.

**5.1 VISUAL STUDIO**

In this project the Microsoft visual studio is used as an IDE.

Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

First and foremost, it is an editor that gets out of our way. The delightfully frictionless edit-build-debug cycle means less time fiddling with our environment, and more time executing on our ideas.

Visual Studio Code supports macOS, Linux, and Windows - so we can hit the ground running, no matter the platform.

At its heart, Visual Studio Code features a lightning-fast source code editor, perfect for day-to-day use. With support for hundreds of languages, VS Code helps us be instantly productive with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more. Intuitive keyboard shortcuts, easy customization and community-contributed keyboard shortcut mappings let us navigate our code with ease.

For serious coding, we'll often benefit from tools with more code understanding than just blocks of text. Visual Studio Code includes built-in support for IntelliSense code completion, rich semantic code understanding and navigation, and code refactoring.

And when the coding gets tough, the tough get debugging. Debugging is often the one feature that developers miss most in a leaner coding experience, so we made it happen. Visual Studio Code includes an interactive debugger, so we can step through source code, inspect variables, view call stacks, and execute commands in the console.

VS Code also integrates with build and scripting tools to perform common tasks making everyday workflows faster. VS Code has support for Git so we can work with source control without leaving the editor including viewing pending changes diffs.

Customize every feature to our liking and install any number of third-party extensions. While most scenarios work "out of the box" with no configuration, VS Code also grows with us, and we encourage us to optimize our experience to suit our unique needs.

VS Code includes enriched built-in support for Node.js development with JavaScript and TypeScript, powered by the same underlying technologies that drive Visual Studio. VS Code also includes great tooling for web technologies such as JSX/React, HTML, CSS, SCSS, Less, and JSON.

Architecturally, Visual Studio Code combines the best of web, native, and language-specific technologies. Using Electron, VS Code combines web technologies such as JavaScript and Node.js with the speed and flexibility of native apps. VS Code uses a newer, faster version of the same industrial-strength HTML-based editor that has powered the “Monaco” cloud editor, Internet Explorer's F12 Tools, and other projects. Additionally, VS Code uses a tools service architecture that enables it to integrate with many of the same technologies that power Visual Studio, including Roslyn for .NET, TypeScript, the Visual Studio debugging engine, and more.

Visual Studio Code includes a public extensibility model that lets developers build and use extensions, and richly customize their edit-build-debug experience.

**5.2 PYTHON**

In this project python is used as a programming language for development.

In technical terms, Python is an object-oriented, high-level programming language with integrated dynamic semantics primarily for web and app development. It is extremely attractive in the field of Rapid Application Development because it offers dynamic typing and dynamic binding options.

Python is relatively simple, so it's easy to learn since it requires a unique syntax that focuses on readability. Developers can read and translate Python code much easier than other languages. In turn, this reduces the cost of program maintenance and development because it allows teams to work collaboratively without significant language and experience barriers.

Additionally, Python supports the use of modules and packages, which means that programs can be designed in a modular style and code can be reused across a variety of projects. Once a module or package is developed by an user, it can be scaled for use in other projects, and it's easy to import or export these modules.

One of the most promising benefits of Python is that both the standard library and the interpreter are available free of charge, in both binary and source form. There is no exclusivity either, as Python and all the necessary tools are available on all major platforms. Therefore, it is an enticing option for developers who don't want to worry about paying high development costs.

**5.3 GOOGLE COLAB**

In this project, google colab is used as an open-source IDE.

Google Colaboratory is a free online cloud-based Jupyter notebook environment that allows us to train our machine learning and deep learning models on CPUs, GPUs, and TPUs.

It gives us a decent GPU for free, which we can continuously run for 12 hours. For most data science folks, this is sufficient to meet their computation needs.

Google Colab gives us three types of runtime for our notebooks:

* CPUs,
* GPUs, and
* TPUs

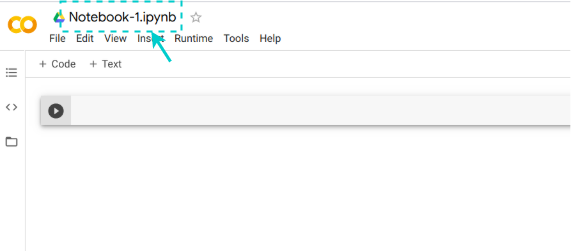
Colab gives us 12 hours of continuous execution time. After that, the whole virtual machine is cleared and we have to start again. We can run multiple CPU, GPU, and TPU instances simultaneously, but their resources are shared between these instances.

Figure 5.1 Google Colab Notebook

Colab notebooks allow us to combine executable code and rich text in a single document, along with images, HTML, LaTeX and more. When we create our own Colab notebooks, they are stored in our Google Drive account. We can easily share our Colab notebooks with co-workers or friends, allowing them to comment on our notebooks or even edit them. To learn more, see Overview of Colab. To create a new Colab notebook we can use the File menu above, or use the following link: create a new Colab notebook. Colab notebooks are Jupyter notebooks that are hosted by Colab.

As a developer, we can perform the following using Google Colab;

* Write and execute code in Python
* Create/Upload/Share notebooks
* Import/Save notebooks from/to Google Drive
* Import/Publish notebooks from GitHub
* Import external datasets
* Integrate PyTorch, TensorFlow, Keras, OpenCV
* Free Cloud service with free GPU

**CHAPTER – 6**

**RESULTS AND DISCUSSIONS**

**6.1 INTRODUCTION**

This chapter discusses about the practical results obtained while implementing the project.

**6.2 RESULTS OBTAINED**

To begin with, testing of the trained model, we can split our project into modules of implementation that is done.

Dataset collection involves the process of collecting dataset from Kaggle.

The dataset has been collected for the project and the below figure can be seen as follows:

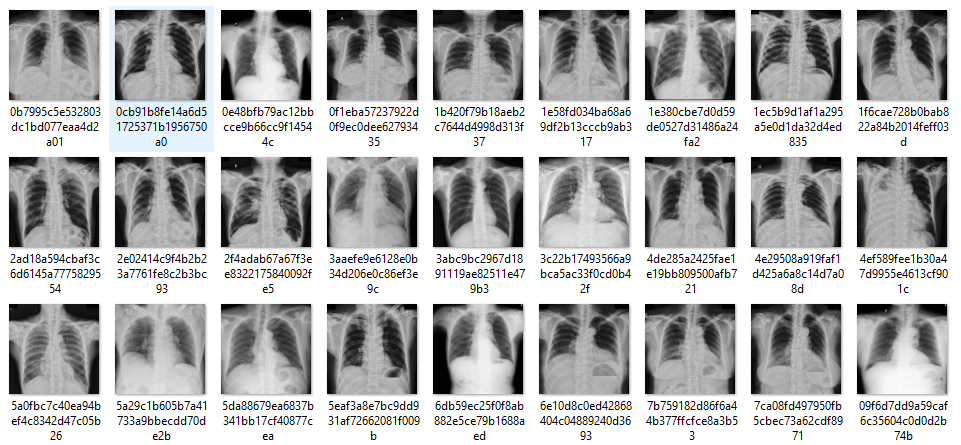


Figure 6.1 Dataset Collection

The below figure shows visualization of VINDR-CXR 28 chest xray abnormalities dataset.

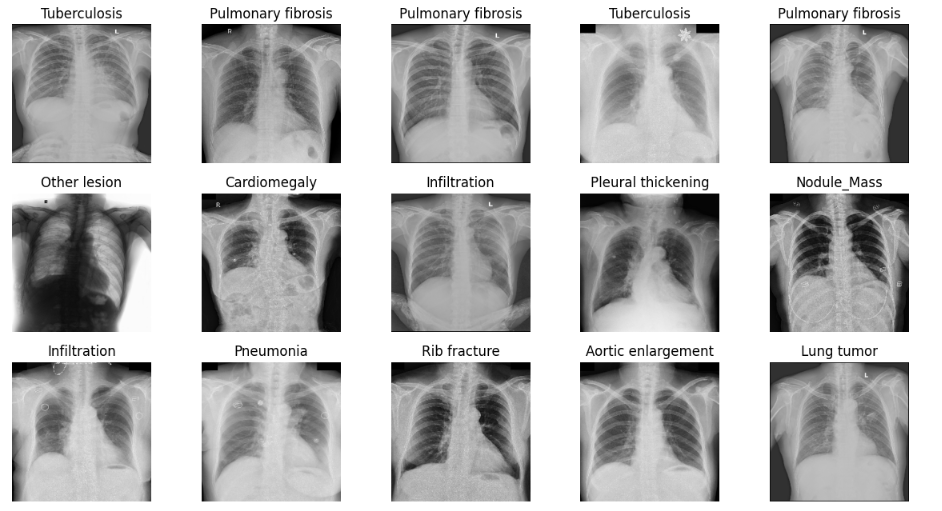


Figure 6.2 Data Visualization

Then these datasets are pre-processed from convert the images into required size format so that it can be made ready for training with the model.

The below figure shows the simple pre-processing techniques used for image resizing.

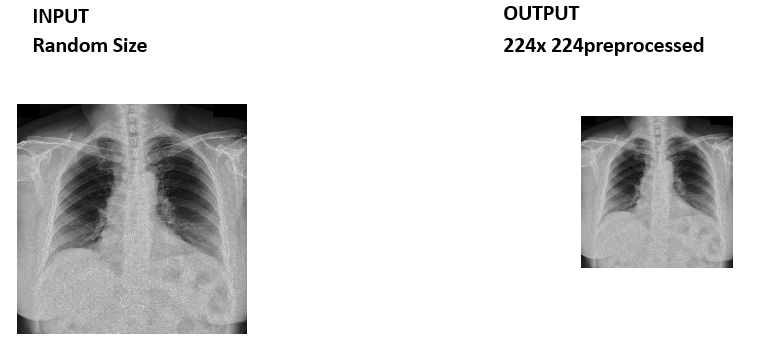


Figure 6.3 Simple Pre-processing

The below image is input of Data Augmentation.

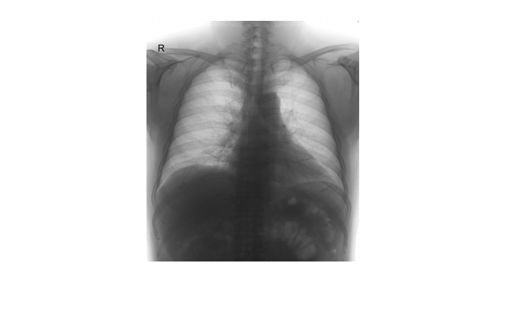


Figure 6.4 Input of Data Augumentation.

The below images represent output of Data Augmentation.

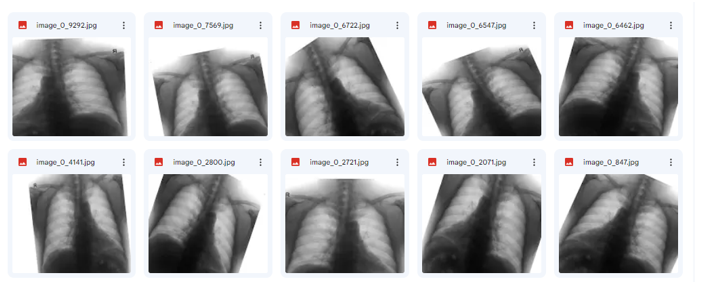


Figure 6.5 Data Augmentation Output.

The below figure shows that how the classes are distributed using Bar Plot

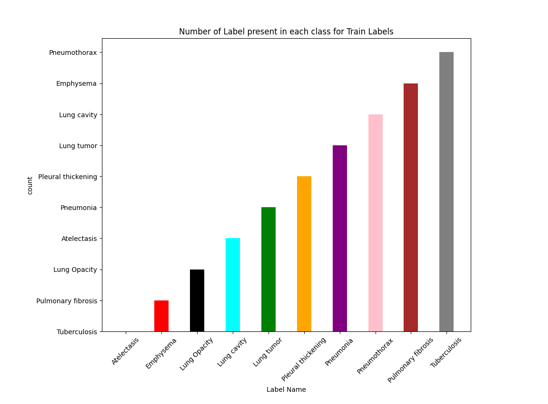


Figure 6.6 Class Distribution Using Box Plot

The below Figure shows that how much data present in each class in percentage.

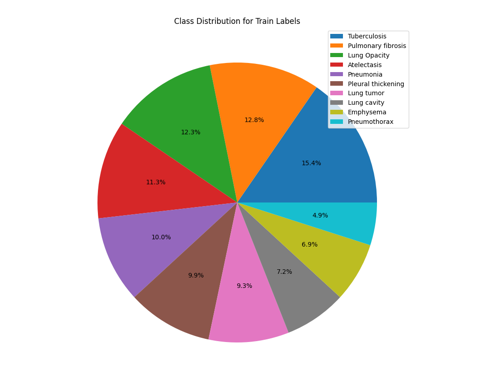


Figure 6.7 Class Distribution in Percentage Using Pie Chart

Feature extraction methods such as InceptionV3, Xception, InceptionResNetV2 and NASNetLarge under transfer learning are used to extract the features.

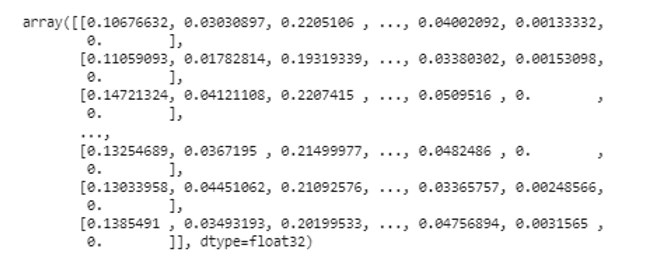


Figure 6.8 Feature Extraction Using InceptionV3, Xception, InceptionResNetV2 and NASNetLarge.

The below Figure shows that Number of data present in Training and Testing Set.

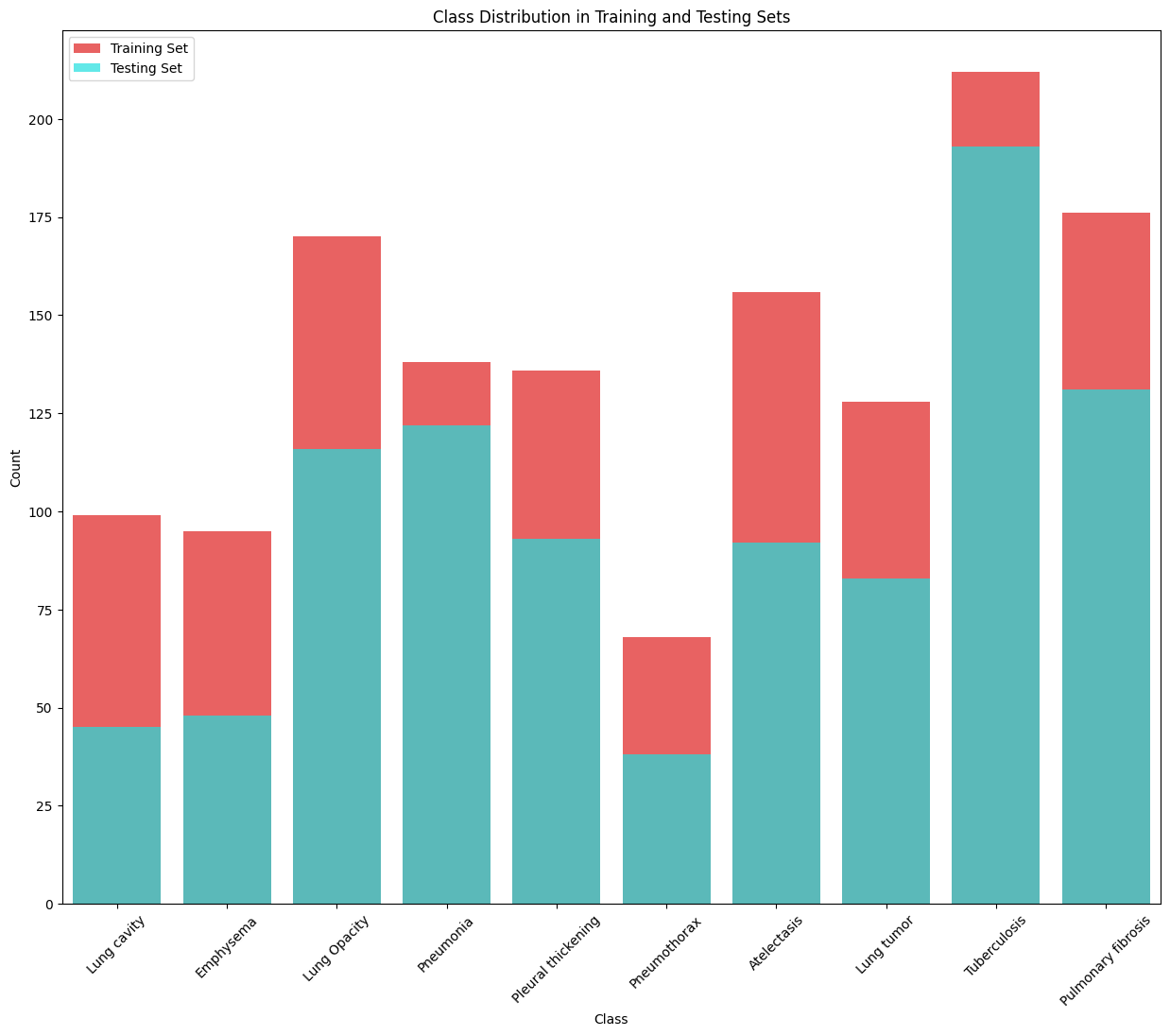


Figure 6.9 Number of Data Present in Training and Testing Set

The below Figure shows that Number of Data present in Training and Testing Dataset

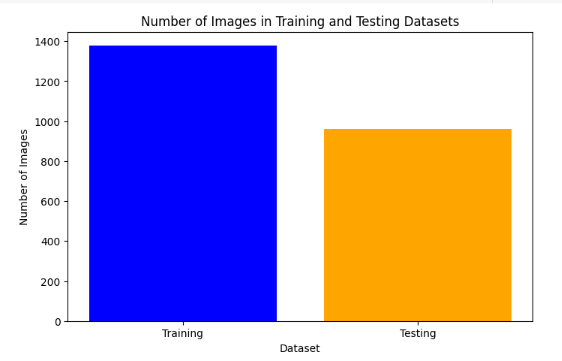


Figure 6.10 Number of Data present in Training and Testing Dataset.

The below classification report displays the precision, recall, F1, and support scores for the model.

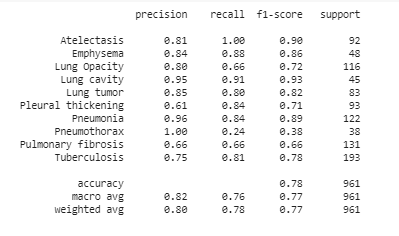
****

Figure 6.11 Classification Report

The below confusion matrix visualizes and summarizes the performance of a classification.

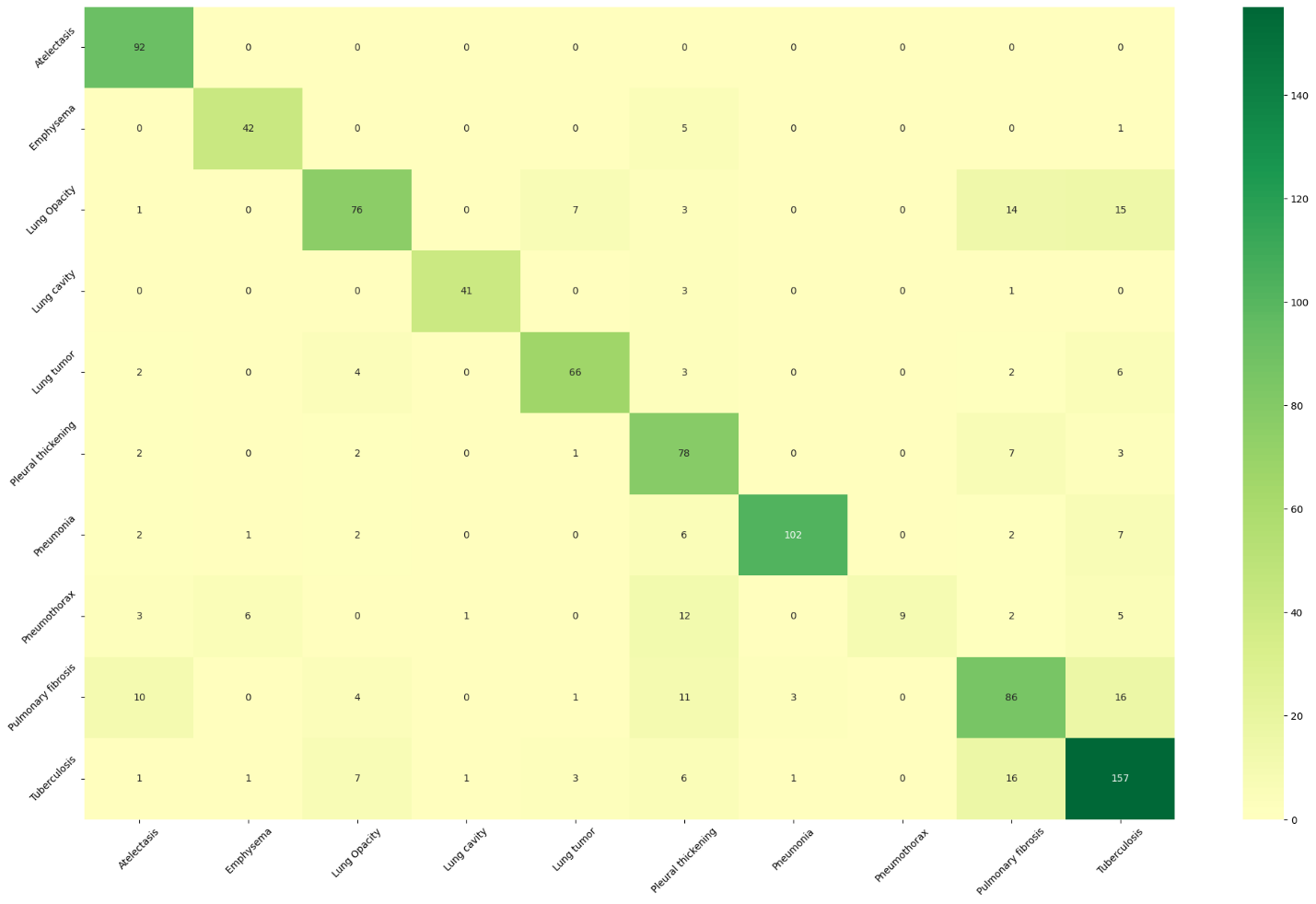
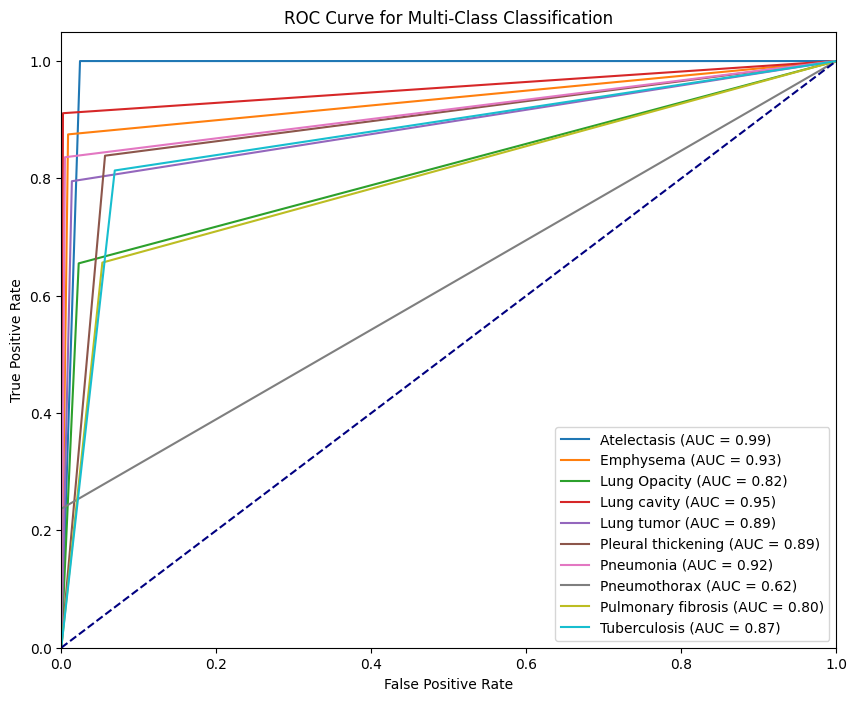


Figure 6.12 Confusion Matrix

The below graph showing the performance of a classification model at all classification thresholds.

Figure 6.13 ROC Curve

The below Figure shows Different Metric Comparison

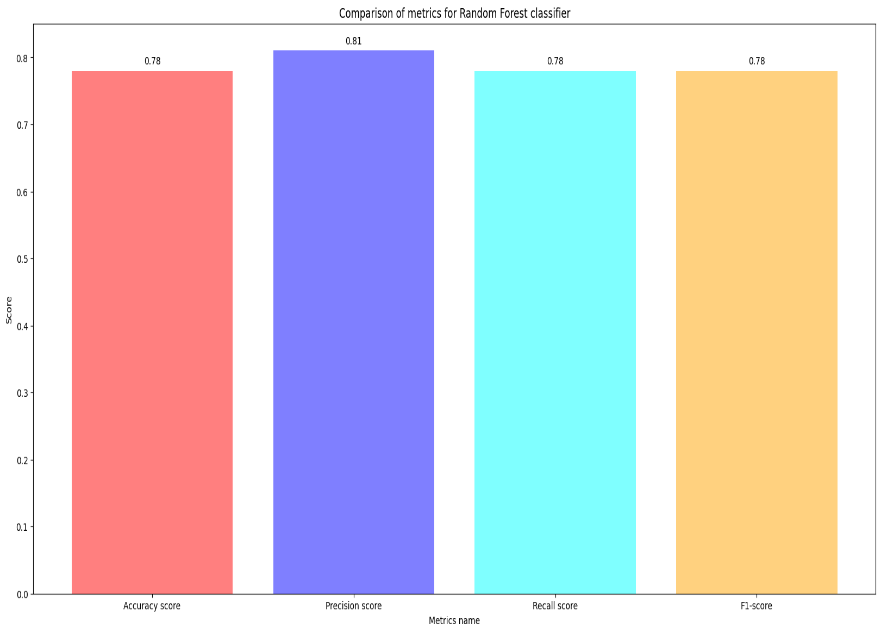


Figure 6.14 Different Matric Comparison

The below Figure shows training Result of YOLO V7 Algorithm.

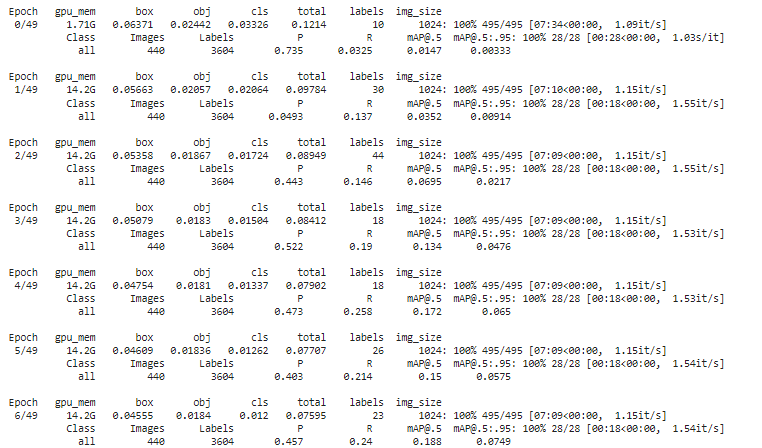


Figure 6.15 Training Result of YOLO V7 Algorithm.

The below Figure shows Training Metrics of YOLO V7 algorithm.

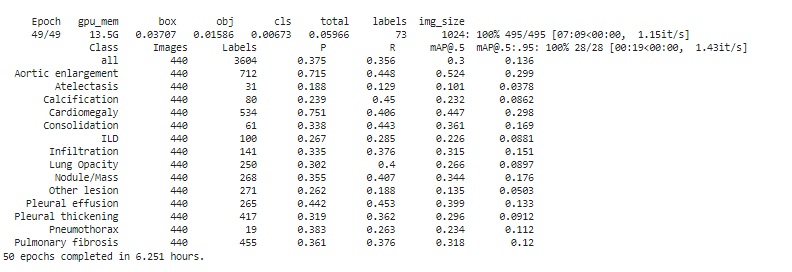


Figure 6.16 Training Metrics of YOLO V7 algorithm.

The below confusion matrix visualizes and summarizes the performance of a classification.

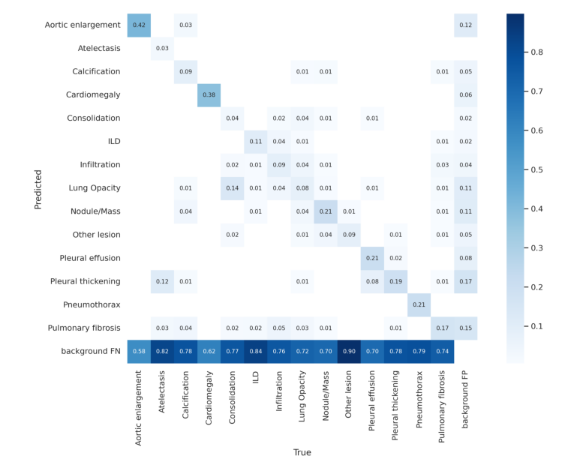


Figure 6.17 Confusion Matrix

The below Figure shows Backend Integrated with Frontend.



Figure 6.18 Backend Integrated Frontend Using Flask Framework and NGROK

The below Figure shows Login Page in Web Application.

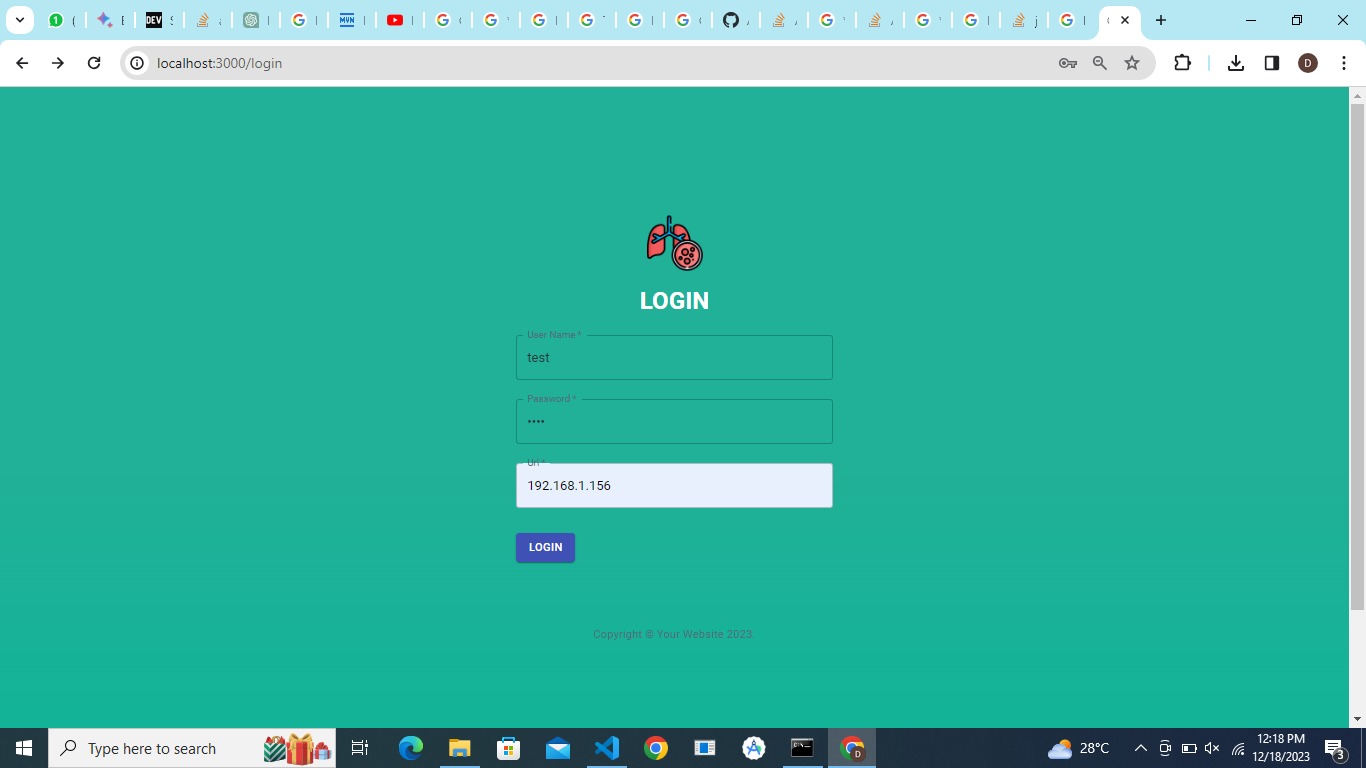


Figure 6.19 Login Page

The below Figure shows uploading page in Web Application.

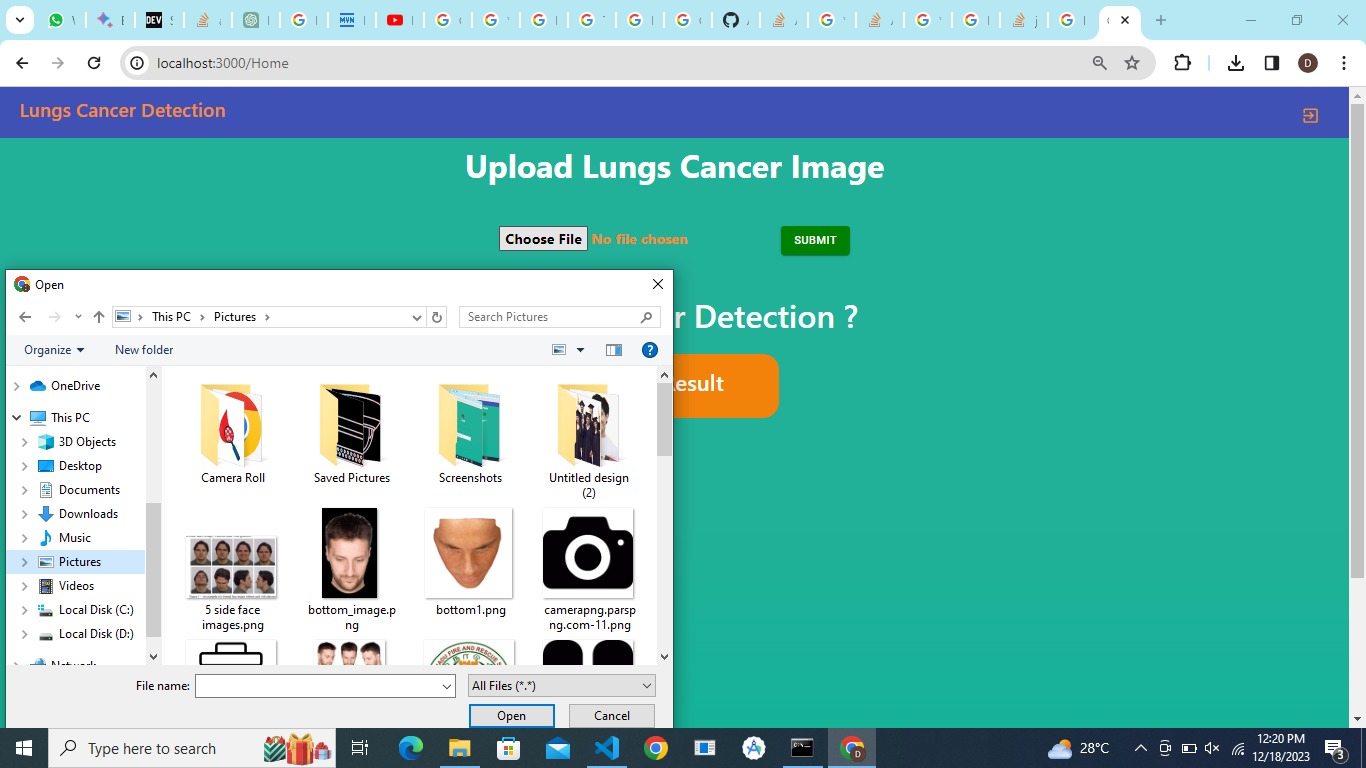


Figure 6.20 Upload Image

The below Figure shows prediction of Pulmonary fibrosis via heat map and yolov7 detecting infiltration and pulmonary fibrosis and ild

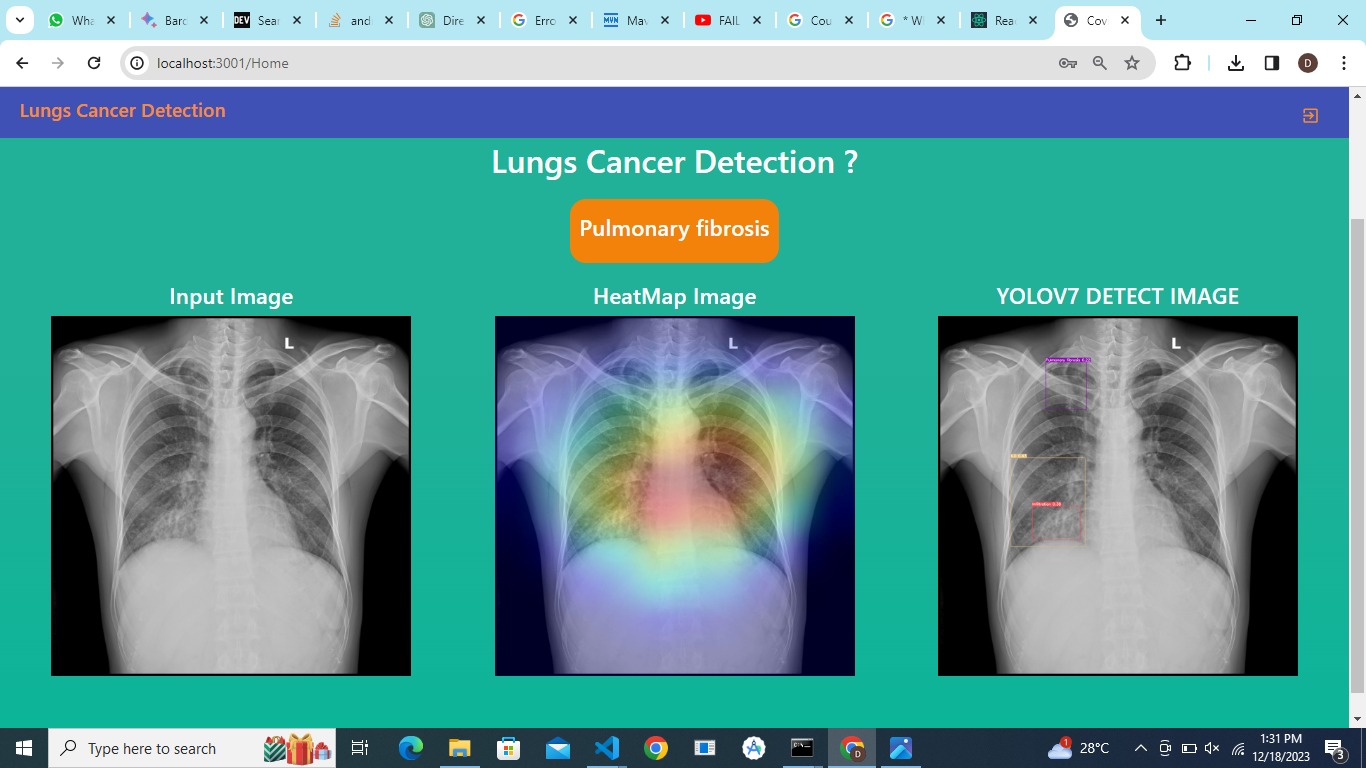


Figure 6.21 Prediction Process

The below Figure shows prediction Lung opacity. Heatmap and yolov7 detects nodules, cardiomegaly, lung opacity

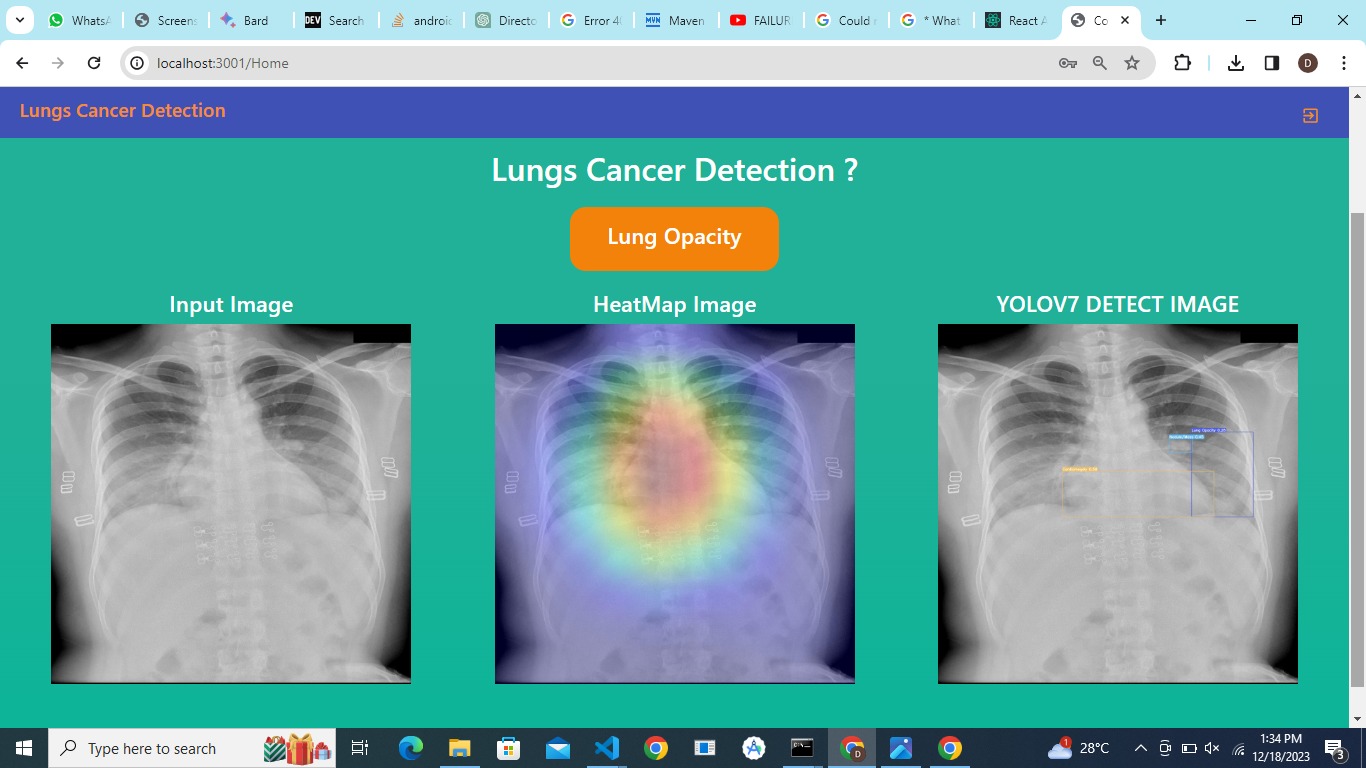


Figure 6.22 Prediction Process

**CHAPTER – 7**

**CONCLUSION & FUTURE WORK**

**7.1 CONCLUSION:**

The project has been successfully implemented to detect and classification of at an early stage using the deep learning and machine learning algorithms and provide prior measures to avoid the disease. The machine learning algorithm such as Random Forest Classifier and deep learning algorithm such as yolov7- model to effectively determine the lung disease. This also help in providing efficient treatment in a most cheap way and eventually reduce the time required for finding the in the current state. There is lot of scope to improve the technology as the diagnosis can be done in several means.

**7.2 FUTURE WORK:**