CANCER PREDICTION WEB APP USING MACHINE LEARNING AND DEEP LEARNING

A Project report submitted in partial fulfilment of the requirements for

the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

G Nishkala (318126512078) Sai KiranPatro (318126512098)

T Sannath Kumar (318126512107) Venkat Naidu G(318126512117)

Under the guidance of

P.Devi Pradeep

ECE DEPARTMENT

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(UGC AUTONOMOUS)

(*Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade*)

Sangivalasa, bheemilimandal, visakhapatnamdist.(A.P)

2021-2022

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(UGC AUTONOMOUS)

(***Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC)***

Sangivalasa, Bheemili Mandal, Visakhapatnam dist. (A.P)

CERTIFICATE

This is to certify that the project report entitled "**CANCER PREDICTION WEB APP USING MACHINE LEARNING AND DEEP LEARNING"** Submitted by **Sai Kiran Patro (318126512098), Venkat Naidu G (318126512117), T Sannath kumar (318126512107), G Nishkala (318126512078)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Electronics & Communication Engineering** of Andhra University, Visakhapatnam is a record of bonafide work carried out under my guidance and supervision.

Project Guide

P. Devi pradeep

Mr. P Devi Pradeep
Assistant Professor
Department of E.C.E
ANITS

Assistant Professor
Department of E.C.E.
Anil Neerukonda
Institute of Technology & Sciences
Sangivalasa, Visakhapatnam-531 162

Head of the Department

[illegible]

Dr.V.Rajyalakshmi
Professor&HOD
Department of E.C.E
ANITS

Head of the Department
Department of E C E
Anil Neerukonda Institute of Technology & Sciences
Sangivalasa - 531 162

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guide **Mr.P.Devi Pradeep**, Department of Electronics and Communication Engineering, ANITS, for his guidance with unsurpassed knowledge and immense encouragement. We are grateful to **Dr.V. Rajyalakshmi**, Head of the Department, Electronics and Communication Engineering, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the **Principal and Management, ANITS, Sangivalasa,** for their encouragement and cooperation to carry out this work.

We express our thanks to all **teaching faculty** of Department of ECE, whose suggestions during reviews helped us in accomplishment of our project. We would like to thank **all non-teaching staff** of the Department of ECE, ANITS for providing great assistance in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last, but not the least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

PROJECT STUDENTS

Nishkala G (318126512078)

Sai Kiran Patro (318126512098)

T Sannath Kumar (318126512107)

Venkat Naidu G (318126512117)

ABSTRACT

In today’s world image processing and machine learning has been the most trending technology which is used in almost all applications from automation to healthcare sector. Machine learning is one of the core fundamentals of Artificial Intelligence which can solve real life problems. Healthcare is one of the major concern where Artificial Intelligence, ML , DL can solve the problems. This project describes with a graphical user interface(GUI) .

The idea is to implement features for bio-medical applications. The App UI consists of different feature buttons and the user has to upload the image or data and the output will be displayed on a new window. This application consists of two features one is to predict breast cancer and another one is to predict lung cancer. This application can be viewed on both mobile and web. The technology used are Deep learning, Machine learning , web. The libraries which used are Sklearn, Matplotlib, numpy , pandas,keras etc .

CONTENTS

LIST OF FIGURES

LIST OF ABBREVATIONS

CT	Computed Tomography
MRI	Magnetic Resonance Imaging
RF	Radio Frequency
VGG 16	Visual Geometry Group 16
CNN	Convolutional Neural Network
DL	Deep Learning
CNS	Central Nervous System
EBV	Epstein – Barr virus
GPU	Graphics Processing Unit
GUI	Graphical User Interface
ANN	Artificial Neural Network
DT	Decision Tree
DBN	Data Bus Network
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network

CHAPTER 1
INTRODUCTION

Health care is one of the major concern and as per the records cancer is one the most common health problem which is identified in Indians. Nearly 1 million people are being affected by cancer. Technology can play a vital role in predicting cancer cells before it gets worsen. Cancer is the second biggest cause of mortality worldwide, accounting for 9.6 million fatalities in 2018, or one in every six deaths.

To reduce the death rate of patients, the primary contact with cancer is required. Cancer is a vast category of illnesses that can begin in practically any organ or tissue of the body and spread to other organs when abnormal cells proliferate uncontrolled, invade adjacent regions of the body, and/or move to other organs. Men’s cancers include lung, prostate, colorectal, stomach, and liver cancer, whereas women's cancers include breast, colorectal, lung, cervical, and thyroid cancer. Breast cancer has become one of the most prevalent causes of mortality among women. Tumor classification can be used to diagnose breast cancer. Tumors are divided into two categories: malignant and benign tumours.To distinguish between these malignancies, doctors require a reliable diagnostic process. However, even professionals find it difficult to recognise malignancies in most cases. As a result, diagnosis requires the automation of diagnostic systems. Breast cancer, being the most common cancer in women, has historically had a high incidence and fatality rate. Breast cancer is predicted to account for 25% of all new cancer diagnoses and 15% of all cancer deaths among women globally, according to the most recent cancer data. Cancer detection is the process of determining whether or not a person has cancer. In the early 1980s, a CAD (Computer-Aided Diagnosis) was created to improve the survival rate and efficiency of clinicians while evaluating medical pictures. Decision trees, linear regression, random forest, K-nearest neighbours, and other machine learning algorithms have a significant influence in health care. We've also gone through the many deep learning methodologies, strategies, and algorithms that may be used for cancer diagnosis, detection, and prediction.

CHAPTER 2
LITERATURE SURVEY

Over the past decades, a continuous evolution related to cancer research has been performed. Scientists applied different methods, such as screening in early stage, in order to find types of cancer before they cause symptoms. Moreover, they have developed new strategies for the early prediction of cancer treatment outcome. With the advent of new technologies in the field of medicine, large amounts of cancer data have been collected and are available to the medical research community.

ML, a branch of Artificial Intelligence, relates the problem of learning from data samples to the general concept of inference. The last two decades a variety of different ML techniques and feature selection algorithms have been widely applied to disease prognosis and prediction. Major types of ML techniques including CNN(Convolution neural Networks) and DT(Decision Tree) have been used for nearly three decades in cancer detection.

According to the recent PubMed results regarding the subject of ML and cancer more than 7510 articles have been published until today. The vast majority of these publications makes use of one or more ML algorithms and integrates data from heterogeneous sources for the detection of tumors as well as for the prediction/prognosis of a cancer type.

Chapter 3:
MACHINE LEARNING

Introduction:

Machine learning is a subfield of computer science that evolved from the study of pattern recognition and computational learning theory in artificial intelligence. Machine learning is a form of AI that enables a system to learn from data rather than through explicit programming. However, machine learning is not a simple process. Machine learning explores the construction and study of algorithms that can learn from and make predictions on data. Such algorithms operate by building a model from example inputs in order to make data-driven predictions or decisions, rather than following strictly static program instructions.

It uses a variety of algorithms that iteratively learn from data to improve, describe data, and predict outcomes. As the algorithms ingest training data, it is then possible to produce more precise models based on that data. A machine learning model is the output generated when you train your machine learning algorithm with data. After training, when you provide a model with an input, you will be given an output. For example, a predictive algorithm will create a predictive model. Then, when you provide the predictive model with data, you will receive a prediction based on the data that trained the model. Machine learning is now essential for creating analytics models.

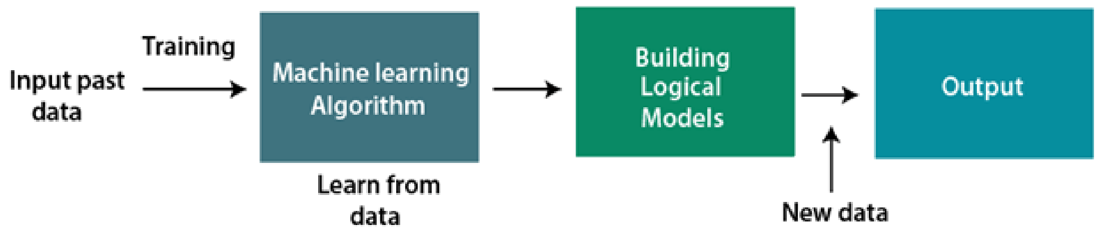


Fig 3.1: Block diagram that shows how machine

3.1.1 Data:

It is useful to characterize learning problems according to the type of data they use. This is a great help when encountering new challenges, since quite often problems on similar data types can be solved with very similar techniques.

3.2.2 Training Set and Test Set:

In machine learning, an unknown universal dataset is assumed to exist, which contains all the possible data pairs as well as their probability distribution of appearance in the real world. While in real applications, what we observed is only a subset of the universal dataset due to the lack of memory or some other unavoidable reasons. This acquired dataset is called the training set (training data) and used to learn the properties and knowledge of the universal dataset. In general, vectors in the training set are assumed independently and identically sampled (i.i.d) from the universal dataset.

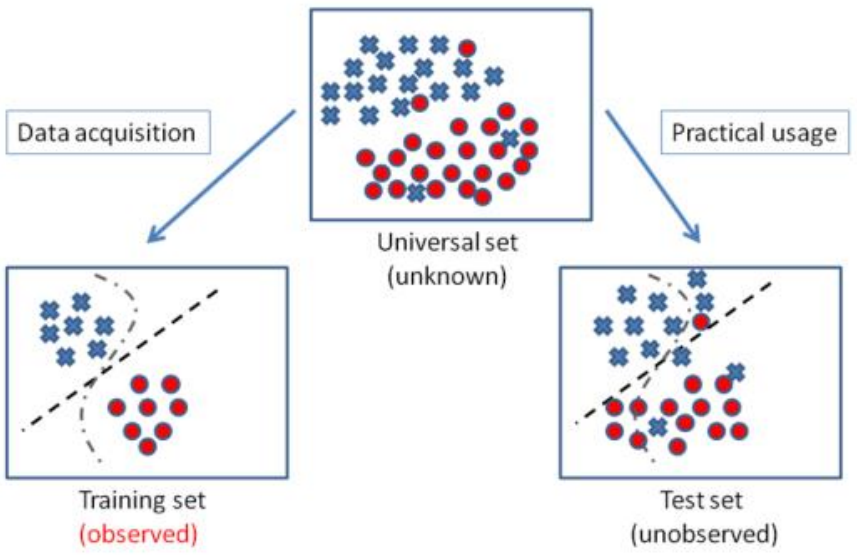


Fig 3.2 : figure showing how data is used to train and test the model

3.2 Types of Machine Learning:

Machine learning techniques are required to improve the accuracy of predictive models. There are different approaches based on the type and volume of the data. In this section, we discuss the categories of machine learning.

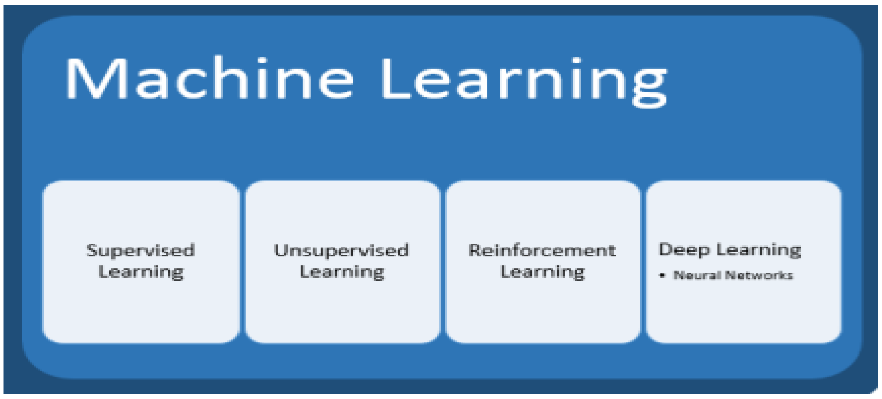


Fig 3.3 : figure showing different types of machine

3.2.1 Supervised Learning:

Supervised learning typically begins with an established set of data and a certain understanding of how that data is classified. Supervised learning is intended to find patterns in data that can be applied to an analytics process. This data has labeled features that define the meaning of data. For example, there could be millions of images of animals and include an explanation of what each animal is and then you can create a machine learning application that distinguishes one animal from another. By labeling this data about types of animals, you may have hundreds of categories of different species. Because the attributes and the meaning of the data have been identified, it is well understood by the users that are training the modeled data so that it fits the details of the labels. When the label is continuous, it is a regression; when the data comes from a finite set of values, it known as classification. In essence, regression used for supervised learning helps you understand the correlation between variables. An example of supervised learning is weather forecasting.

By using regression analysis, weather forecasting takes into account known historical weather patterns and the current conditions to provide a prediction on the weather. The algorithms are trained using preprocessed examples, and at this point, the performance of the algorithms is evaluated with test data. Occasionally, patterns that are identified in a subset of the data can’t be detected in the larger population of data. If the model is fit to only represent the patterns that exist in the training subset, you create a problem called over fitting. Over fitting means that your model is precisely tuned for your training data but may not be applicable for large sets of unknown data. To protect against over fitting, testing needs to be done against unforeseen or unknown labeled data. Using unforeseen data for the test set can help you evaluate the accuracy of the model in predicting outcomes and results. Supervised training models have broad applicability to a variety of business problems, including fraud detection, recommendation solutions, speech recognition, or risk analysis.

3.2.2 Unsupervised Learning:

Unsupervised learning is best suited when the problem requires a massive amount of data that is unlabeled. For example, social media applications, such as Twitter, Instagram , Snapchat , and so on all have large amounts of unlabeled data. Understanding the meaning behind this data requires algorithms that can begin to understand the meaning based on being able to classify the data based on the patterns or clusters it finds. Therefore, the supervised learning conducts an iterative process of analyzing data without human intervention. Unsupervised learning is used with email spam-detecting technology. There are far too many variables in legitimate and spam emails for an analyst to flag unsolicited bulk email. Instead, machine learning classifiers based on clustering and association are applied in order to identify unwanted email.

Unsupervised learning algorithms segment data into groups of examples (clusters) or groups of features. The unlabeled data creates the parameter values and classification of the data. In essence, this process adds labels to the data so that it becomes supervised. Unsupervised learning can determine the outcome when there is a massive amount of data. In this case, the developer doesn’t know the context of the data being analyzed, so labeling isn’t possible at this stage. Therefore, unsupervised learning can be used as the first step before passing the data to a supervised learning process.

Unsupervised learning algorithms can help businesses understand large volumes of new, unlabeled data. Similarly to supervised learning , these algorithms look for patterns in the data; however, the difference is that the data is not already understood. For example, in healthcare, collecting huge amounts of data about a specific disease can help practitioners gain insights into the patterns of symptoms and relate those to outcomes from patients. It would take too much time to label all the data sources associated with a disease such as diabetes. Therefore, an unsupervised learning approach can help determine outcomes more quickly than a supervised learning approach.

3.2.3 Reinforcement learning:

Reinforcement learning is a behavioral learning model. The algorithm receives feedback from the analysis of the data so the user is guided to the best outcome. Reinforcement learning differs from other types of supervised learning because the system isn’t trained with the sample data set. Rather, the system learns through trial and error. Therefore, a sequence of successful decisions will result in the process being “reinforced” because it best solves the problem at hand.

One of the most common applications of reinforcement learning is in robotics or game playing. Take the example of the need to train a robot to navigate a set of stairs. The robot changes its approach to navigating the terrain based on the outcome of its actions. When the robot falls, the data is recalibrated so the steps are navigated differently until the robot is trained by trial and error to understand how to climb stairs. In other words, the robot learns based on a successful sequence of actions. The learning algorithm has to be able to discover an association between the goal of climbing stairs successfully without falling and the sequence of events that lead to the outcome.

Reinforcement learning is also the algorithm that is being used for self-driving cars. In many ways, training a self-driving car is incredibly complex because there are so many potential obstacles. If all the cars on the road were autonomous, trial and error would be easier to overcome. However, in the real world, human drivers can often be unpredictable. Even with this complex scenario, the algorithm can be optimized over time to find ways to adapt to the state where actions are rewarded. One of the easiest ways to think about reinforcement learning is the way an animal is trained to take actions based on rewards.

3.2.4 Neural networks and deep learning:

Deep learning is a specific method of machine learning that incorporates neural networks in successive layers in order to learn from data in an iterative manner. Deep learning is especially useful when you're trying to learn patterns from unstructured data. This topic is briefly discussed in Chapter 5.

3.3 Algorithms:

Machine learning algorithms are different from other algorithms. With most algorithms, a programmer starts by inputting the algorithm. However, with machine learning the process is flipped. With machine learning, the data itself creates the model. The more data that is added to the algorithm, the more sophisticated the algorithm becomes. As the machine learning algorithm, is exposed to more and more data, it is able to create increasingly accurate algorithm.

3.3.1 Logistic Regression:

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**.

Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**. In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1). The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc. Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets. Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:

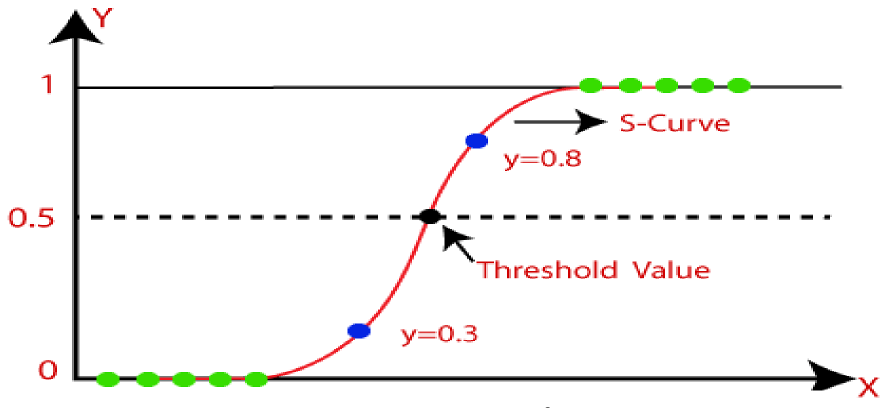


Fig 3.4: Logistic function

3.3.2 DECISION TREE:

Decision Tree is a **Supervised learning technique** that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where **internal nodes represent the features of a dataset, branches represent the decision rules** and **each leaf node represents the outcome.** In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset.

It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. In order to build a tree, we use the **CART algorithm,** which stands for **Classification and Regression Tree algorithm.** A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into sub trees.

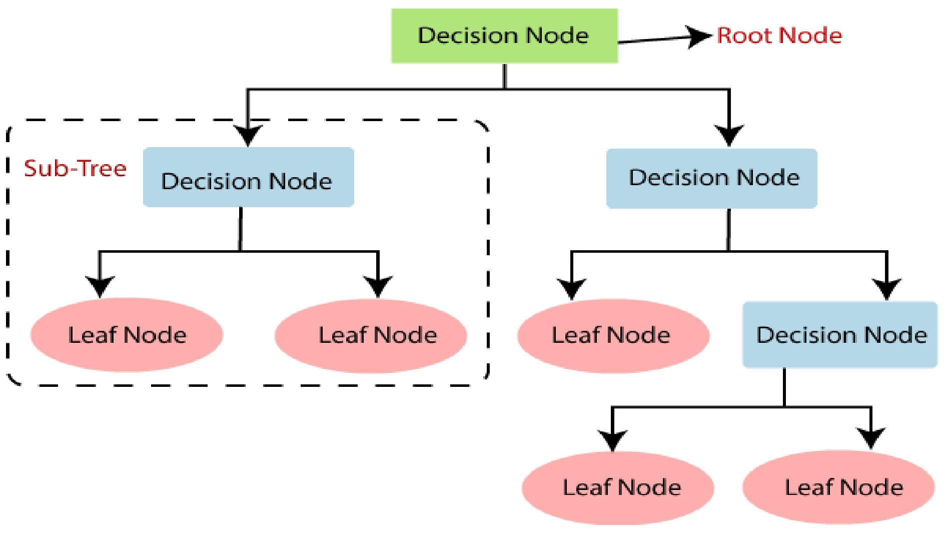


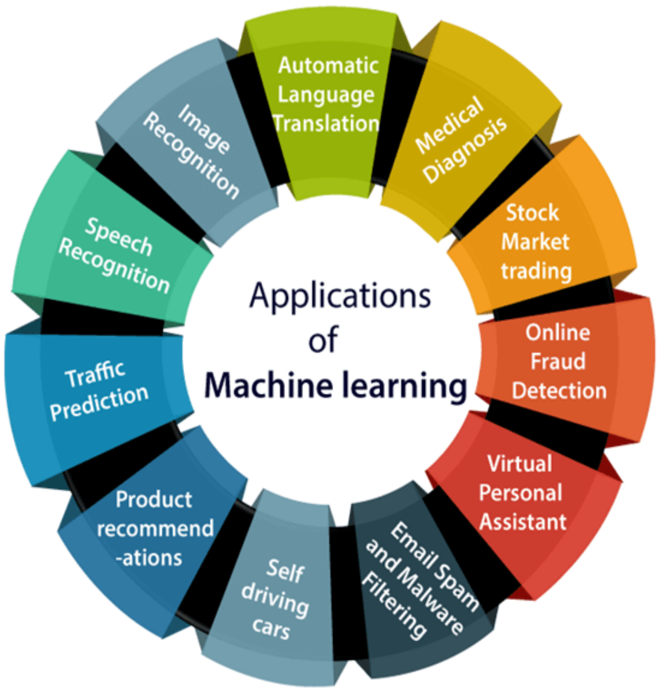
Fig 3.5: General structure of Decision Tree

3.3.3 Random forest:

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.The greater number of trees in the forest leads to higher accuracy and prevents the problem of over fitting. **This topic is briefly discussed in Chapter 4.**

3.4 Applications :

1. Image Recognition
2. Speech Recognition
3. Traffic prediction
4. Product recommendations
5. Self-driving cars
6. Email Spam and Malware Filtering
7. Virtual Personal Assistant
8. Online Fraud Detection
9. Stock Market trading
10. Medical Diagnosis
11. Automatic Language Translation


Applications of Machine learning
Automatic Language Translation
Medical Diagnosis
Stock Market trading
Online Fraud Detection
Virtual Personal Assistant
Email Spam and Malware Filtering
Self driving cars
Product recommend -ations
Traffic Prediction
Speech Recognition
Image Recognition

CHAPTER-4

RANDOM FOREST ALGORITHM

4.1 Introduction

A random forest is a supervised machine learning technique built using decision tree algorithms. To forecast behavior and results, this algorithm is used in a variety of sectors, including banking and e-commerce. "Random Forest is a classifier that comprises a number of decision trees on different subsets of the provided dataset and takes the average to enhance the predicted accuracy of that dataset," as the name implies. Rather of depending on a single decision tree, the random forest collects the forecasts from each tree and predicts the final output based on the majority votes of the predictions.

What is Decision Tree?

The Decision Node and the Leaf Node are the two nodes of a Decision tree. Decision nodes are used to make any decision and have several branches, whereas Leaf nodes are the results of such decisions and have no further branches. The judgement or test is based on the characteristics of the presented dataset.

It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.Below diagram explains the general structure of a decision tree:

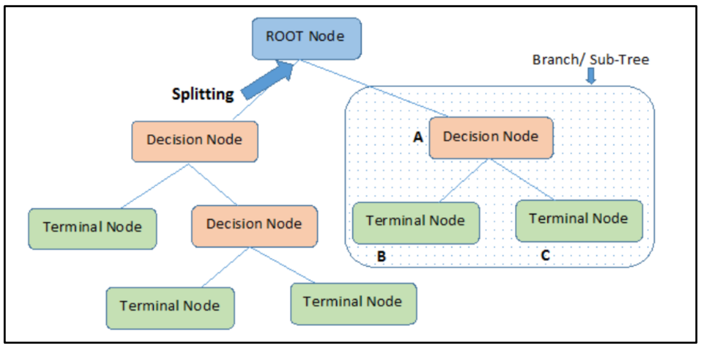


Fig: 4.1-Structure of decision tree

Decision Tree Terminologies

- **Root node:** The base of the decision tree.
- **Splitting:** The process of dividing a node into multiple sub-nodes.
- **Decision node:** When a sub-node is further split into additional sub-nodes.
- **Leaf node:** When a sub-node does not further split into additional sub-nodes; represents possible outcomes.
- **Branch:** A subsection of the decision tree consisting of multiple node

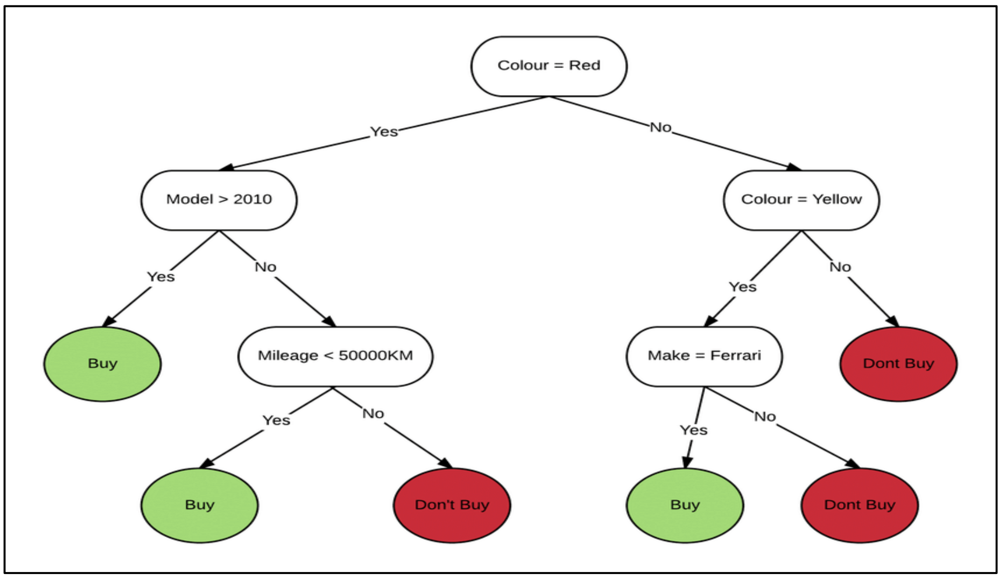


Fig: 4.2- Decision tree example.

Advantages of Decision Tree

- The algorithm is simple to understand, interpret and visualize as the idea is mostly used in our daily lives. Output of a Decision Tree can be easily interpreted by humans.
- Decision Tree looks like simple if-else statements which are very easy to understand.
- Decision Tree can handle both continuous and categorical variables.
- The data type of decision tree can handle any type of data whether it is numerical, categorical, or Boolean.

Disadvantages of Decision Tree

- Decision tree often involves higher time to train the model.
- A small change in the data can cause a large change in the structure of the decision tree causing instability.
- For a lot of category labels, the process quality of the choice tree could increase.

Why use Random Forest?

Below are some points that explain why we should use the Random Forest algorithm:

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

Important Terms to Know

There are different ways that Random Forest algorithm makes data decisions, and consequently, there are some important related terms to know. Some of these terms include:

- **Entropy:** It is a measure of randomness or unpredictability in the data set.
- **Information Gain:** A measure of the decrease in the entropy after the data set is split is the information gain.
- **Leaf Node:** A leaf node is a node that carries the classification or the decision.
- **Decision Node:** A node that has two or more branches.
- **Root Node:** The root node is the topmost decision node, which is where you have all of your data.

4.2 Working of Random Forest Algorithm

We can understand the working of Random Forest algorithm with the help of following steps −

Step 1 − First, start with the selection of random samples from a given dataset.

Step 2 − Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.

Step 3 − In this step, voting will be performed for every predicted result.

Step 4 − At last, select the most voted prediction result as the final prediction result.

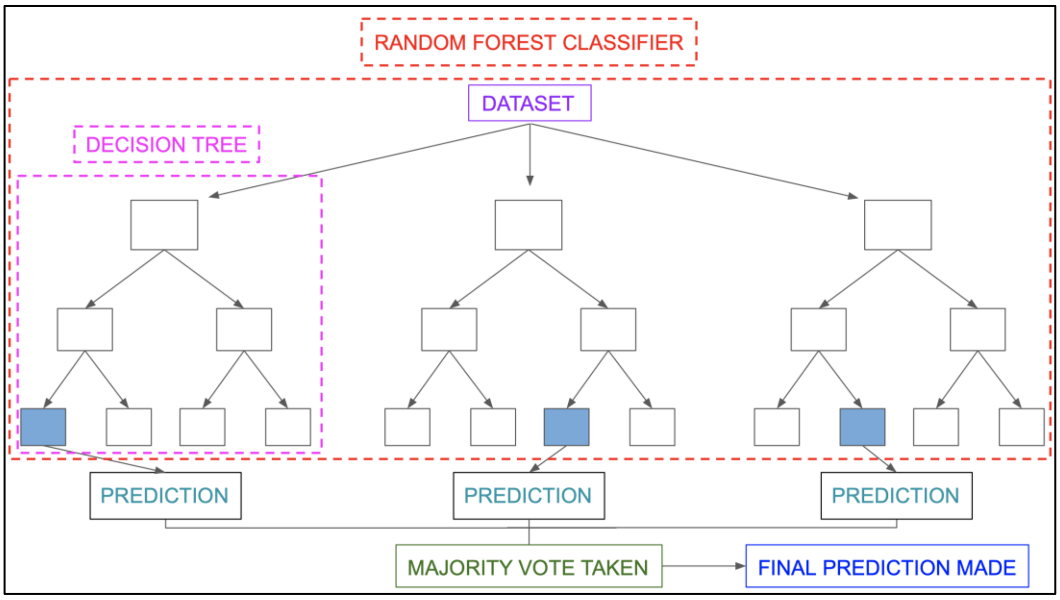


Fig: 4.3- Random Forest Classifier

Example for Random Forest

Let's imagine we want to categories the many sorts of fruits in a bowl based on numerous characteristics, however the bowl is crowded with many alternatives. You would generate a training dataset with information about the fruit, such as colours, diameters, and particular labels (i.e., apple, grapes, etc.) The data would next need to be split by sorting it into the smallest pieces possible so that it could be distributed as widely as feasible. You might wish to start by sorting your fruits by diameter, then by colour. You'd want to keep dividing until that specific node no longer requires it and you can forecast a certain fruit with 100 percent accuracy.

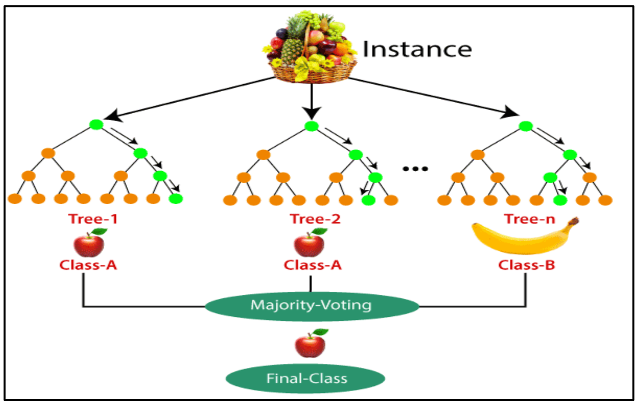


Fig: 4.4- Random forest example

Advantages of random forest

- It can perform both regression and classification tasks.
- A random forest produces good predictions that can be understood easily.
- It can handle large datasets efficiently.
- The random forest algorithm provides a higher level of accuracy in predicting outcomes over the decision tree algorithm.

Disadvantages of random forest

- It requires much computational power as well as resources as it builds numerous trees to combine their outputs.
- It also requires much time for training as it combines a lot of decision trees to determine the class.
- Due to the ensemble of decision trees, it also suffers interpretability and fails to determine the significance of each variable.

4.4 Features of Random Forests

- It is unexcelled in accuracy among current algorithms.
- It runs efficiently on large data bases.
- It can handle thousands of input variables without variable deletion.
- It gives estimates of what variables are important in the classification.
- It generates an internal unbiased estimate of the generalization error as the forest building progresses.

- It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.
- It has methods for balancing error in class population unbalanced data sets.
- Generated forests can be saved for future use on other data.

4.5 Some major Applications of Random Forest in different sectors:

- **Banking Industry**
  - Credit Card Fraud Detection
  - Customer Segmentation
  - Predicting Loan Defaults
- **Healthcare and Medicine**
  - Cardiovascular Disease Prediction
  - Diabetes Prediction
  - Breast Cancer Prediction
- **Stock Market**
  - Stock Market Prediction
  - Stock Market Sentiment Analysis
  - Bitcoin Price Detection
- **E-Commerce**
  - Product Recommendation
  - Price Optimization
  - Search Ranking

CHAPTER 5
DEEP LEARNING

5.1 Introduction:

Deep learning is a branch of machine learning which is completely based on artificial neural networks, as neural network is going to mimic the human brain so deep learning is also a kind of mimic of human brain. In deep learning, we don't need to explicitly program everything. The concept of deep learning is not new. It has been around for a couple of years now. It's on hype nowadays because earlier we did not have that much processing power and a lot of data.

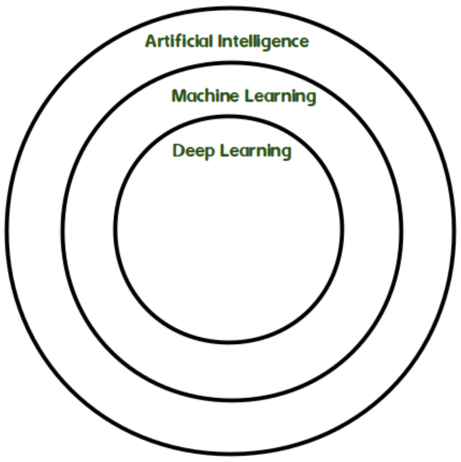


Fig 5.1: Indicates how AI, ML & DL are connected

5.2 Architectures of Deep Learning:

Deep learning architecture consists of deep/neural networks of varying topologies. The general principle is that neural networks are based on several layers that proceed data–an input layer (raw data), hidden layers (they process and combine input data), and an output layer (it produces the outcome: result, estimation, forecast, etc.).

5.2.1 Recurrent Neural Networks:

RNN is one of the fundamental network architectures from which other deep learning architectures are built. RNNs consist of a rich set of deep learning architectures. They can use their internal state (memory) to process variable-length sequences of inputs. Let’s say that RNNs have a memory. Every processed information is captured, stored, and utilized to calculate the final outcome. This makes them useful when it comes to, for instance, speech recognition. Moreover, the recurrent network might have connections that feedback into prior layers (or even into the same layer). This feedback allows them to maintain the memory of past inputs and solve problems in time.

RNNs are very useful when it comes to fields where the sequence of presented information is key. They are commonly used in NLP (i.a.chatbots), speech synthesis, and machine translations.

Currently, There are two types of RNN:

- Bidirectional RNN: They work two ways the output layer can get information from past and future states simultaneously.
- Deep RNN: Multiple layers are present. As a result, the DL model can extract more hierarchical information.

5.2.2 LSTM: Long Short-Term Memory

It’s also a type of RNN. However, LSTM has feedback connections. This means that it can process not only single data points (such as images) but also entire sequences of data (such as audio or video files).

LSTM derives from neural network architectures and is based on the concept of a memory cell. The memory cell can retain its value for a short or long time as a function of its inputs, which allows the cell to remember what’s essential and not just its last computed value.

A typical LSTM architecture is composed of a cell, an input gate, an output gate, and a forget gate. The cell remembers values over arbitrary time intervals, and these three gates regulate the flow of information into and out of the cell.
The input gate controls when new information can flow into the memory.The output gate controls when the information that is contained in the cell is used in the output. The forget gate controls when a piece of information can be forgotten, allowing the cell to process new data.

5.2.3 Convolutional Neural Networks

This architecture is commonly used for image processing, image recognition, video analysis, and NLP.
CNN can take in an input image, assign importance to various aspects/objects in the image, and be able to differentiate one from the others. The name 'convolutional' derives from a mathematical operation involving the convolution of different functions. CNNs consist of an input and an output layer, as well as multiple hidden layers. The CNN's hidden layers typically consist of a series of convolutional layers.This topic will be briefly discussed in chapter 6

5.2.4 Deep Belief Network

DBN is a multilayer network (typically deep, including many hidden layers) in which each pair of connected layers is a Restricted Boltzmann Machine (RBM). Therefore, we can state that DBN is a stack of RBMs. DBN is composed of multiple layers of latent variables ("hidden units"), with connections between the layers but not between units within each layer. DBNs use probabilities and unsupervised learning to produce outputs. Unlike other models, each layer in DBN learns the entire input. In CNNs, the first layers only filter inputs for basic features, and the latter layers recombine all the simple patterns found by the previous layers. DBNs work holistically and regulate each layer in order.

5.3 Working of Deep Learning:

Neural networks are layers of nodes, much like the human brain is made up of neurons. Nodes within individual layers are connected to adjacent layers. The network is said to be deeper based on the number of layers it has. A single neuron in the human brain receives thousands of signals from other neurons. In an artificial neural network, signals travel between nodes and assign corresponding weights. A heavier weighted node will exert more effect on the next layer of nodes. The final layer compiles the weighted inputs to produce an output. Deep learning systems require powerful hardware because they have a large amount of data being processed and involves several complex mathematical calculations. Even with such advanced hardware, however, deep learning training computations can take weeks.

Deep learning systems require large amounts of data to return accurate results; accordingly, information is fed as huge data sets. When processing the data, artificial neural networks are able to classify data with the answers received from a series of binary true or false questions involving highly complex mathematical calculations.

For example, a facial recognition program works by learning to detect and recognize edges and lines of faces, then more significant parts of the faces, and, finally, the overall representations of faces. Over time, the program trains itself, and the probability of correct answers increases. In this case, the facial recognition program will accurately identify faces with time.

5.4 Advantages of Deep Learning:

1. It robust enough to understand and use novel data, but most data scientists have learned to control the learning to focus on what’s important to them. Deep learning takes advantage of this by allowing you to control the learning, but not the statistical modeling.

2. It allows us to teach a specific task rather than teaching the system how to learn. We can use different examples to train a particular model or we can use a very simple training set and simply ask it to learn.

3. It can become any kind of system. It can be for one thing, such as just a face recognition, or for another, such as an image reconstruction. It can be with a large number of weights, or with a very small number. It can be linear or nonlinear.

4. It will be much harder to determine where the flaws exist, or where it is creating false positives.

5. It is not affected by computation power. Hence, it can gain insights much more quickly and thus, it can tackle problems that are traditionally tricky to solve.

6. It has a high dimensionality. This means that we can create more learning models by adding more layers to our neural network.

7. It allows us to study the world as a non-supervised structure. If you look at neurons, they have such varied functions and shapes.

8. It can go and get a new image from its own memory.

9. It can adapt automatically to all data, but it makes for a nice alternative to traditional machine learning that relies on human expertise

10. It handles everything at a much higher level of abstraction than your standard neural network, so the training process is, at its core, much less complex.

11. It allows us to retain a lot of information, even on the basis of a very tiny or badly known object. And we are in the process of learning these ways of achieving efficiency for the vision.

12. It can see more than one and can learn with more information.

13. It gets its results more quickly. It learns over time rather than just in a flash.

14. It can learn over time, over billions of examples of images, and, crucially, recognize patterns.

15. It can be used in datasets that are too large, complex and repetitive for traditional computer systems.

16. It can handle large amounts of data for small networks with a much lower learning cost.

Applications of Deep Learning:

1.Automatic speech recognition

Large-scale automatic speech recognition is the first and most convincing successful case of deep learning. LSTM RNNs can learn "Very Deep Learning" tasks that involve multi-second intervals containing speech events separated by thousands of discrete time steps, where one time step corresponds to about 10 ms. LSTM with forget gates is competitive with traditional speech recognizers on certain tasks.

The initial success in speech recognition was based on small-scale recognition tasks based on TIMIT.

2.Image recognition

Deep learning-based image recognition has become "superhuman", producing more accurate results than human contestants. This first occurred in 2011 in recognition of traffic signs, and in 2014, with recognition of human faces.

3.Deep learning-trained vehicles now interpret 360° camera views.Another example is Facial Dysmorphology Novel Analysis (FDNA) used to analyze cases of human malformation connected to a large database of genetic syndromes.

4.Visual art processing Edit

Closely related to the progress that has been made in image recognition is the increasing application of deep learning techniques to various visual art tasks. DNNs have proven themselves capable, for example, of identifying the style period of a given painting.

5. Neural Style Transfer

Capturing the style of a given artwork and applying it in a visually pleasing manner to an arbitrary photograph or video generating striking imagery based on random visual input fields.

6. Virtual Assistants

Virtual Assistants are cloud-based applications that understand natural language voice commands and complete tasks for the user.Amazon Alexa, Cortana, Siri, and Google Assistant are typical examples of virtual assistants. They need internet-connected devices to work with their full capabilities. Each time a command is fed to the assistant, they tend to provide a better user experience based on past experiences using Deep Learning algorithms.

7. Chatbots

Chatbots can solve customer problems in seconds. A chatbot is an AI application to chat online via text or text-to-speech. It is capable of communicating and performing actions similar to a human. Chatbots are used a lot in customer interaction, marketing on social network sites, and instant messaging the client. It delivers automated responses to user inputs. It uses machine learning and deep learning algorithms to generate different types of reactions.

8. Healthcare

Deep Learning has found its application in the Healthcare sector. Computer-aided disease detection and computer-aided diagnosis have been possible using Deep Learning. It is widely used for medical research, drug discovery, and diagnosis of life-threatening diseases such as cancer and diabetic retinopathy through the process of medical imaging.

9. News aggregation and fake news detection

Deep Learning allows you to customize news depending on the readers' persona. You can aggregate and filter out news information as per social, geographical, and economic parameters and the individual preferences of a reader. Neural Networks help develop classifiers that can detect fake and biased news and remove it from your feed. They also warn you of possible privacy breaches.

10. Robotics

Deep Learning is heavily used for building robots to perform human-like tasks. Robots powered by Deep Learning use real-time updates to sense obstacles in their path and pre-plan their journey instantly. It can be used to carry goods in hospitals, factories, warehouses, inventory management, manufacturing products, etc.

.

CHAPTER-6

CONVOLUTIONAL NUERAL NETWORK

6.1 Introduction:

The first intention of trying to “understand the scene” is one of the base ideas in computer vision that lead to a continuous increase in the need to apprehend the high-level context in images regarding object recognition and image classification. By becoming a fundamental visual expertise that Computer Vision systems require, the field has rapidly grown. Images have become ubiquitous in a variety of fields as so many people and systems extract vast amounts of information from imagery. Information that can be vital in areas such as robotics, hospitals, self-driving cars, surveillance or building 3D representations of objects. While each of the above-mentioned applications differs by numerous factors, they share the common process of correctly annotating an image with one or a probability of labels that correlates to a series of classes or categories. This procedure is known as image classification and, combined with machine learning, it has become an important research topic in the field, on account of the focus on the understanding of what an image is representative of. The complex process of identifying the type of materials in diverse tasks linked to image-based scene perspectives has taken advantage of the combination of machine learning techniques applied to the up-to-date development of neural networks. This outlines the challenging problem of material classification due to the variety of the definite features of materials. The state of-the-art solutions rely massively on the attention that Computer Vision systems have received, which led to a series of algorithms being developed and images being collected in datasets.

Object detection and recognition are two main ways that have been implemented over multiple decades that are at the center of Computer Vision systems at the moment. These approaches are presented with challenges such as scale, occlusion, view point, illumination or background clutter, all issues that have been attempted as research topics that provided functionality that led to the introduction of Neural Networks and Convolutional Neural Networks .

6.2 Convolutional Neural Networks:

CNN models stand for one of the oldest deep neural networks hierarchy that have hidden layers among the general layers to help the weights of the system learn more about the features found in the input image. A general CNN architecture looks like the one shown in Figure 2 and consist of distinct types of layers. The convolutional layer applies an array of weights to all of the input sections from the image and creates the output feature map. The pooling layers simplify the information that is found in the output from the convolutional layer.

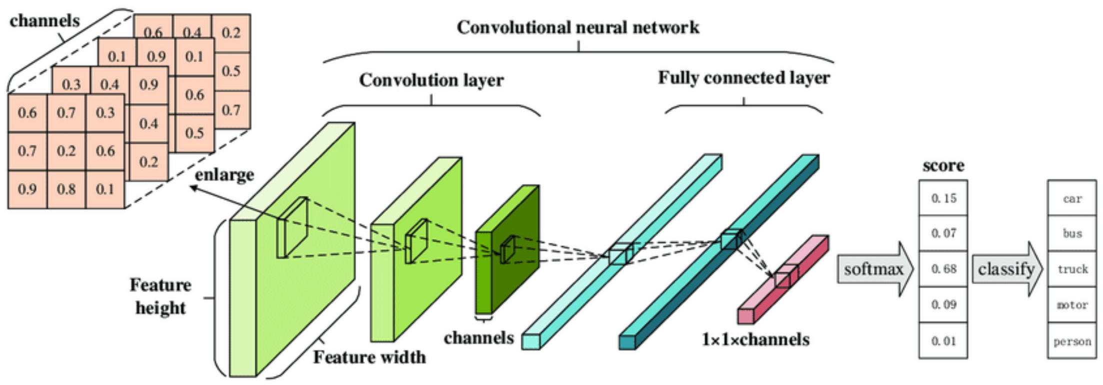


Fig 6.1:A General CNN layer hierarchy

The last layer is the fully connected layer that oversees the gathering of the findings from former layers and provides an N-dimensional vector, where N stands for the total number of classes. In the context of materials, it gives information on the texture of the material. Figure 6.1 gives a more descriptive identification of all the components of the pipeline, in particular the CNN architectures that this project uses for the challenging task of cancer classification

1. Convolutional Layer

The convolutional layer is the first layer of a convolutional network. While convolutional layers can be followed by additional convolutional layers or

pooling layers, the fully-connected layer is the final layer. With each layer, the CNN increases in its complexity, identifying greater portions of the image. Earlier layers focus on simple features, such as colors and edges. As the image data progresses through the layers of the CNN, it starts to recognize larger elements or shapes of the object until it finally identifies the intended object.
The convolutional layer is the core building block of a CNN, and it is where the majority of computation occurs. It requires a few components, which are input data, a filter, and a feature map. Let’s assume that the input will be a color image, which is made up of a matrix of pixels in 3D. This means that the input will have three dimensions—a height, width, and depth—which correspond to RGB in an image. We also have a feature detector, also known as a kernel or a filter, which will move across the receptive fields of the image, checking if the feature is present. This process is known as a convolution.

The feature detector is a two-dimensional (2-D) array of weights, which represents part of the image. While they can vary in size, the filter size is typically a 3x3 matrix; this also determines the size of the receptive field. The filter is then applied to an area of the image, and a dot product is calculated between the input pixels and the filter. This dot product is then fed into an output array. Afterwards, the filter shifts by a stride, repeating the process until the kernel has swept across the entire image. The final output from the series of dot products from the input and the filter is known as a feature map, activation map, or a convolved feature.

The weights in the feature detector remain fixed as it moves across the image, which is also known as parameter sharing. Some parameters, like the weight values, adjust during training through the process of backpropagation and gradient descent. However, there are three hyperparameters which affect the volume size of the output that need to be set before the training of the neural network begins. These include:

1. The number of filters affects the depth of the output. For example, three distinct filters would yield three different feature maps, creating a depth of three.

2. Stride is the distance, or number of pixels, that the kernel moves over the input matrix. While stride values of two or greater is rare, a larger stride yields a smaller output.

3. Zero-padding is usually used when the filters do not fit the input image. This sets all elements that fall outside of the input matrix to zero, producing a larger or equally sized output. There are three types of padding:

Valid padding: This is also known as no padding. In this case, the last convolution is dropped if dimensions do not align.

Same padding: This padding ensures that the output layer has the same size as the input layer

Full padding: This type of padding increases the size of the output by adding zeros to the border of the input.

Ultimately, the convolutional layer converts the image into numerical values, allowing the neural network to interpret and extract relevant patterns.

2. **Pooling Layer**

Pooling layers, also known as downsampling, conducts dimensionality reduction, reducing the number of parameters in the input. Similar to the convolutional layer, the pooling operation sweeps a filter across the entire input, but the difference is that this filter does not have any weights. Instead, the kernel applies an aggregation function to the values within the receptive field, populating the output array. There are two main types of pooling:

1.Max Pooling

As the filter moves across the input, it selects the pixel with the maximum value to send to the output array. As an aside, this approach tends to be used more often compared to average pooling.

2.Average Pooling

As the filter moves across the input, it calculates the average value within the receptive field to send to the output array.
While a lot of information is lost in the pooling layer, it also has a number of benefits to the CNN. They help to reduce complexity, improve efficiency, and limit risk of overfitting.

3. **Fully – Connected Layer**

The name of the full-connected layer aptly describes itself. As mentioned earlier, the pixel values of the input image are not directly connected to the output layer in partially connected layers. However, in the fully-connected layer, each node in the output layer connects directly to a node in the previous layer.

This layer performs the task of classification based on the features extracted through the previous layers and their different filters. While convolutional and pooling layers tend to use ReLu functions, FC layers usually leverage a softmax activation function to classify inputs appropriately, producing a probability from 0 to 1.

6.3 CNN Architecture :

The first CNN architecture is the VGG_CNN_F model that has eight learned layers with five convolutional and two fully-connected, where “F” stands for fast. It is called the fast model due to the first layer having a stride of four pixels with 64x11x11 dimensions. The last layer gives an output encompassing around 4096 features detected in the input image that are further passed along to the classifier. The second CNN architecture is the VGG_CNN_M model that it is called medium because its algorithm trains slower. This is due to the first layer having a stride two pixel that will change and decrease down to a stride one pixel in its third convolutional layer. Same as the CNN_F architecture, the last layer also outputs 4096 features. The third CNN is the VGG_CNN_S that has a similarly built structure like the abovementioned architectures, which the distinction of the stride decreasing to the second convolutional layer instead of the third convolutional layer. This affects the training phase by slowing it down. These architectures can be studied in Figure 6.2.

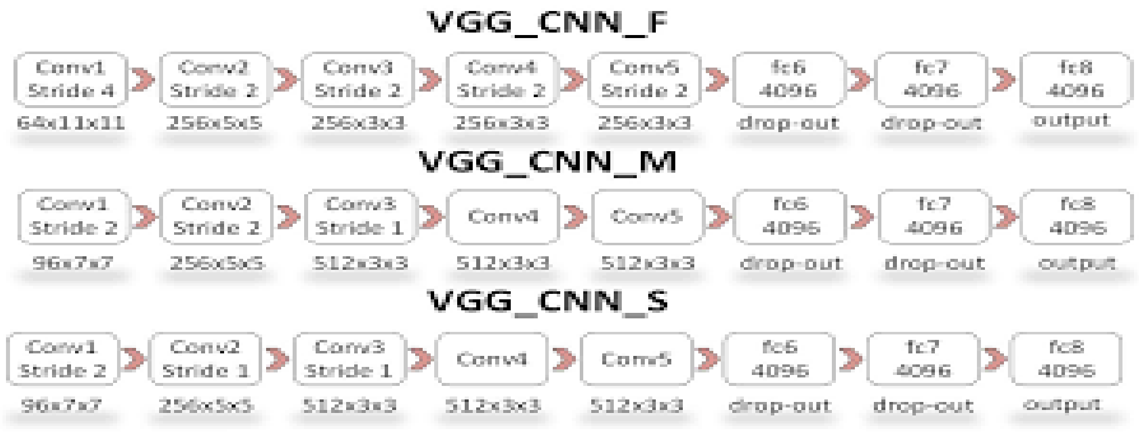


Fig 6.2 VGG CNN architectures overview

A new state-of-the-art architecture called GoogLeNet entered the scene and provided an improved network with larger depths and widths as well as fewer parameters used. With its 22 layers, the lower layers are formed of convolutional layers used to solve the performance issue caused by the expansion in the depth of the network, and the higher layers consist of the pooling layers which are introduced at all stages in the network and 9 brand-new type of hierarchies called inception that are modules stacked one upon another used for dimension reduction and can be up to 100 layers each. Compared to VGG_CNN_F . it uses less parameters within its network and offers an alternative to using fully connected layers. Since it presents a new type of architecture where models do not need to have the layers structured sequentially, it is extremely important that it is one of the CNN models tested. The fifth and sixth CNN models used are VGG16 and VGG19 very similar to the first three types of architectures consisting of eight learned layers. The new models test their deeper network on the ImageNet7 dataset and achieve better results than GoogLeNet and VGG_CNNs where the architecture's test error is around 7.0% compared to 7.9% outputted by GoogLeNet.

Two noticeable characteristics are:

1. The stride is fixed to one pixel across the majority of the network and
2. The convolutional layer is not followed by a pooling layer at all stages in the hierarchy. Although the configuration of the fully connected layers is the same in all networks, the first two levels gather 4096 features, while the third one only has 1000 channels. Since these models are mainly trained and tested on only one database, the question is whether they will achieve similar results on other datasets and generalize as good for other tasks. Furthermore, a data augmentation technique has been used during the training of the network. The structure of these three architectures can be seen in Figure 6.3

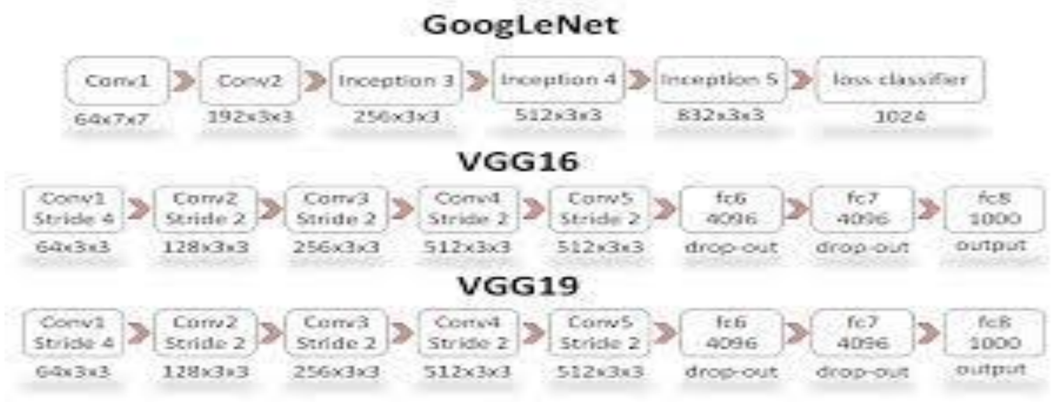


Fig 6.3: GoogLeNet and VGG architectures overview

6.4 Applications of CNN:

1. Image recognition

CNNs are often used in image recognition systems. In 2012 an error rate of 0.23% on the MNIST database was reported.When applied to facial recognition, CNNs achieved a large decrease in error rate.

2. Video analysis

Compared to image data domains, there is relatively little work on applying CNNs to video classification. Video is more complex than images since it has another (temporal) dimension. However, some extensions of CNNs into the video domain have been explored. One approach is to treat space and time as equivalent dimensions of the input and perform convolutions in both time and space. Another way is to fuse the features of two convolutional neural networks, one for the spatial and one for the temporal stream.

3. Anomaly Detection

A CNN with 1-D convolutions was used on time series in the frequency domain (spectral residual) by an unsupervised model to detect anomalies in the time domain.

CHAPTER-7

MODEL TRAINING

7.1 Building a DL model:

Deep learning is an increasingly popular subset of machine learning. Deep learning models are built using neural networks. A neural network takes in inputs, which are then processed in hidden layers using weights that are adjusted during training. Then the model spits out a prediction. The weights are adjusted to find patterns in order to make better predictions. The user does not need to specify what patterns to look for the neural network learns on its own.

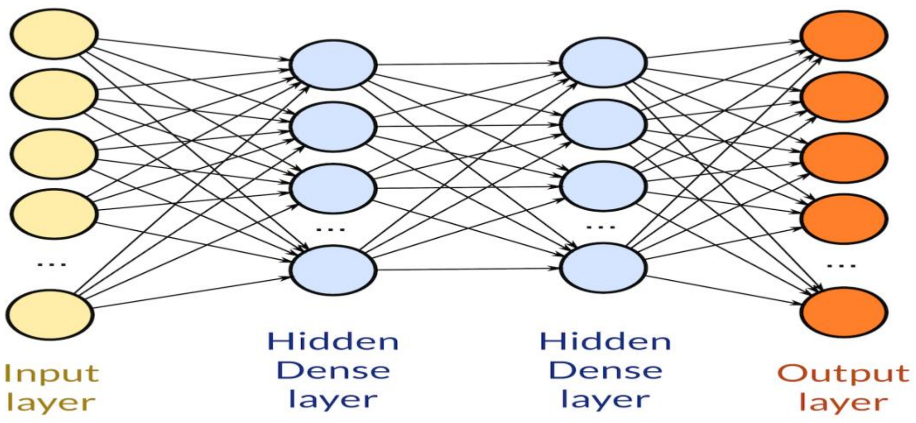


Fig 7.1: Deep Learning model

1. Importing libraries :

```
import pandas as pd
from sklearn.ensemble import RandomForestClassfier
from sklearn.feature_selection import SelectFromModel
```

2. In all feature selection procedures, it is a good practice to select the features by examining only the training set. This is to avoid overfitting. So considering we have a train and a test dataset. We select the features from the train set and then transfer the changes to the test set later.

```
X_train,y_train,X_test,y_test = train_test_split(data,test_size=0.3)
```

3.The model fitting and feature selection altogether in one line of code. Firstly, specify the random forest instance, indicating the number of trees. Then use selectFromModel object from sklearn to automatically select the features.

```
sel = SelectFromModel(RandomForestClassifier(n_estimators = 100))
sel.fit(X_train, y_train)
```

SelectFromModel will select those features which importance is greater than the mean importance of all the features by default, but we can alter this threshold if we want.

4. To see which features are important we can use get_support method on the fitted model.

```
sel.get_support()
```

It will return an array of boolean values. True for the features whose importance is greater than the mean importance and False for the rest.

5. Now make a list and count the selected features.

```
selected_feat= X_train.columns[(sel.get_support())] len(selected_feat)
```

It will return an Integer representing the number of features selected by the random forest.

6. To get the name of the features selected

```
print(selected_feat)
```

It will return the name of the selected features.

7. We can also check and plot the distribution of importance.

pd.series(sel.estimator_,feature_importances_,.ravel()).hist()

It will return a histogram showing the distribution of the features selected using this feature selection technique.

7.2 Python :

Python is a programming language developed by Guido Van Rossum (1991). The Python language mainly organized an object-oriented process with the goal to help the programmers in various aspects such as clear writing, logical code for mini and big projects. Usually, Python supports various programming paradigms at the same time mainly technical, object-oriented, and efficient programming. Python was considered as a replacement to the ABC language in 1980. Python 2.0, contains new features such as the list of understandings and a trash collection system that is able to collect reference cycles and released in 2000, whereas Python 3.0, is not completely backward-compatible and released in 2008.

7.3 Open CV :

Open CV is developed by Intel, primarily attentive on the real-time data with computer vision. The open CV is an open-source code for a computer vision library that provides free uses and cross-platform with an open-source license.

7.4 TensorFlow:

TensorFlow is also a library (open-source) for various computer programming with a range of variety of tasks. TensorFlow is the free representative mathematical library, which was used for ML functions mainly neural networks (Abadi, Barham, et al., 2016). Google Brain team developed the TensorFlow library for core uses like mainly for research and production (Abadi, Agarwal, et al., 2016). TensorFlow was distributed under the Apache License 2.0 on 9th November,2015.

7.5 KERAS:

Keras is a neural network library (open-source) and coding in Python programming language that able to run on most of the high levels of TensorFlow, Theano, or Microsoft Cognitive Toolkit. Usually, Keras developed to facilitate rapid experimentation using deep neural-networks with user-approachable, modular, and extensible. The Keras was established by François Chollet, a Google engineer in the research project of ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System). Chollet is also one of the authors of the XCeption deep neural network model (Chollet, 2017). Interestingly, Google's TensorFlow supporting Keras in TensorFlow's core public library in back 2017. Cholletsuggested that the Keraswas considered being an edge rather than a self-governing ML-framework. Keras library provides a higher-level approach, an extra impulsive set of concepts that create it simple to develop DL based models irrespective use of computational backend. In this context, Microsoft incorporatesCNTK backend to Keras, which is available as of CNTK v2.0.

Keras library comprises various neural-network-based operations including types of layers, objectives, activation functions, optimizers, and host tools that make suitable for working with image and text data mining. The code is introduced on GitHub, and community support forums that contain GitHub issues, and slack-channel. Keras support for convolutional, persistent neural-networks, and supplementary general utility layers (dropout, batch normalization, and pooling). Furthermore, Kerasgive consents to his users for the development of deep models on smartphones like iOS and Android, on the web, and Java Virtual Machine. Additionally, Keras permit the use of circulated training of DL models on clusters of GPUs and Tensor processing units (TPUs). In recent times, Keras claims more than 200,000 users as 2017 and 10thutmost cited tools in the KD Nuggets (2018) software poll and also approximately 22% usage. Therefore, DL neural-networks with Keras library is an essential tool for AI-based technologies.

7.5.1 Why we focus on Keras Library :

There are various reason for using Keras library compare with that of other existing alternatives:

(1) Keras is an application program interface (API), mainly designed for human not for machines that monitors superlative practices for reducing intellectual load,

(2) Keras library provides reliable and simple APIs that reduces required number of user actions and also offers clear and unlawful feedback upon IGI GLOBAL PROOF may not be copied or distributed 7 An Introduction to Deep Convolutional Neural Networks WithKeras error, thereby more productive,

(3) Keras combined with the lower-level DL languages mainly TensorFlow that enables to implement anything like tf.keras, the Keras API impeccably with your TensorFlow workflows. Therefore, Keras library is used research and industry. Figure. 2 shows the ranking of different DL frameworks. Approximately more than 250,000 individual users of Keras registered as 2018 due to stronger approval in the industry and research compare with that of other DL framework except for TensorFlow. Keras is also a favorite choice among DL researchers like CERN, and NASA due to its advantages over other DL frameworks. Keras is a user-friendly neural network library written in Python.

7.6 Training the model:

Now we will train our model. To train, we will use the ‘fit()’ function on our model with the following five parameters: training data (train_X), target data (train_y), validation split, the number of epochs and callbacks.

The validation split will randomly split the data into use for training and testing. During training, we will be able to see the validation loss, which give the mean squared error of our model on the validation set. We will set the validation split at 0.2, which means that 20% of the training data we provide in the model will be set aside for testing model performance.

The number of epochs is the number of times the model will cycle through the data. The more epochs we run, the more the model will improve, up to a certain point. After that point, the model will stop improving during each epoch. In addition, the more epochs, the longer the model will take to run. To monitor this, we will use ‘early stopping’.

Early stopping will stop the model from training before the number of epochs is reached if the model stops improving. We will set our early stopping monitor to 3. This means that after 3 epochs in a row in which the model doesn’t improve, training will stop. Sometimes, the validation loss can stop improving then improve in the next epoch, but after 3 epochs in which the validation loss doesn’t improve, it usually won’t improve again.

Sample Code:

```
fromkeras.callbacksimportEarlyStopping
#set early stopping monitor so the model stops training when it won't improve anymore
early_stopping_monitor = EarlyStopping(patience=3)
#train model
model.fit(train_X, train_y, validation_split=0.2, epochs=30,
callbacks=[early_stopping_monitor])
```

7.7 Predicting the data:

If you want to use this model to make predictions on new data, we would use the ‘predict()’ function, passing in our new data. The output would be ‘malginant/benign’ predictions.

Sample code:

```
test_y_predictions = model.predict(test_X)
```

CHAPTER 8

UI/UX Design & Web Server

8.1 Introduction

The UI stands for user interface which is the first priority for development of any application where the user interacts to full fill his desire.The goal of effective UI is to make the user's experience easy and intuitive, requiring minimum effort on the user's part to receive maximum desired outcome.

UI is created in layers of interaction that appeal to the human senses (sight, touch, auditory and more). They include both input devices like keyboard, mouse, trackpad, microphone, touch screen, fingerprint scanner, e-pen and camera and output devices like monitors, speakers and printers. Devices that interact with multiple senses are called "**multimedia user interfaces**". For example, everyday UI uses a combination of tactile input (keyboard and mouse) and a visual and auditory output (monitor and speakers).

8.2 Types of UI

Graphical User Interface

A tactile UI input with a visual UI output.

Touch User Interface

User interface through haptics or touch. Most smartphones, tablets and any device that operates using a touch screen use haptic input.

Command Line Interface

A command-line interface (CLI) is a text-based user interface (UI) used to run programs, manage computer files and interact with the computer.

8.3 Importance of UI & UX

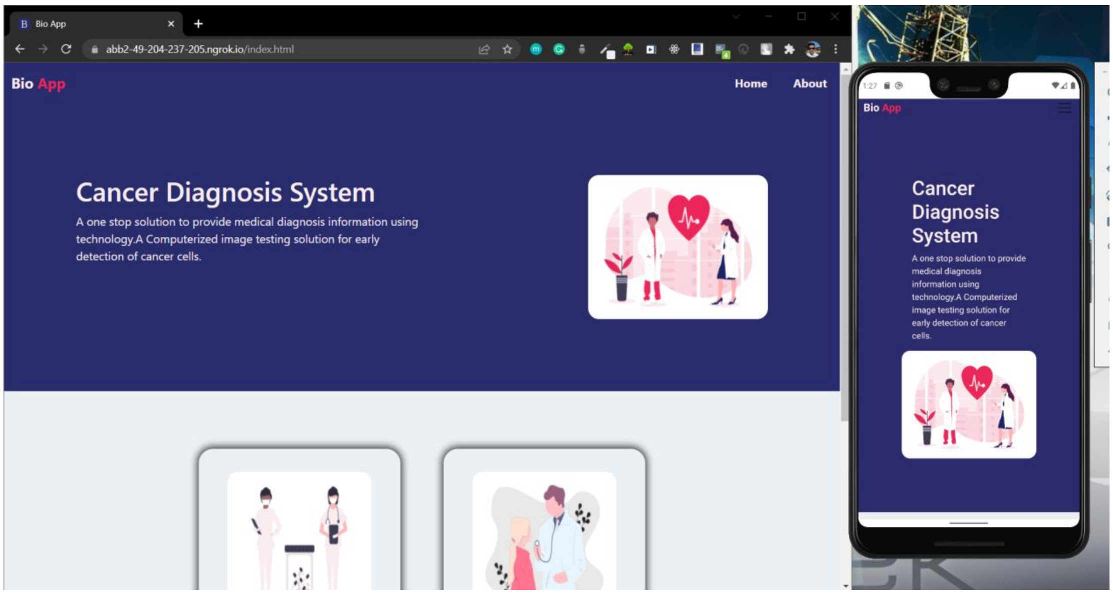
User interface is important to meet user expectations and support the effective functionality of your site. A well-executed user interface facilitates effective interaction between the user and the program, app or machine through contrasting visuals, clean design and responsiveness.

User interface and user experience are related and equally important to the execution of a project, but the specifics differ. Mainly, UI is designed around the intended look and feel of the site, app or program while UX spans the entire process of conceptualization, development and delivery. Additionally, UX can be referenced in relation to nearly any product, while UI can only pertain to digital products.

8.4 Responsive Design-

The responsive design deals with design and development of UI which is flexible to any screen size. In responsive design the UI components automatically adjusts it's position depending on the screen size. This responsive design helps in accessing the website or App on any screen size which ultimately improves user experience(UX).

The responsiveness of the application is achieved with the help of **flexbox technology** integrated to the UI which makes it flexible to any screen size. The user can interact with the application without loosing any information on the application UI.



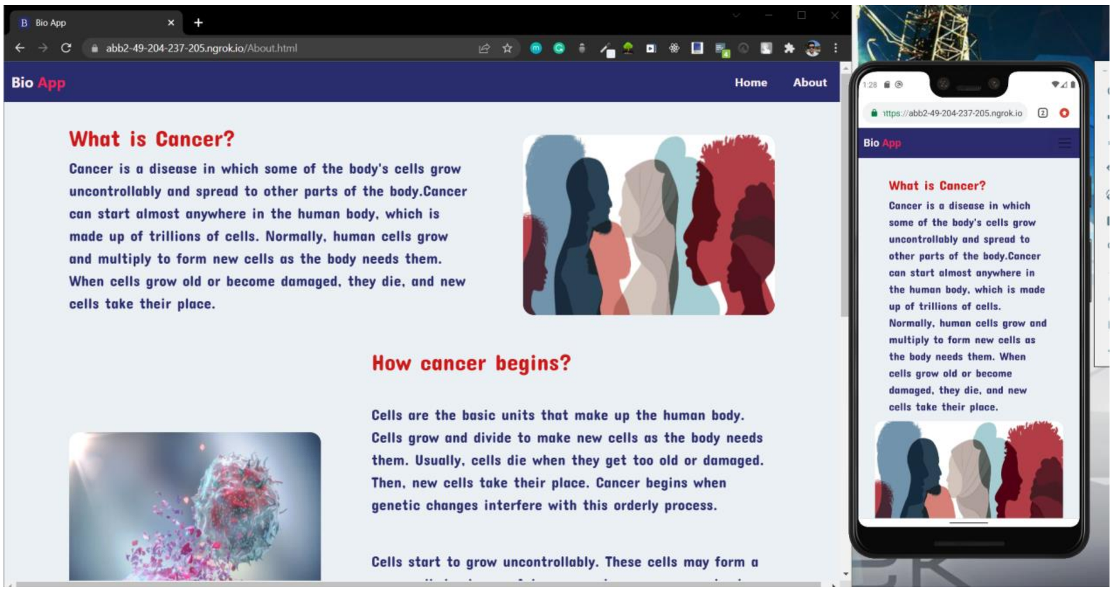
Cancer Diagnosis System

A one stop solution to provide medical diagnosis information using technology.A Computerized image testing solution for early detection of cancer cells.

Cancer Diagnosis System

A one stop solution to provide medical diagnosis information using technology.A Computerized image testing solution for early detection of cancer cells.

Fig 8.1: Web/Android User Interface



What is Cancer?

Cancer is a disease in which some of the body's cells grow uncontrollably and spread to other parts of the body.Cancer can start almost anywhere in the human body, which is made up of trillions of cells. Normally, human cells grow and multiply to form new cells as the body needs them. When cells grow old or become damaged, they die, and new cells take their place.

How cancer begins?

Cells are the basic units that make up the human body. Cells grow and divide to make new cells as the body needs them. Usually, cells die when they get too old or damaged. Then, new cells take their place. Cancer begins when genetic changes interfere with this orderly process.

Cells start to grow uncontrollably. These cells may form a

What is Cancer?

Cancer is a disease in which some of the body's cells grow uncontrollably and spread to other parts of the body.Cancer can start almost anywhere in the human body, which is made up of trillions of cells. Normally, human cells grow and multiply to form new cells as the body needs them. When cells grow old or become damaged, they die, and new cells take their place.

Fig 8.2: About Cancer Page

8.5 Introduction to Web Server-

Flask is backend python framework which is used to build web applications.A Web-Application Framework, often known as a Web Framework, is a set of modules and libraries that allow programmers to create apps without having to write low-level code like protocols and thread management. Flask uses the WSGI tools and the Jinja2 template engine.In this project flask framework is used as a to handle backend server and connected to the frontend part of the web application .

8.6 Model Integration-

In this project flask framework is used to serve as a web server to our application .Initially the model files are created with the best algorithm which was gave best accuracy. After the model files are being created then its loaded into the server file which is used for predicting breast cancer and lung disease. The trained models which are used for predicting breast cancer and lung disease is integrated with frontend of the application and the backend is handled by the flask server.The request is made to the server and in response the document is being rendered by the browser.

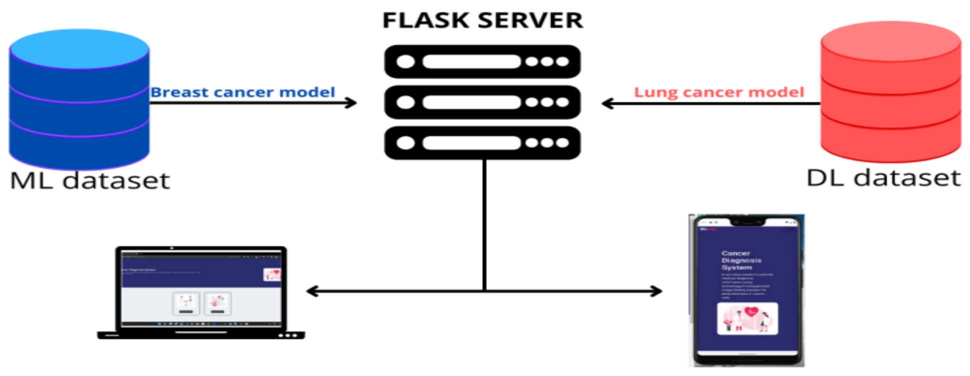


Fig 8.3: Model Integration to the server

8.7 Front End Integration-

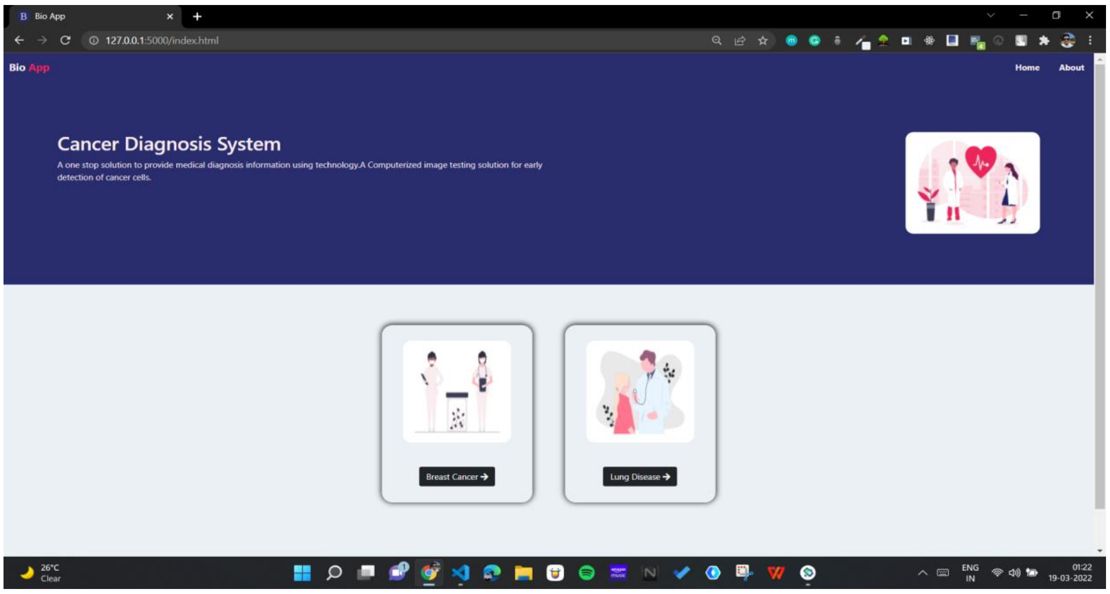
The front end of the application is built using html,css and javascript and used flexbox technology to make it responsive.Different routes are created which are being served by the server at the backend. The flask server supports REST Api architecture which supports get,post ,put delete,patch request methods. In this project we have used two different request methods which are get and post methods.

CHAPTER 9

RESULTS

The results obtained are pretty accurate and as expected below are the figures of user interface and outputs obtained.

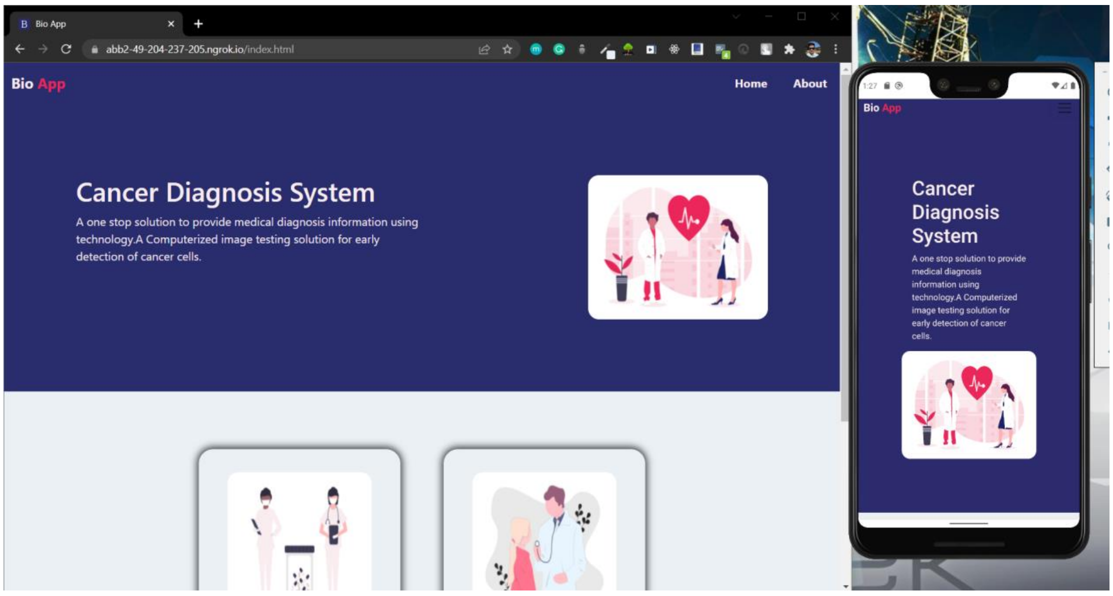
9.1 WEB/ ANDROID UI



Cancer Diagnosis System

A one stop solution to provide medical diagnosis information using technology.A Computerized image testing solution for early detection of cancer cells.

Fig 9.1 Web User Interface



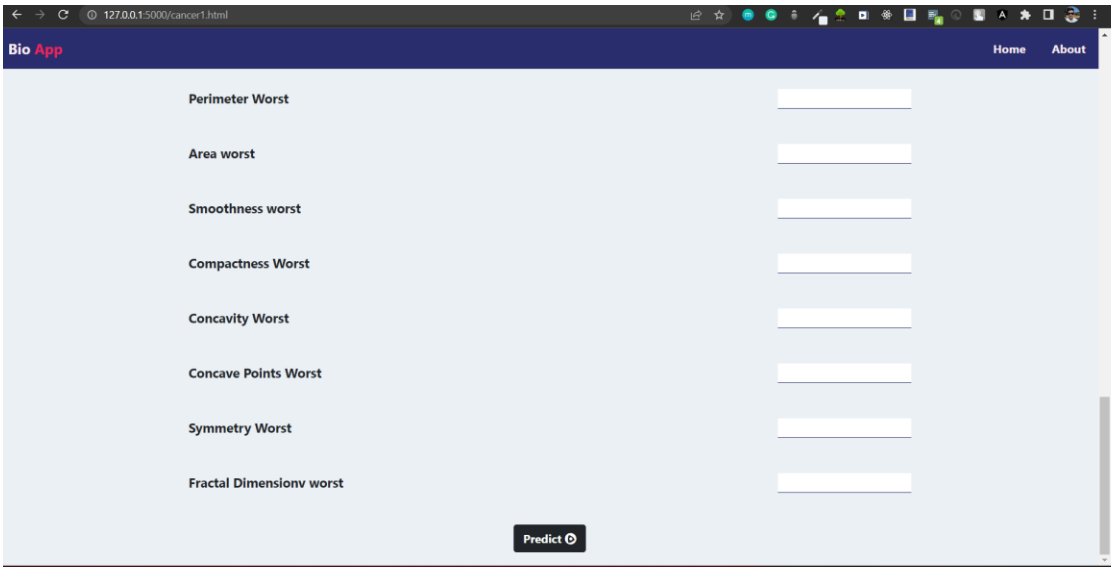
Cancer Diagnosis System

A one stop solution to provide medical diagnosis information using technology.A Computerized image testing solution for early detection of cancer cells.

Fig 9.2 Web/Android User Interface

9.2 FEATURE 1 BREAST CANCER-

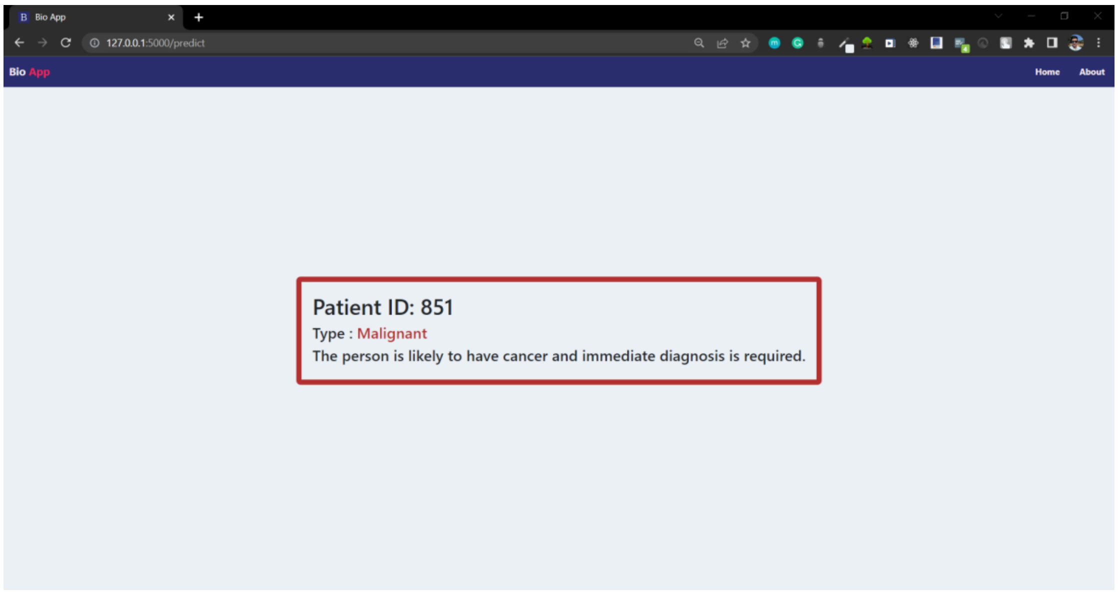
The user has to enter the details in the input field and click on predict feature button to view the result.



Perimeter Worst	
Area worst	
Smoothness worst	
Compactness Worst	
Concavity Worst	
Concave Points Worst	
Symmetry Worst	
Fractal Dimensionv worst	

Fig 9.3 Feature 1 input form

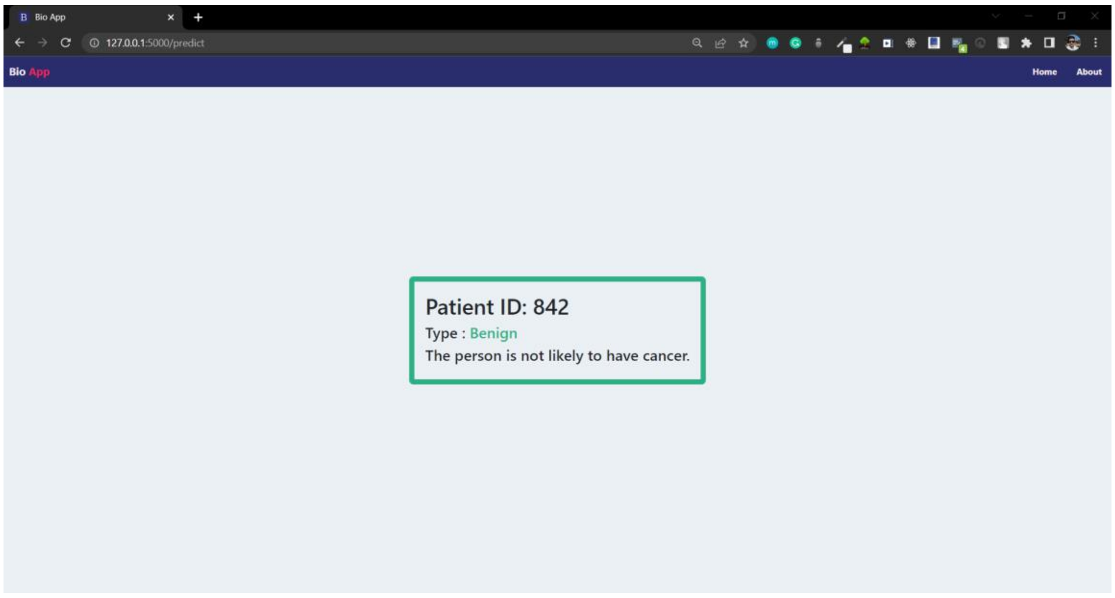
When the patient has malignant type tumor then the result is displayed like shown below.



Patient ID: 851
Type : Malignant
The person is likely to have cancer and immediate diagnosis is required.

Fig 9.4 Feature 1 Result Malignant

When the patient has benign type tumor then the result is displayed like shown below.



Patient ID: 842
Type : Benign
The person is not likely to have cancer.

Fig 9.5 Feature 1 Result Benign

9.3 FEATURE 2 LUNG DISEASE-

The user has to upload an image by clicking the select file feature button and then a window pops up where the user has to navigate to the image file location and select the file .

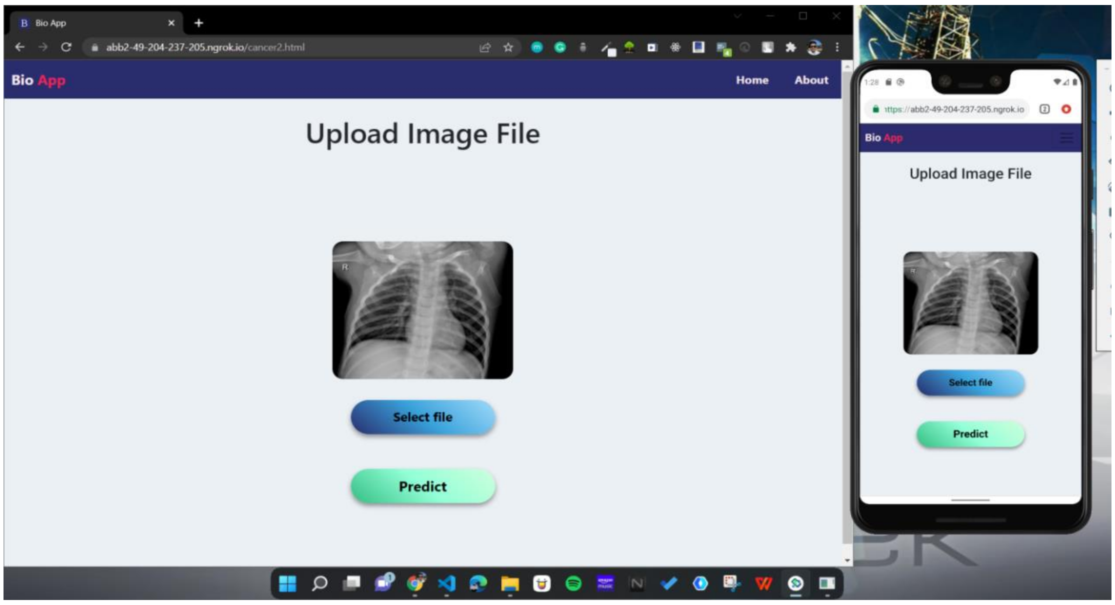


Fig 9.6 Feature 2 Upload Image UI

After selecting the file the user has to click on predict feature button to get the results.

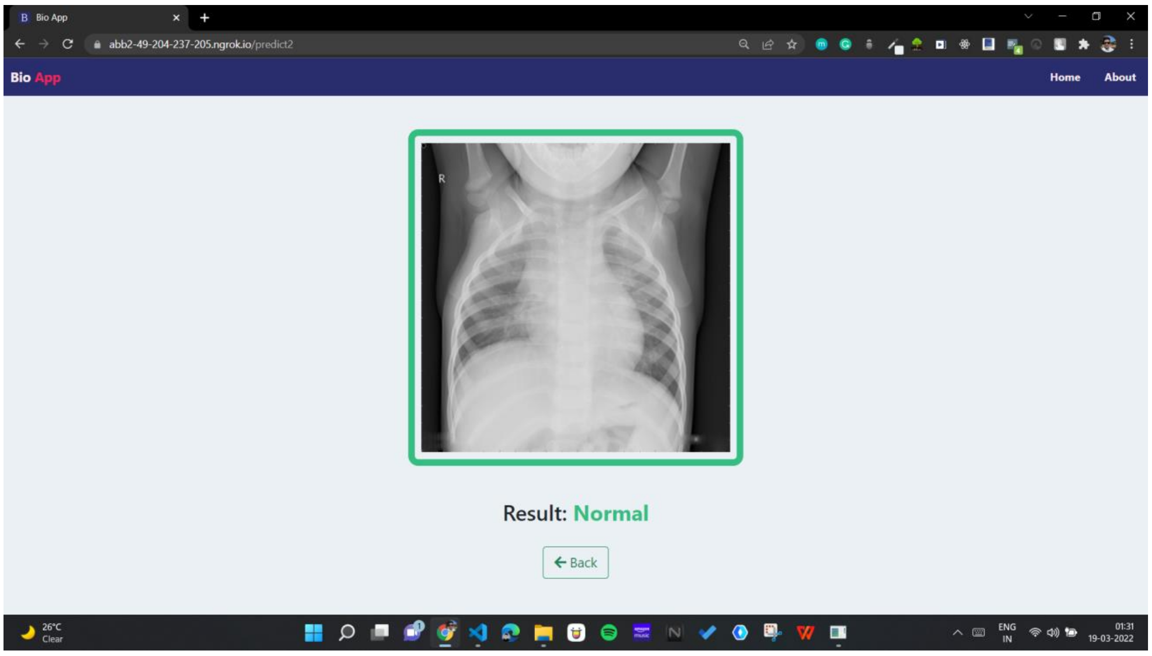


Fig 9.7 Feature 2 Result Normal

The above image depicts that the person is not having any lung disease and after getting the result the user can go back to check for another image files by clicking on the back button.

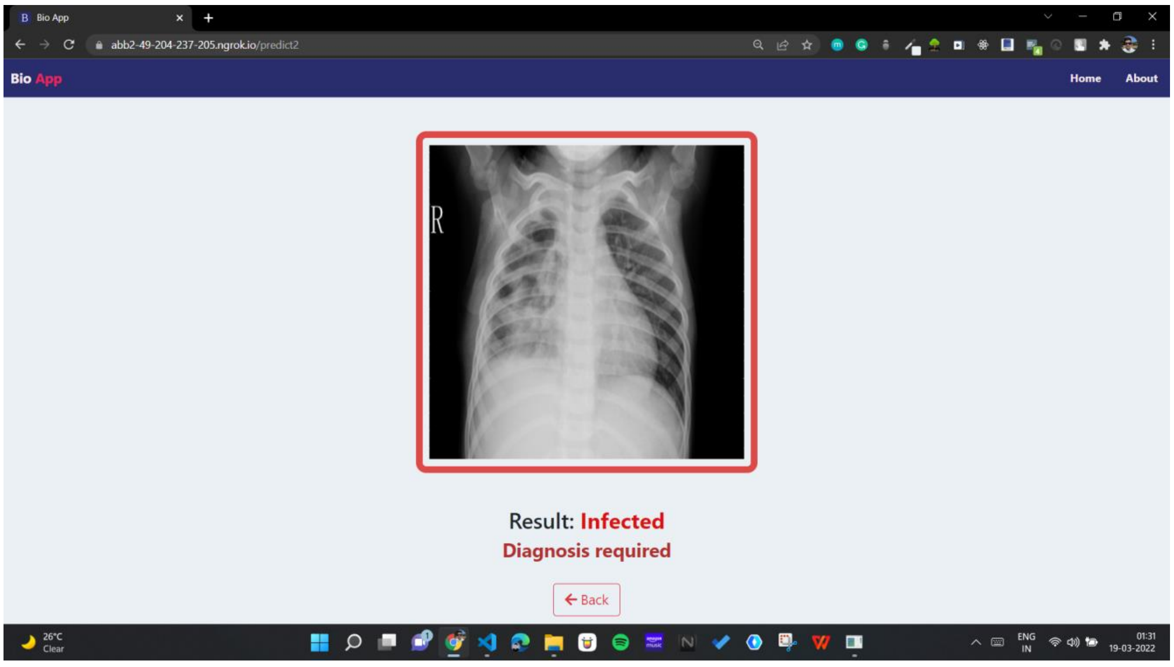


Fig 9.8 Feature 2 Result Infected

The above image shows that the person is having lung disease which could lead to lung cancer so immediate diagnosis is required.

CHAPTER 10

CONCLUSION

We have developed an application which has two features where feature 1 is predicting breast cancer and feature 2 is predicting lung disease which is a primary symptom for lung cancer. We have used deep computer science techniques which includes artificial intelligence , Deep learning and machine learning that solves the major problem in health care sector that is predicting cancer. We have a proposed a model which takes data or image files as an input and predicts whether the person is having cancer or not at early stage and the prediction accuracy is around 96% . The application is developed using modern technology which can run on any computing device which has browser capability and it is scalable. We can integrate as many features we want that can be solved with the help of computers in health care domain.

CANCER PREDICTION WEB APP USING MACHINE LEARNING AND DEEP LEARNING

P.Devi Pradeep, J.Bhaskara Rao, Sai Kiran Patro, G. Venkat Naidu, T. Sannath Kumar, G. Nishkala

Department of ECE, Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam, India
pradeep.ece06@gmail.com, janabhaskar@gmail.com, kiranpatro.2018.ece@anits.edu.in,naidugorli.2018.ece@anits.edu.in, sannathkumar.2018.ece@anits.edu.in, nishkala.2018.ece@anits.edu.in

Keywords:
Mobile ,Web application , Deep learning , Machine Learning , GUI, Keras.

A B S T R A C T

In today's world, image signal processing and ML, AI are the most popular technologies, with applications ranging from automation to healthcare. Machine learning is one of the core fundamentals of AI which can solve real life problems. Healthcare domain is one of the major concern where AI, ML , DL can solve the problems. This project describes with user interface (UI) .The idea is to implement features for bio-medical applications. The UI consists of different buttons and the user has to upload the file which should be in the form of any image format or data and the output will be displayed on a new window.This program has two functions: one for predicting breast cancer and the other for predicting lung cancer. This app is available for both mobile and online use.The technology used are Web and subsets of AI which are DL, ML . The libraries which used are Sklearn, Matplotlib, numpy , pandas,keras etc .

1. INTRODUCTION

Health care is one of the major concern and as per the records cancer is one the most common health problem which is identified in Indians. Nearly 1 million people are being affected by cancer. Technology can play a vital role in predicting cancer cells before it gets worsen.In 2018, cancer was the 2nd leading cause of death globally for 9.6 million deaths, or one out of every six deaths.
To reduce the death rate of patients, the primary contact with cancer is required.Cancer is very dangerous because it can grow on any organ, tissue and can spread to other organs.Breast cancer has become one of the most prevalent causes of mortality among women. Tumor classification can be used to diagnose breast cancer. Tumors are divided into two categories: malignant type and benign type tumors.To distinguish between these malignancies, doctors require a reliable diagnostic process. However, even professionals find it difficult to recognise malignancies in most cases. As a result, diagnosis requires the automation of diagnostic systems. Breast cancer, being the most frequent cancer in women, has a high incidence and mortality rate in the past.Detecting cancer cells is the process of determining whether or not a person has cancer.In the early 1980s, a CAD (Computer-Aided Diagnosis) was created to improve the survival rate and efficiency of clinicians while evaluating medical pictures. Decision trees, linear regression, random forest, K-nearest neighbors, and other ML algorithms have a significant influence in health care.

2. PROBLEM IDENTIFICATION

To determine the type of tumor using the most accurate machine training and deep training methods.The graphical user interface is designed and integrated with models for the ease of use to user.

3. PROPOSED METHODOLOGY

The Project is implemented using Vscode as a code editor to code and retrieve the User Interface output from the flask as a local server, and utilized the breast cancer dataset and lung cancer photos from kaggle. The project employs supervised learning algorithms and classification techniques such as Random Forest, Decision Tree, and CNN architecture, a deep learning algorithm, in our process.

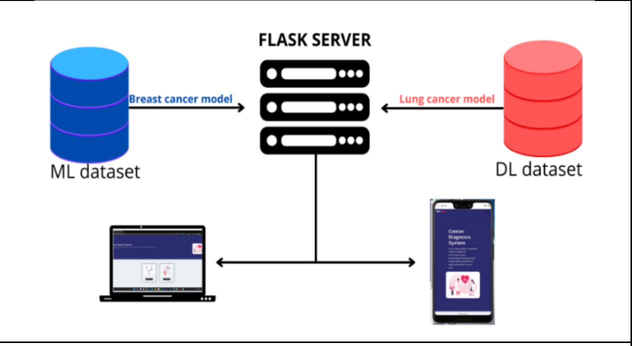


Figure 1. Proposed Methodology

3. 1.Data Updation

The data is downloaded from kaggle is in dictionary format in excel sheet for breast cancer and images for lung cancer which is useful for deep learning algorithm. The dataset for feature 1 consists of statistical values and it has one dependent target column which depends on other independent vectors. The target or dependent column contains "Malignant" and "Benign" type tumor.The feature 2 consists of images and classified into two types one is normal and another one is infected type and the images are augmented so that we get more images from the exsisting images for training our DL model.

3.2 Graphical Representation of Data

The graphical depiction of information and data is known as data representation . Graphical representation of data will make make it easy to examine and comprehend trends, outliers, and patterns in data by employing visual components like charts, graphs, and maps.

Gathering data and interpreting it is the first step before creating the model. The data is visualized thoroughly and altered according to the model requirements.

3.2.1 Dataset for Breast cancer

The data is in Numerical , thus we must create a pair plot of our dataset, which is already divided into two groups, benign 1 and malignant 0.Now we've used our dataset's counter plot to tally how many patients had benign and malignant tumors in total.

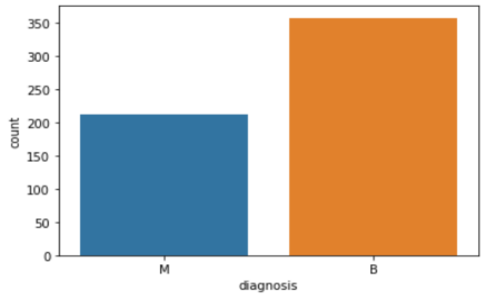


Figure 2 . Total count of malignant and benign tumor patients in counter-plot

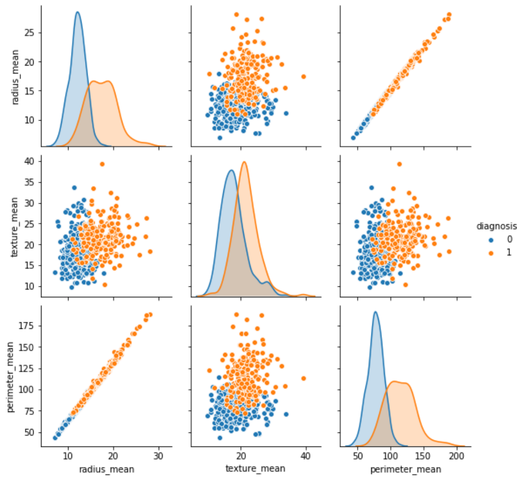


Figure 3. Dataset's counter plot

Now we've used our dataset's counter plot to tally how many patients had benign and malignant tumors in total. So the number of malignant tumor cases is 220-230, whereas the number of benign tumour cases is much higher than the number of malignant tumour cases.

3.2.2 Image Set For Pneumonia Lung Disease

CNN which is a deep learning algorithm requires image data set to train and test for accuracy. The image set contain more than 5000 images which includes both normal and infected images. Below table depicts the count of images in test and train path.

TABLE 1. Count of Images for the Test and Train of Machine

Type	Normal	Infected
Train	1341	3875
Test	234	390

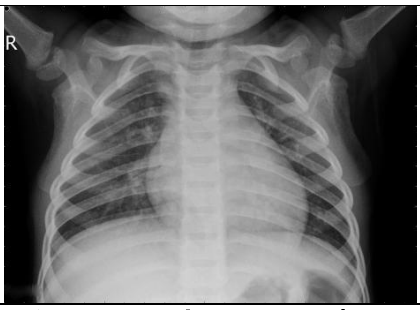


Figure 4. Normal X-ray image of Lung

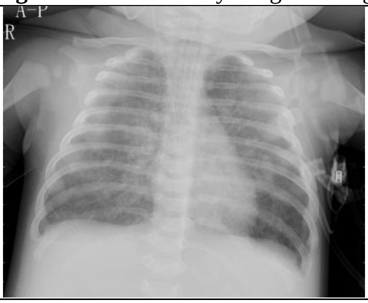


Figure 5. Infected x-ray image of Lung

4. MODEL SELECTION

This is the most significant step in the development of a system when Data Scientists apply various sorts of ML and DL Algorithms. There are two types of ML algorithms: supervised and unsupervised. Deep learning is subset of ML which simply tries to mimic human behaviour. For feature one which is predicting breast cancer the supervised learning technique is used and chose the one which gave maximum accuracy and for feature 2 which is prediction of Lung disease the VGG16 CNN architecture is used.

4.1 Feature 1 Breast Cancer

Feature 1 of this web application is prediction of breast cancer . Supervised learning Technique is used.The supervised learning algorithm learns from the training data, allowing you to forecast unpredicted data outcomes. It aids in the optimization of performance requirements via experience, as well as the solution of numerous sorts of real-world computation issues, and such classifiers are briefly detailed below.

I) Logistic Regression

Logistic Regression is a supervised ML algorithm which has independent vectors and one dependent which needs to be predicted. In our project feature 1 the dependent variable is type of tumor which is dependent on different parameters.The result will contain either 0 or 1 which means no or yes.

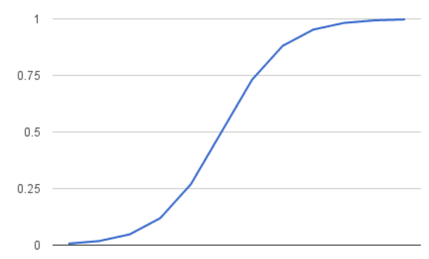


Figure 6. Prediction range of Logistic regression

II) Decision Tree

The goal of our project is to use the Decision Tree algorithm to diagnose breast cancer. The tree method is an example of supervised learning. Because decision trees need little processing, they are quick to implement and have poor accuracy.

III) Random Forest

Random Forest is ML algorithm which consists of decision Trees and out of all decision trees which gives is maximum accuracy is considered.It takes more processing but gives best performance accuracy. In this project accuracy we got is 97.3%.

4.2 Feature 2 Lung Disease

Feature 2 of this web application is prediction of Lung disease pneumonia which is symptom of lung cancer. A deep learning technique is used for prediction and model is trained with the image set from kaggle. The performance of machine learning classifiers trained with features extracted by VGG16DCNNdiagnoseof pneumonia from chest x-ray pictures is investigated in this research.

The VGG16 DCNN with weights learned on the ImageNet database is utilised to extract features from the pictures in this investigation. The ImageNet database contains more than 14 million photos divided into 1000 categories. Because pre-trained models like VGG16 have previously learnt to extract features from photos as well as discriminate between images of various classes, they have performed well when applied to datasets with comparable domains. Because VGG16 DCNN outperformed other pre-trained models in medical imaging, it was employed for feature extraction in this investigation.VGGNet or VGG16 DCNN includes 144 million parameters and 16 convolutional layers with 3 x 3 receptive fields, 5 Max-Pooling layers for spatial pooling with pool size 2 x 2, and 3 completely connected layers with the final layer triggered using Soft-max function. The hidden layers in this DCNN are activated using the Rectified nonlinearity(ReLU) activation function, while the fully linked layers are regularised using dropout regularisation. Figure 5 depicts the structure of VGG16's architecture. ML classifiers are trained using the characteristics retrieved by the VGG16 in the form of feature vectors.

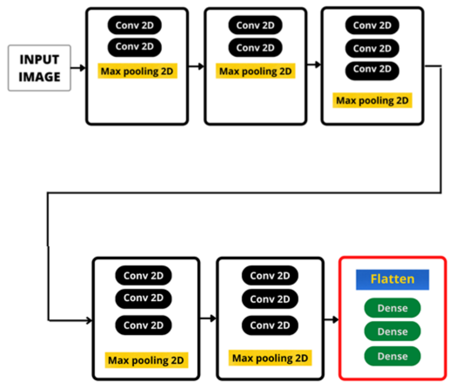


Figure 7. Architecture of VGG16 CNN

In CNN to compute loss a variety of optimization strategies are available.In this project we have used the following optimizer, loss function,Activation Function.

4.2.2 Adam optimizer

An optimizer is a function that adjusts the neural network's properties like weights and learning rate. It aids in the improvement of precision. The technique is particularly efficient when dealing with large situations with a lot of data or parameters. It is quick and uses little memory. We're utilizing the Adam optimizer here, which works for both convex and non-convex functions. It is comparable to momentum and employs exponential decay averaging for previous squared gradients, which we designated by vt.

$$m_t = \beta m_{t-1} + (1-\beta)\left[\frac{\delta L}{\delta w_t}\right] \tag{1}$$

The updated weight equation is given by

$$W_{t+1} = W_t - \frac{\alpha}{\sqrt{\hat{v} - \varepsilon}} * \hat{m} \tag{2}$$

4.2.3 Loss Function

A function is a measure of how good your prediction model does it's work.it quantifies the difference between the expected outcome and the outcome produced.In this project we have used categorical crossentropy .We have more than 1 image classification hence we can utilize loss function to get more accuracy.

4.2.4 Activation Function

A function decides whether a neuron should be activated or not. ex- RELU, SOFTMAX.
We have used softmax activation function.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{3}$$

σ = softmax
$\vec{z}$ = input vector
e^{z_i} = standard exponential function for input vector
K = number of classes in the multi-class classifier
e^{z_j} = standard exponential function for output vector
e^{z_j} = standard exponential function for output vector

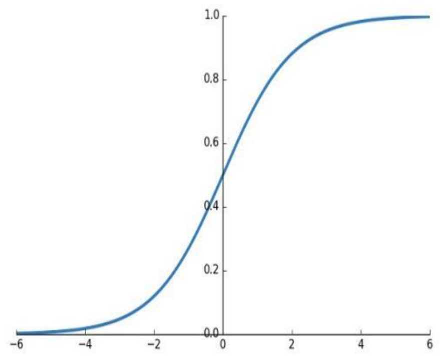


Figure 8. Softmax Activation Function

5. MODEL ACCURACY

Model accuracy is crucial step before deploying a machine learning model into production environment.It helps us to determine which algorithm gives best results and minimum loss. For a model the difference between actual output and predict output should be minimum for high accuracy.

5.1 Breast Cancer (Feature 1)

The accuracy is calculated with the help of confusion matrix.It's a summary of classification issue prediction outcomes, including the number of accurate and wrong predictions, which are summarized with count values and divided down by class. The secret to the confusion matrix is this. It demonstrates how your classification model might become perplexed while making predictions. It provides insight into not just the errors produced by a classifier, but also the sorts of errors made.

	Predicted **NO**	Predicted **YES**
Actual **NO**	**TP** True Positive	**FN** False Negative
Actual **YES**	**FP** False Positive	**TN** True Negative

Accuracy = (TP + TN) / (TP + TN + FP + FN)

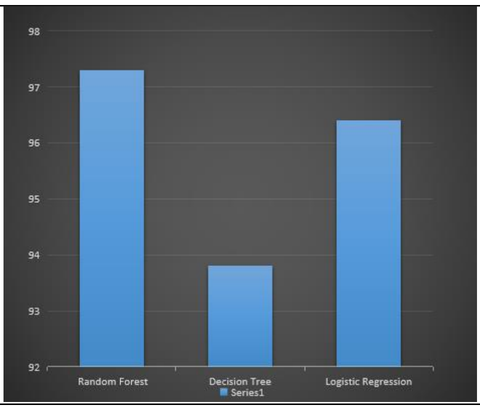


Figure 9. Bar graph depicts accuracy of Ml algorithms

Out of above tested supervised algorithms random forest found to be more accurate with better performance and it solves the over fitting problem.

5.2 Lung Disease(Feature 2)

Our model was put to the test on its test dataset once it had been trained. The findings were assessed using the accuracy, recall, precision, and F1 score. This section covers all of the evaluation measures used in this project. When identifying normal and pneumonia cases, true positive (TP) refers to the proportion of pneumonia X-rays correctly identified as pneumonia, true negative (TN) refers to the proportion of normal X-rays correctly identified as normal, false positive (FP) refers to the proportion of pneumonia X-rays incorrectly identified as pneumonia, and false-negative (FN) refers to the proportion of normal X-rays incorrectly identified as normal.

5.2.1 Accuracy

The accuracy tells about the amount of correctness of the trained model.It is the total number of right predictions over total predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

5.2.2 Recalls

It indicates how many of the model's positive predictions are truly positive, out of a total of positive labels.

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{Total\ Actual\ Positive} \tag{5}$$

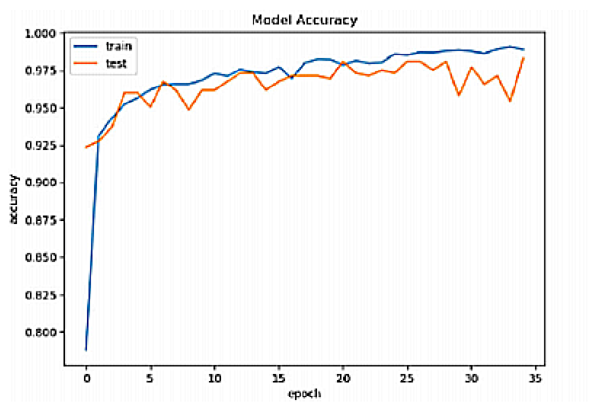


Figure 10. Accuracy vs Epoch graph of Adam optimizer

The training data accuracy graph abruptly grows from epoch 0 to epoch 2, then progressively increases until epoch 35, which is comparable to 98.9%, while the testing data accuracy graph is 98.2 percent for epoch 35 using the Adam Optimizer.In this project we have set the epoch value to 5 which yielded best results in predicting lung disease.

6.USER INTERFACE

The UI is the most important part of this application where the user tries to interact with the application functions. In this project the application UI consists of two features which helps in prediction of breast cancer and lung disease.The UI is designed using flexbox technology and accessible on web ,Android ,ios. Flexbox technology is flexible and adaptive to different screen size . It helps in showing every details to the user which is there in user interface irrespective of display aspect ratio.This front end application is integrated with the models described above at the backend.The backend of this application is served by Python Flask Framework.

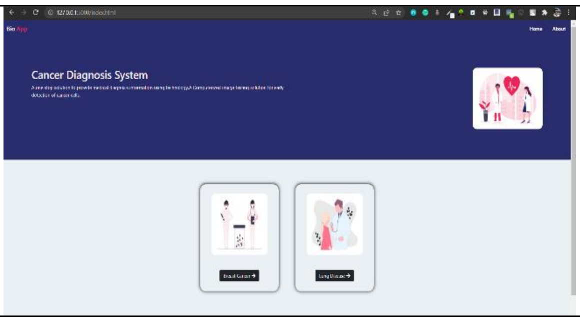
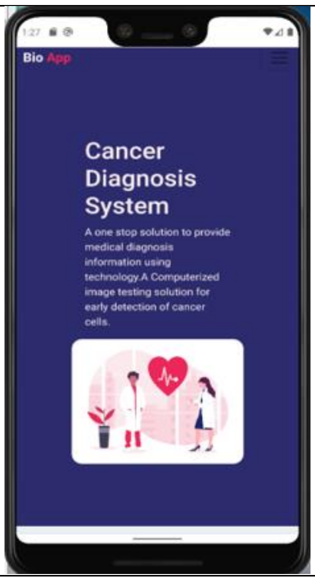


Figure 11 .UI on Web App



Cancer Diagnosis System

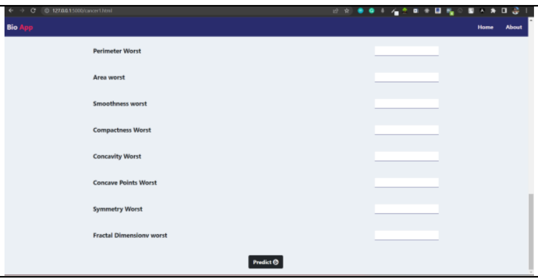
A one stop solution to provide medical diagnosis information using technology.A Computerized image testing solution for early detection of cancer cells.

Figure 12 .UI on Android App

7.RESULTS

The results of both features is shown below from figure 13 onwards. These results are obtained after carefully observing the train test data set and accuracy for both breast cancer and lung disease which is a could be main cause for lung cancer.

7.1 Breast Cancer (Feature 1)



Perimeter Worst

Area worst

Smoothness worst

Compactness Worst

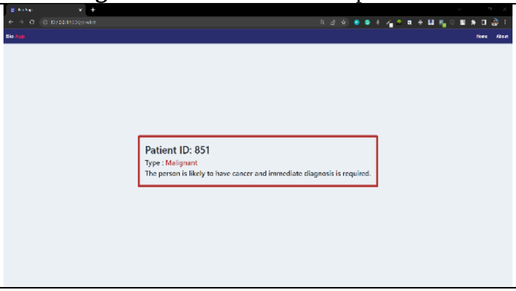
Concavity Worst

Concave Points Worst

Symmetry Worst

Fractal Dimensionv worst

Figure 13. Numerical Data Input Form

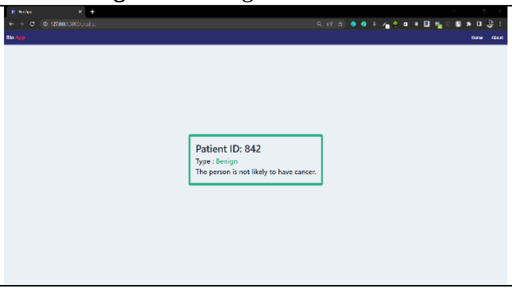


Patient ID: 851

Type : Malignant

The person is likely to have cancer and immediate diagnosis is required.

Figure 14. Malignant Tumor result



Patient ID: 842

Type : Benign

The person is not likely to have cancer.

Figure 15. Benign Tumor result

The above results are for prediction of Breast Cancer using machine learning supervised algorithm. After Uploading the data as a input onto the UI the user can click on predict button to analyse the medical report.

7.2 Lung Disease (Feature 2)

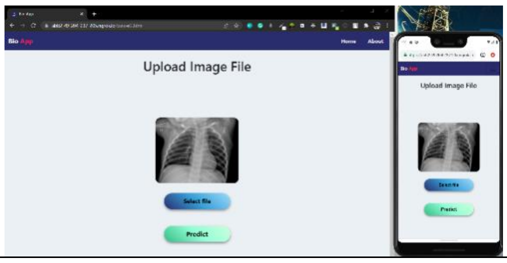


Figure 16. Upload Image

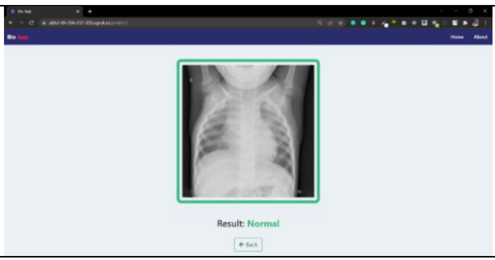


Figure 17. Normal Result

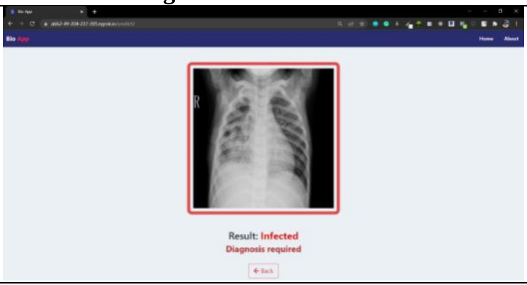


Figure 18. Infected Result

In feature 2 the user has to select an image file as shown in figure 12 which needs to be tested and click on predict button to check for the result .If the person's lung is normal then figure 13 is displayed on the screen else if it is infected then figure 14 is displayed on screen.

8. CONCLUSION

The prediction of breast cancer & lung disease with best accurate algorithm is integrated with our application.The application UI comprises of two features where feature 1 is predicting breast cancer and feature 2 is predicting lung disease which is a primary symptom for lung cancer.However, we can add more features to this application which can be solved using computing devices.The deep computer science techniques which includes artificial intelligence , Deep learning and machine learning that solves the major problem in health care sector that is predicting cancer is implemented. The proposed model takes data or image files as an input and predicts whether the person is having cancer or not at early stage and the prediction accuracy is around 96% . The application is developed using modern technology which can run on any computing device which has browser capability and it is scalable.

9. REFERENCES

1. Kakeda, S., Moriya, J., Sato, H., Aoki, T., Watanabe, H., Nakata, K. "Improved Detection of Lung Nodules on Chest Radiographs Using a Commercial Computer-Aided Diagnosis System", ***American Journal of Roentgenlogy***, 182(2), 505– 510. doi:10.2214/ajr.182.2.1820505.
2. Gurcan, M. N., Sahiner, B., Petrick, N., Chan, H.-P., Kazerooni, E. A., Cascade, P. N., & Hadjiiski, L. 2002. "Lung nodule detection on thoracic computed tomography images: Preliminary evaluation of a computeraided diagnosis system",***Medical Physics***, 29(11), 2552– 2558. doi:10.1118/1.1515762.
3. Awai, K., Murao, K., Ozawa, A., Komi, M., Hayakawa, H., Hori, S., & Nishimura, Y. "Pulmonary Nodules at Chest CT: Effect of Computer-aided Diagnosis on Radiologists' Detection Performance", ***Radiology***, 230(2), 347352. doi:10.1148/radiol.230203004.
4. Anji Reddy V., Soni Badal, "Breast cancer identification and diagnosis techniques", ***Springer*** (2020)
5. S.K., L., Mohanty, S. N., K., S., N., A., & Ramirez, G.,"The optimal deep learning model for classification of lung cancer on CT images",***Future Generation Computer Systems,*** doi:10.1016/j.future.2018.10.009.
6. Worawate A, Thirach, A., Marukatat, S., & WilaiprasitpornT.,"Automatic Lung Cancer Prediction for Chest X-ray Images Using the Deep Learning Approach",***Biomedical International Conference*** (BMEiCON-2018).
7. Haarburger, C., Weitz, P., Rippel, O., & Merhof, D.,"Image-Based Survival Prediction for Lung Cancer Patients Using CNN",***IEEE 16th International Symposium on Biomedical Imaging*** (ISBI 2019). doi:10.1109/isbi.2019.8759499.
8. Shweta K., SunitaSoni Y., "Weighted naive bayes classifier: A predictive model for breast cancer detection",***International Journal Computer Applications.***, 133 (9) (2016), pp. 32-37
9. Devi R.D.H., Devi M.I."Outlier detection algorithm combined with decision tree classifier for early diagnosis of breast cancer" ***Int. J. Adv. Engg. Tech***./Vol. VII/Issue II/April-June, 93 (2016), p. 98
10. Hu, Z., Tang, J., Wang, Z., Zhang, K., Zhang, L., & Sun, Q.,"Deep learning for image-based cancer detection and diagnosis − A survey",***Pattern Recognition***, 83, 134– 149. doi:10.1016/j.patcog.2018.05.5858
11. Vikas C., Saurabh P., and BB Tiwari, "Prediction of benign and malignant breast cancer using data mining techniques", ***Journal of Algorithms & Computational Technology*** 2018, Vol. 12(2) 119–126 ! The Author(s) 2018,pp. 119-126